TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report

On

Cricket Shots Analysis using AI

Submitted By:

Bhuwan Khatiwada (THA075BEI009)

Bishwa Prakash Subedi (THA075BEI011)

Niraj Duwal (THA075BEI028)

Rewan Gautam (THA075BEI032)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2022

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report

On

Cricket Shots Analysis using AI

Submitted By:

Bhuwan Khatiwada (THA075BEI009)

Bishwa Prakash Subedi (THA075BEI011)

Niraj Duwal (THA075BEI028)

Rewan Gautam (THA075BEI032)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor's Degree in Electronics and Communication, and Information Engineering.

Under the Supervision of

Associate Prof. Suramya Sharma Dahal

March, 2022

DECLARATION

We hereby declare that the report of the project entitled "**Cricket Shots Analysis using AI**" which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics, Communication and Information Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Bhuwan Khatiwada (THA075BEI009)

Bishwa Prakash Subedi (THA075BEI011)

Niraj Duwal (THA075BEI028)

Rewan Gautam (THA075BEI032)

Date: March, 2022

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to **the Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled "**Cricket Shots Analysis using AI**" submitted by **Bhuwan Khatiwada**, **Bishwa Prakash Subedi**, **Niraj Duwal** and **Rewan Gautam** in partial fulfillment for the award of Bachelor's Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering

_____

Project Supervisor

Associate Prof. Suramya Sharma Dahal

Department of Electronics and Computer Engineering, Thapathali Campus

_____

External Examiner

_____

Project Coordinator

Er. Umesh Kanta Ghimire

Department of Electronics and Computer Engineering, Thapathali Campus

_____

Head of Department

Er. Kiran Chandra Dahal

Department of Electronics and Computer Engineering, Thapathali Campus

March, 2022

COPYRIGHT

ACKNOWLEDGEMENT

Firstly, we would like to dedicate our regards to the Institute of Engineering (IOE) for the inclusion of this Minor Project on the syllabus for the course of Bachelors of Electronics, Communication and Information Engineering.

Also, we would like to thank our Department of Electronics & Computer Engineering, Thapathali Campus for wonderful learning atmosphere throughout our time here at Thapathali Campus and for giving us with this exciting opportunity to test our knowledge through this Minor project and our supervisor Associate Prof. Suramya Sharma Dahal sir for providing us with proper guidance and feedbacks throughout this project.

The experience of doing this project will surely enrich our technical and teamwork skills to a great extent.

Bhuwan Khatiwada (THA075BEI009)

Bishwa Prakash Subedi (THA075BEI011)

Niraj Duwal (THA075BEI028)

Rewan Gautam (THA075BEI032)

ABSTRACT

Cricket is often described as a contest between bat and ball, and batting is considered one of its prime disciplines. While batting a batter tries to hit the ball with a different variety of shots to guide the ball in any specific direction of their wish. The types of shots include cover drive, straight drive, pull shot, reverse sweep, and many more. It is tedious task to classify shots manually and provide the insights for the batter. Our goal is to classify the shot using AI and to provide an analysis of how well someone plays a particular shot. This can be achieved by collecting the required image data, extracting features, feeding to the machine learning or deep learning algorithms, and generating the classifier output. We have implemented the model using EfficientDet and SVM which works better with less data.

Keywords: AI, CNN, Cricket, ML, Model, Pose, Shots, SVM

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BI	Business Intelligence
BiFPN	Bi-directional Feature Pyramid Network
CCM	Compact Camera Module
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CV	Computer Vision
EC2	Elastic Compute Cloud
EfficientDet	Efficient Detection
EfficientNet	Efficient Network
GPU	Graphics Processing Unit
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
LTS	Long Term Support
MATLAB	MATrix LABoratory
ML	Machine Learning
MySQL	My Structured Query Language

NumPy	Numerical Python
OpenCV	Open Computer Vision
OS	Operating System
PC	Personal Computer
RAM	Random Access Memory
RBF	Radial Basis Function
RCNN	Region-based Convolutional Neural Network
SQL	Structured Query Language
SSH	Secure Shell
SVC	Support Vector Classifier
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TPU	Tensor Processing Unit
TV	Television
UI	User Interface
VOC	Visual Object Classes
XML	Extensible Markup Language
YOLO	You Only Look Once

1. INTRODUCTION

The use of automation and AI in the modern-day has increased in a high number as it can solve complex problems which is very tedious doing manually. It has found its way in various sectors of our day-to-day life from when we get up to when we go to bed. Be it how we get ready for our day ahead, how we prepare food, how we get recommended feeds or how we get our news AI has found its way everywhere. One of the sectors where AI's presence is in increasing demand is sports. With huge amount of data available online and increasing computational resources, there should not be a second thought to not use AI in the sports sector.

1.1 Background

Cricket is one of the most popular sports in the world with a large global audience. It is followed and played by millions. There are mainly three skills in cricket, namely batting, bowling and fielding. Among these batting is considered as the prime skill which moves the game forward, as the game is a matter of who scores the most number of runs. The individuals who perform batting are referred to as batters. The main goal of these batters is to defend their wicket against the ball delivered by the bowlers while scoring as many runs as they can for their team. For this, the batters use their most handy weapon in the field of battle- a cricket bat. A batter physically moves in different positions to be able to counter the variety of balls bowled by the bowler. He/she uses the bat at different angles to access different parts of the ground. Depending on the position of the batter and the angles at which they use the bat there are several varieties of cricket shots such as the cover drive, the straight drive, the pull shot, reverse sweep, and many more.

1.2 Motivation

Cricket is a game that is close to the hearts of many people. There even are some who live and breathe cricket as some would say. It's a dream of many to represent their country at the highest level and they work hard at it to become a better player day in and day out. Our project is dedicated to these dreamers with a little hope that we can be a part of their journey. For any player, the little insights on their game can be of immense importance. Any analysis of their game and how can they improve can be

highly beneficial. Our project aims at classifying the type of shot played and analyzing how comfortable the player is at playing the corresponding shot.

1.3 Problem Definition

Batting is the art of reacting to a ball that can be traveling at extremely high speeds. So, the movements of the batters also must be very quick. To analyze or classify the type of shot played by the batter we require an image of a batter as they are about to connect with the ball. It is possible to classify the shots manually, but it would cost time and manpower. It is also possible to classify the shots developing our own algorithms by building a function to calculate angles between the body parts to see if they hit the shot right. Since, the angles or certain position might not be same for all person, the shots might get ambiguous. For the complex problem like this, AI can be the savior.

1.4 Objectives

The major objectives of this project are as follows:

- To classify the shot played by the batter using machine learning model into six categories (Cover Drive, Straight Drive, Scoop, Cut, Pull, Leg Glance).
- To make a profile for each player displaying frequent and least shots played.
- To determine how well the batter played the shot in three classes (Missed, Edged, Perfect).

1.5 Project Scope and Applications

This project can be used by anyone practicing the art of batting. Cricket academies where many people come to learn how to bat this project can help guide them on the aspects of their game where they need to improve on, or it can be used by any cricket enthusiast personally to evaluate their game. The scope of this project will be on the following:

- In cricket academies
- For broadcasters
- For professional teams

1.6 Report Organization

The presented project report has been categorized into nine chapters.

- Chapter 1 provides the general introduction of the project. It includes all the background information, motivation, objectives, and scope of the project. It explains what the project is and why it is needed.
- Chapter 2 is the literature review; it contains summaries of the pervious works performed and theories related to this project. It gives us idea of how others went about solving the problem that was ahead of us, the methodologies used before, and the results achieved by them.
- Chapter 3 describes the different requirements necessary during the development and implementation of this project. It includes two sub sections, namely Hardware Requirements and Software Requirements. Hardware requirements contains the necessary hardware specifications and Software Requirements contain details of all the software packages and libraries that were used.
- Chapter 4 explains all the details of the dataset used like the categories the dataset is divided into, the number of data in the dataset and how the data set is processed for training the models.
- Chapter 5 contains the System Architecture and Methodology, which explains the overall architecture of the systems, the block diagrams, the workflow, and details of the methodology used to develop the system.
- Chapter 6 explains how the system was implemented to work under the pre-mentioned methodologies. It explains how the hardware and different programming languages, libraries and packages were used to implement the methodology.
- Chapter 7 includes the results and analysis of the obtained results. The results are shown in graphical forms, the accuracy and loss values are also given. Confusion matrix and heatmaps are also provided to evaluate and compare the different outputs for varieties of inputs.
- Chapter 8 describes how the system can be further improved from its present states so that we can increase its scope or application

- Chapter 9 discusses the overall project in summary and gives a brief conclusion about it.
- Chapter 10 contains additional information like the project budget, timeline of the project, used commands and code snippets.
- The references which were taken throughout the course of this project development are included at the end of the report.

2. LITERATURE REVIEW

2.1 Deep Learning

Deep learning is a science of making machines mimic human intelligence to perform required tasks and get better themselves with experience or the data they collect. Computer vision (CV) is a field of study that helps computer "see" and differentiate the different objects or contents in an image or videos. With the advent of deep learning algorithms, the accuracy for object detection has increased drastically. Object detection and tracking are one of the most discussed applications in the field of deep learning. It has been a research area because of its tremendous application in various ways like medical diagnostics, automations, fraud detections, self-driving cars, and a lot more to discover. Complex models like Convolutional Neural Networks (CNN) are the key component of this kind of applications.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep learning algorithms comprised of neurons that self-optimize through learning. The neurons receive input and perform scalar product and non-linear function [1]. In other words, convolution is a linear operation like a linear equation, dot product, or matrix multiplication. CNNs are mostly used in image recognition making CNNs more focused on images. CNNs are comprised of mainly three layers which are convolutional layers, pooling layers, and fully connected layers. CNN architecture has been formed with those layers.

2.2.1 Convolutional Layers

Convolutional layers are the major building blocks used in convolutional neural networks. A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input result in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. The innovation of convolutional neural networks is the ability to automatically learn many filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification.

2.2.2 Pooling Layers

Pooling layers are the layers added after convolution layer in CNN that perform down sampling of an image. They are the building blocks of CNN, and they are used to reduce the dimension of feature maps. As they reduce the dimension of feature maps, it reduces the number of parameters to be learned and number of computations to be performed. Thus, pooling layers come very handy in CNN as they reduce overfitting of the model.

There are different types of pooling which are as:

1. Max pooling

   Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

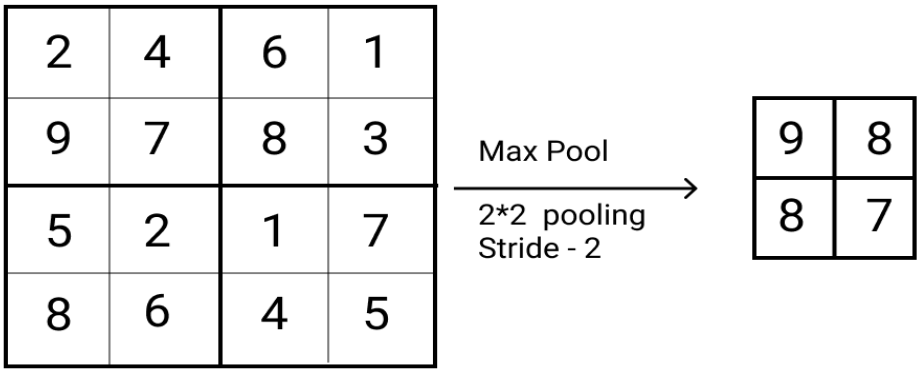


Figure 2-1: Max Pooling

2. Average pooling

   Average pooling computes the average of the elements present in the region of feature map covered by the filter. Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of features present in a patch.

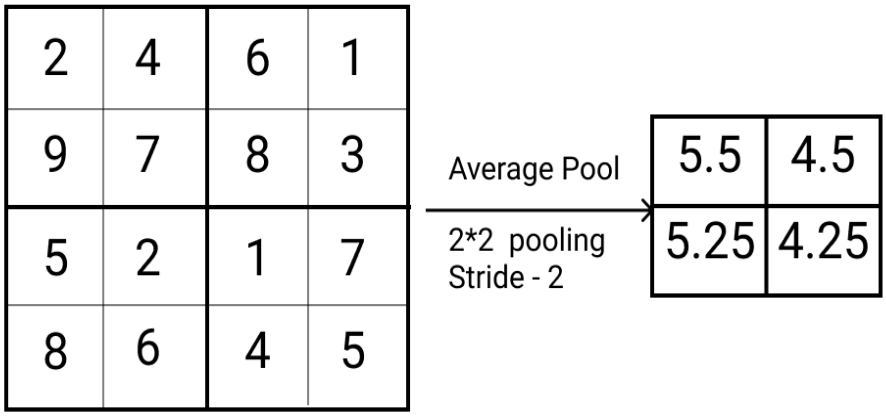


Figure 2-2: Average Pooling

2.3 Previous Works

Some works have been done on cricket analysis in last decade using different approaches. A group of researchers from Bangladesh which includes D Karmaker, AZM E Chaudhary, M S U Miah, M A Imran, M H Rahman had published a paper "Cricket Shot Classification using Motion Vector" in 2015. It was based on directional optical flow vectors created by body parts from videos and then transferring the vectors to angle [2]. The paper included 8 cricket shots classified in a frame according to angle ranges for each shot.

Table 2-1: Vector sum and angle range of four shot class [2]

Shot	**Square Cut**	**Hook**	**Flick**	**Off Drive**
Angle Range (in degree)	+136 to +180	+46 to +90	-45 to -89	-90 to -134
Vector Summation	-4.96e+03 +1.73e+03i	+8.96e + 02 +1.9e + 02i	+6.81e+02 -2.2e+03i	-6.19e+02 +2.37e+03i

The accuracy level of this approach was not satisfactory, the maximum accurate result was obtained on Off Drive shot which was 63.57%.

In another research paper, Foysal, Islam, Karim and Neehal published a paper entitled "Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots". They used 13 layered Convolutional Neural Network to extract the feature to classify six categories of cricket shots [3]. The CNN model consisted of three convolution layer, three max pooling layer, four dropout layer and two dense layers.

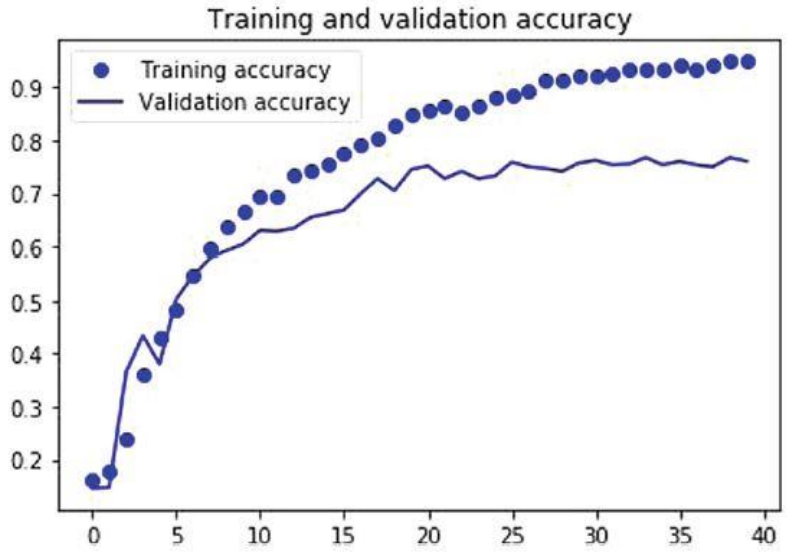


Figure 2-3: Training and Validation Accuracy [3]

If we analyze the performance of this model, the model seems to be overfitting because there is huge gap in accuracy level of training and validation data.

We approached to solve this with a different kind of technique to classify cricket shots. Instead of directly feeding the data to the algorithm, we extracted features using pose detection of the batter, filtered the necessary features and fed it to the machine learning algorithm. Since the pose of the batter highly determines the shot being played, the coordinates of the body parts gave us better results.

Object detection is a computer vision technique whose job is to locate a certain object within an image or video. Several research has been carried out in object detection in recent years and authors have published their paper citing as 'state of the art' model. Some of the popular models include R-CNN, Fast R-CNN, YOLO, etc. EfficientDet is one of those object detection models which was published in 2020 by Google Brain team. Model efficiency has been drastically improved in recent times but in expense of

number of parameters. EfficientDet offers higher accuracy as well as reduction in number of parameters making the model efficient.

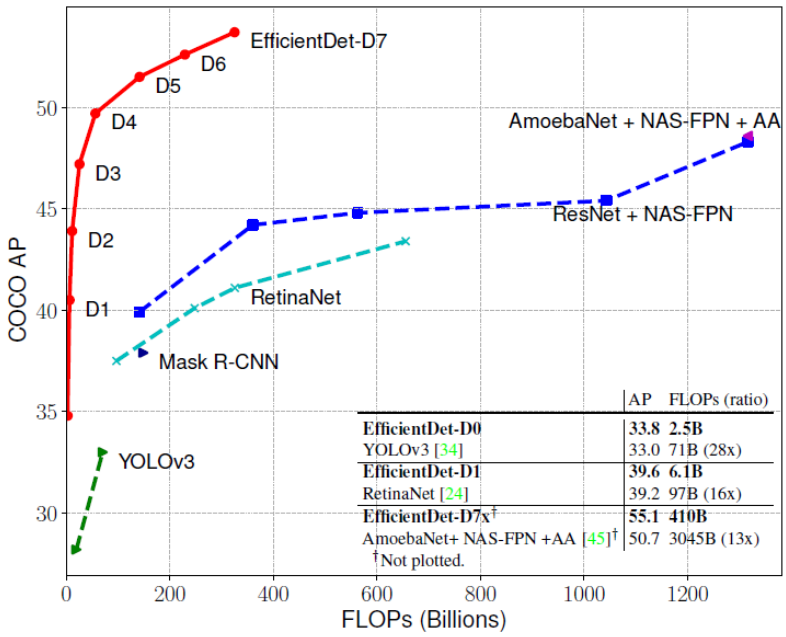


Figure 2-4: Accuracy vs number of parameters [4]

EfficientDet utilizes several optimization and backbone tweaks, such as the use of a BiFPN, and a compound scaling method that uniformly scales the resolution, depth and width for all backbones, feature networks and box/class prediction networks at the same time [4].

Image resolution,

$$R_{input} = 512 + \emptyset.128 \tag{2.1}$$

BiFPN,

$$W_{bifpn} = 64\left(1.35^{\emptyset}\right) \tag{2.2}$$

$$D_{bifpn} = 2 + \emptyset \tag{2.3}$$

Box/class predict network,

$$D_{box} = D_{class} = 3 + \emptyset/3 \tag{2.4}$$

3. REQUIREMENT ANALYSIS

The system captures the footage of the batter and analyzes it. The project is more focused on software part and hardware components are there to support and make use of the software.

3.1 Hardware Requirement

Any normal PC is preferable for training the model using Google Colab but if we want to train it on our own system, high CPU processors, high GPU or TPU and RAM are necessary. This is because deep learning must deal with lots of data and would cost us time. For the deployment, we need a mobile phone for the app that we have built to predict the shots on.

3.2 Software Requirement

The system requires coding which is written in python for backend purpose and for frontend we used dart programming language for developing mobile application. Under the hood, frameworks like TensorFlow, OpenCV contribute to deep learning and flutter contribute to improvement of UI rendering and cross platform application development. For training of model, we used Google Colab which is free cloud service that provides free GPU. The testing of the model can be obtained in PyCharm which is a python IDE.

3.2.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics [5]. Python is easy and simple where it takes no time to make a program from pseudo-code. Python's popularity these days can be examined with its large number of modules and packages encouraging code reusability. The question might be raised like "Why use Python?" which is quite slow compared to a language like C++ when speed is of major concern. It is because python is handy when it comes to deep learning. Different high-level machine learning libraries exist for only python. Also, there is quite many tutorials or communities that will help learners to step the foot in deep learning using python focusing on logics and understanding neural networks rather than the syntax.

3.2.2 TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning developed by the Google Brain team for internal Google use [6]. Built in 2015, it is the most widely used machine learning and deep learning package. Tensors are the multidimensional arrays which are commonly operated in neural networks and hence the name 'TensorFlow'. It helps developers to create multi layers neural networks to implement classification, regression, discovering prediction and creation. TensorFlow is designed in Python programming language that makes it easy to understand. The library can run on multiple CPUs and GPUs and is available in various platforms. It has high scalability of computation across machines.

3.2.3 OpenCV

OpenCV is an open-source computer vision and machine learning software library [7]. It is written in C and C++. It was focused to work for computational efficiency in real-time applications which takes advantage of multicore processors. In OpenCV, we would get at least 30 frames per second which is why it performs faster in real-time detection. The library has more than 2500 optimized machine learning and computer vision algorithms. The applications of these algorithms include detecting and recognizing faces, objects, detecting colors, extract 3D models, track movement in the camera, determine human actions in the camera, etc.

3.2.4 NumPy

Deep learning involves heavy computational task and python was not made for fast numeric solutions. It was 2005 when Travis Oliphant came to rescue with his NumPy. NumPy, or Numerical Python, is an open-source python library for scientific computing. It can perform various operations on arrays which are multidimensional. NumPy is suitable for operating advanced mathematical operations more efficiently and with less code.

Python with the support of numpy (also scipy and matplotlib) has now replaced numerical computing environment like MATLAB in recent years. The open-source software library like NumPy is the choice of many researchers and data scientist.

3.2.5 Matplotlib

When we are performing some machine learning tasks and want to see how our model has performed and see how it has been learning. We do not want that information to be loaded in some power BI, tableau or even excel for visualization. Python is a powerful language and one of the reasons behind this is that it has library for everything we need. Matplotlib is an amazing python package for data visualization. We can create various types of graphs including scatter plot, bar plot, histogram, boxplot, pie chart, heatmaps, etc. Using these graphs, we can analyze and see how well our problem has been solved easily.

3.2.6 Mediapipe

MediaPipe is an opensource framework developed by Google which offers applied machine learning pipelines. When using mediapipe, it offers readymade solutions for cases so there's no need of training the model. It is extremely easy to build pose detecting system using MediaPipe.

3.2.7 Dart

Dart is a client-optimized language for developing fast application. It is an open-source general-purpose object-oriented programming language. The main goal of dart language is to develop applications suitable for cross platform. It was originally developed by Google. It can be used for both server side as well as the user side.

3.2.8 Flutter

Flutter is a simple and high-performance framework based on Dart language. It is a cross-platform software development framework that was presented by Google in 2015 and was first released in May 2017. “Everything is widget in flutter”. It means flutter widgets are everywhere in an application. Even the app itself is a widget in flutter. There are two classed based widgets in flutter. One is stateless widget and other stateful widget. we cannot change state on stateless widget but can change state on stateful widget.

3.2.9 SQL

Structured Query Language (SQL) is a standard language for accessing and manipulating databases. [8] It includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language, but there are many different versions of the SQL language. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres, and SQL Server use SQL as their standard database language. SQL is widely popular because it offers the following advantages:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures, and views.

3.2.10 Google Colab

Google Colaboratory or Google Colab is a free Jupyter Notebook environment. It is a free cloud-based service by Google, so no setup or software installation is required. It runs Python 3 and comes with many popular Python libraries such as Pandas, NumPy, Tensorflow, Keras, OpenCV preinstalled so it is useful in works regarding Data Science and Machine Learning. Colab requires a Google account and any notebooks created are stored in Google Drive associated to that account. Colab leverages the collaboration features of Google Docs, where we can share our notebook with multiple people easily and can work on the same notebook at the same time without any issue. By connecting Colab to Google Drive, we get upto 15 GB of disk space for storing our datasets. Google also provides use of free NVIDIA TESLA K80 GPU to help execute our code better.

3.2.11 Pandas

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. It is a python library used for analyzing data and working with datasets. It has functions for analyzing,

cleaning, exploring, and manipulating data. It allows us to analyze big data and make conclusions based on statistical theories. It can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

3.2.12 Scikit-learn

Scikit-learn commonly known as sklearn is the python library which is mostly used for machine learning. It is open source and free package. It provides efficient tools for machine learning and statistically modeling that includes classification, regression, clustering etc. through a consistent interface in python. It is built upon NumPy, SciPy and Matplotlib and it is largely written in Python. Sklearn focuses more on modeling the data. Some of the popular groups of models provided by sklearn are supervised learning algorithm, unsupervised learning algorithm, clustering, cross validation, feature extraction, feature selection etc.

3.2.13 Amazon Web Services (AWS)

Amazon web service is a service of amazon that provides on demand cloud computing platforms and APIs to individual and companies on a metered pay-as-you-go basis. AWS is widely used and implemented by many companies. The fees are based on the entire combination of usage of hardware, software, OS, networking features that are chosen by the subscribers and the security is provided to the subscribers' system by Amazon itself. Rather than building actual physical server Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply through cloud computing [9].

3.2.14 Fast API

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints [10]. Some of the major features of FastAPI are:

- High-performance: As the name suggests, FastAPI is fast. It’s considered to be one of the fastest Python frameworks currently available, on par with NodeJS and GO.

- Robust: We can create production-ready code using automatic interactive documentation.
- Intuitive: FastAPI was designed to be easy to use and learn. It offers great editor support and documentation.
- Quick to code: FastAPI increases speed of developing features by 200%-300%.
- Fewer bugs: It reduces around 40% of induced bugs, requires less time for debugging.
- Compatible: It works well with the open standards for APIS, OpenAPI (previously known as Swagger), and JSON schema.
- Plugins: We can easily create plugins using dependency injection.

4. DATASET ANALYSIS

The dataset for shot classification was collected using web scrapping and capturing manually through our mobile devices. The types of shots we collected were Cut Shot, Cover Drive, Straight Drive, Leg Glance, Pull Shot and Scoop. There were around 100 images for each shot which was increased to 600 by performing data augmentation through rotating, shifting, changing brightness, and inserting noise. So total dataset for shot classification was 3600.

For preprocessing the image, we extracted pose features using mediapipe. We were able to extract co-ordinates of 33 3D body parts landmarks.

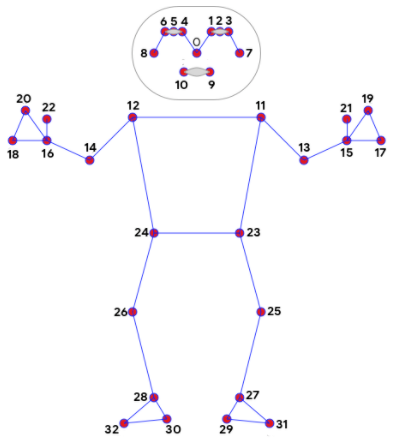


Figure 4-1: Mediapipe Body Landmarks [11]

Through one body part we could get the co-ordinate values for x, y, z axis and visibility. The sample csv dataset obtained after preprocessing is tabulated below:

	NOSE_x	NOSE_y	NOSE_z	NOSE_vis	LEFT_EYE_INNER_x	LEFT_EYE_INNER_y	LEFT_EYE_INNER_z	LEFT_EYE_INNER_vis
0	0.830450	0.833832	-0.053076	0.998836	0.817762	0.844470	-0.096186	0.999057
1	0.782654	0.799939	-0.166235	0.997595	0.766684	0.798882	-0.164054	0.997885
2	0.743410	0.744364	-0.351658	0.997786	0.734523	0.748967	-0.390113	0.998021
3	0.860915	0.461075	-1.368510	0.997924	0.848542	0.430726	-1.377788	0.998156
4	0.640927	0.741360	-0.965192	0.997877	0.629888	0.780623	-0.960477	0.998104
...	...	...	...	...	...	...	...	...
2857	0.310141	0.237103	-0.247231	0.987217	0.308201	0.223787	-0.236822	0.986523
2858	0.406546	0.339149	-0.580032	0.988353	0.401364	0.340548	-0.594165	0.987633
2859	0.526533	0.214372	-0.194660	0.989237	0.534293	0.200841	-0.152990	0.988406
2860	0.330912	0.203980	-0.153397	0.989479	0.323251	0.178546	-0.142844	0.988284
2861	0.460860	0.260569	-0.341700	0.988478	0.457445	0.213761	-0.349036	0.986082

Figure 4-2: Sample Dataset

For detection of bat, 201 images were used which were labelled and annotated using labelImg [19]. Annotation is stored in XML format for each image file item. The XML file contains the image width, height, depth, bounding box co-ordinates and the label it belongs to.

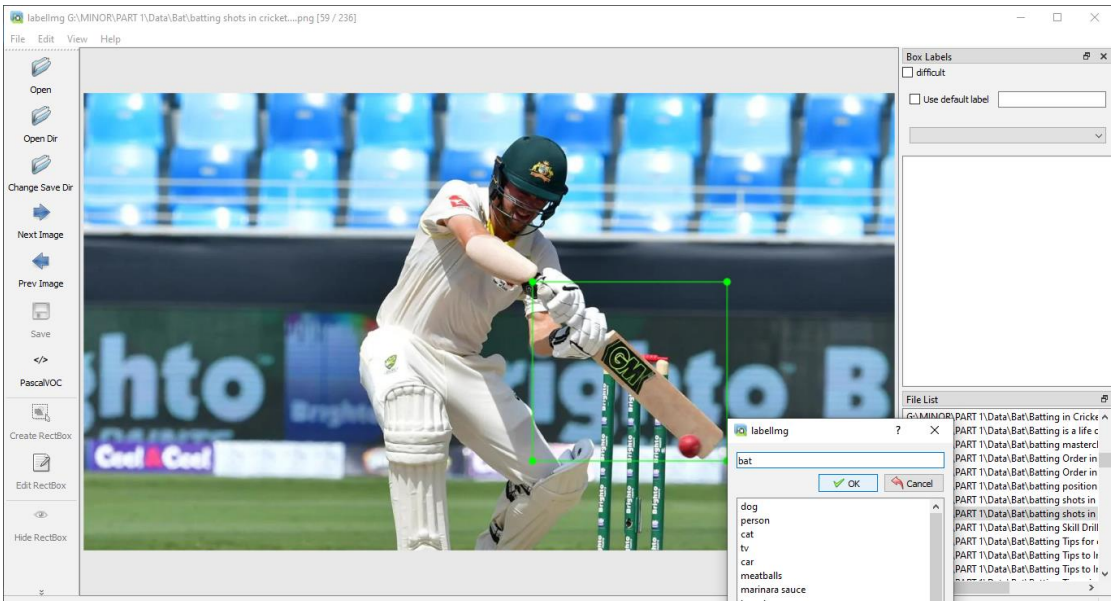


Figure 4-3: LabelImg Interface [12]

5. SYSTEM ARCHITECTURE AND METHODOLOGY

5.1 System Architecture

i. Data Collection and Feature Extraction

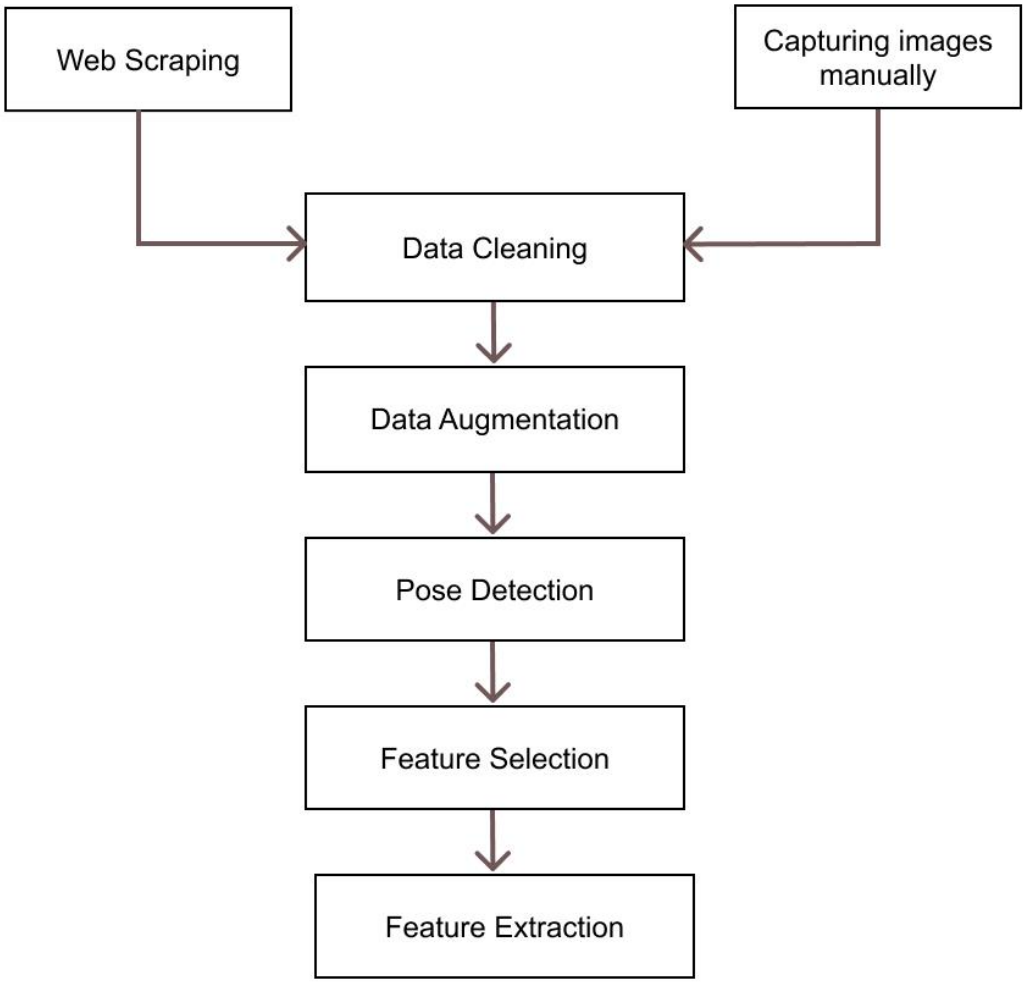


Figure 5-1: Data Collection and Feature Extraction Block Diagram

The data we require for our tasks is the images of batters playing a particular shot which is not available publicly on the internet. So, we scrapped it through google images and also captured images manually from ground to increase the data. A lot of data was misleading while collecting the data which needed to be cleaned so we cleaned them. Also, we used data augmentation technique to increase the data by modifying already collected images through rotating, shifting, zooming, etc. Since, the pose of the batter highly impacts the shot the batter is playing, we used it as a feature.

ii. Training

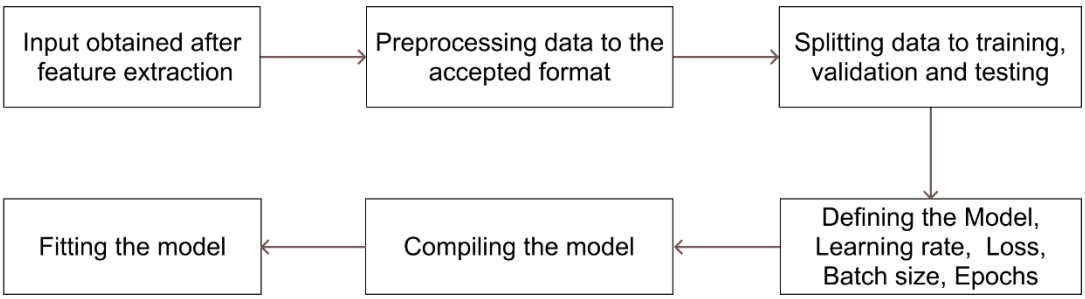


Figure 5-2: Training Block Diagram

After feature extraction, the data needs to be processed to the accepted format of the model which might be changing dimensions and the image size. Then, we need to split the data to training, validation, and testing dataset because the model should not be tested on same data it was trained on. This helps the model to perform better on unseen data. Then, we define the model, learning rate to find how fast or slow to go to find the minimum loss, batch size to train in batches, epochs to determine how many times we want to train the whole data. Now, the model is compiled, and the input is fitted to the model.

iii. ER Diagram

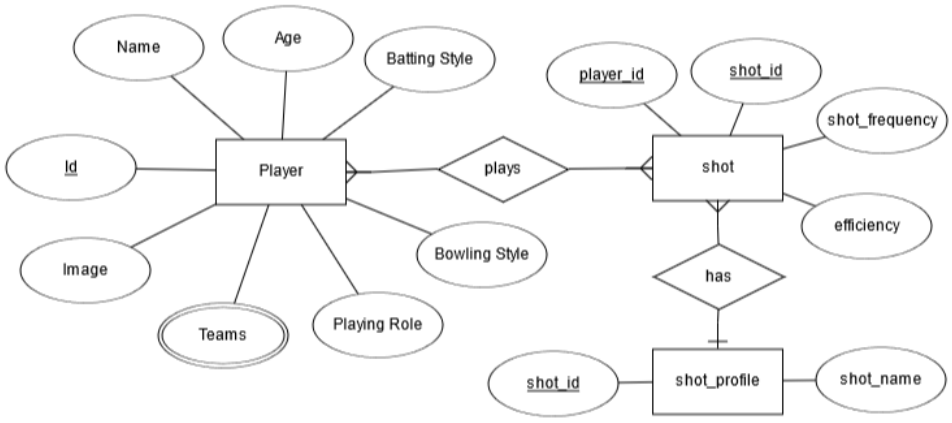


Figure 5-3: ER Diagram

The entity relationship diagram explains the architecture of our database. We have three entities Player, shot and shot_profile. The different attributes are:

- Player – id, name, age, image, batting style, bowling style, teams, playing role
- Shot – shot_id, player_id, shot_frequency, efficiency
- Shot_profile – shot_id, shot_name

Here ‘id’ is the primary key for the player entity, ‘shot_id’ for shot_profile entity, ‘shot_id’ and ‘player_id’ jointly as candidate key for shot entity. The relationship between player and shot entities is ‘plays’ because player plays the shot. Player and shot have many to many relationships. Also, the relationship between shot and shot_profile is ‘has’ because shot has shot_profile. Shot and shot_profile have many to one relationship.

iv. System Block Diagram

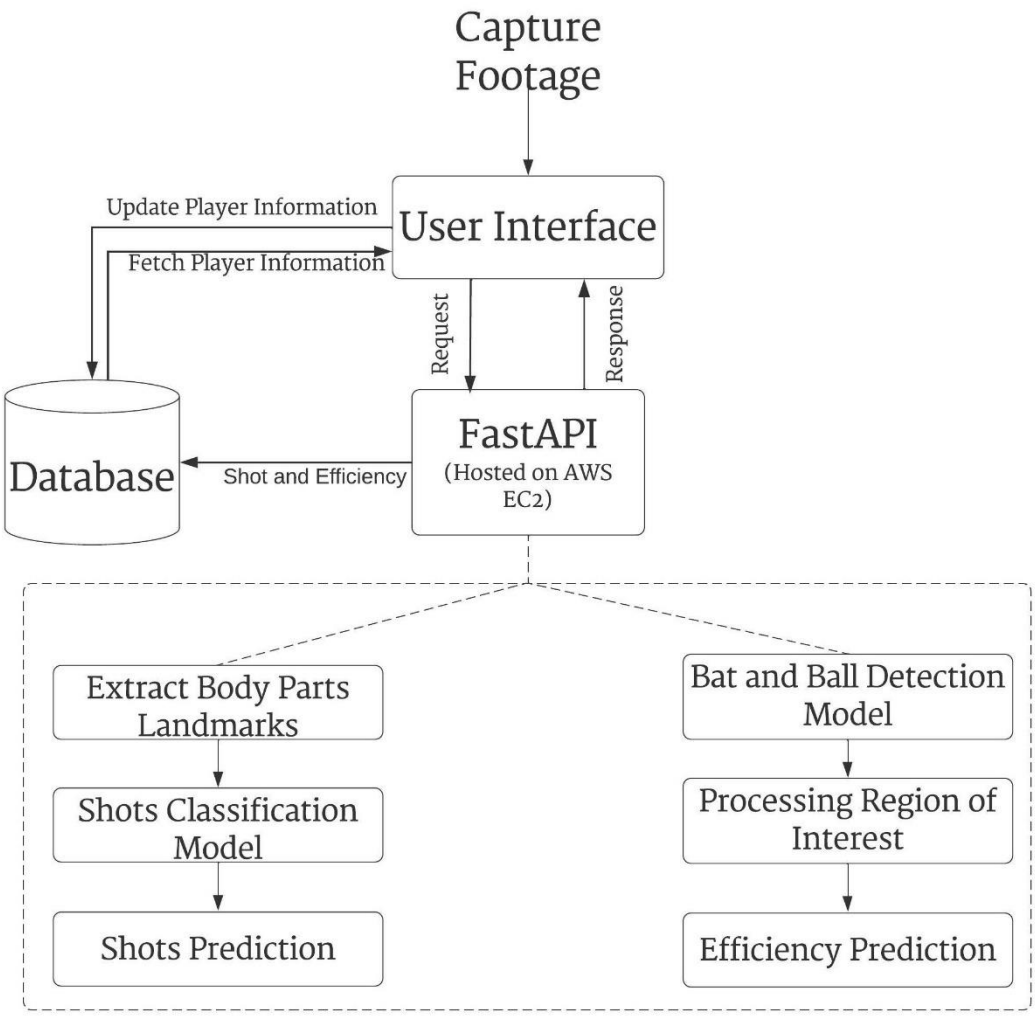


Figure 5-4: Working System Block Diagram

The above figure shows the system block diagram. Through the mobile application, the input image can be captured using device's camera or already captured image can be uploaded as input. This can be performed with the assistance of user interface. The user interface has features like uploading or capturing images, adding player profiles for new players and a portal for recent cricketing news and articles. The player profile includes information like name of player, age, batting style, bowling style, playing role and team. Through UI, we store this information in a database and fetch them from the same database whenever required.

We obtain the required input through the UI. The FastAPI which is hosted on AWS EC2 instance server loads the model and helps process the given input. The processing

steps can be seen in the given expanded view. For shot classification, firstly the body part landmarks are extracted from the input images and thus extracted features are fed to the shot classification SVM model which finally predicts the shot played. And for the shot efficiency, we feed the image to the bat detection EfficientDet model, the model processes the image to determine the region of interest i.e. the area of the bat. Then upon detecting the ball, depending on whether the ball lies within the region of interest or not, the efficiency of shot is predicted.

5.2 Working principle

i. Support Vector Machines (SVM)

For the classification of different cricketing shots, we are using Support Vector Machine (SVM) classifier. SVM is a supervised machine learning algorithm which works by plotting the data items in the space determined by the number of features we have in our dataset and finding the decision boundary i.e a hyper-plane that separates the classes.

A hyperplane in a n-dimensional space is a decision boundary of n-1 dimensional subset of that space that divides the space into required parts. SVM creates number of hyperplanes, and the best hyperplane is chosen according to marginal distance between nearest data items of each class. So, the hyper plane with the maximum marginal distance is considered optimal by SVM.

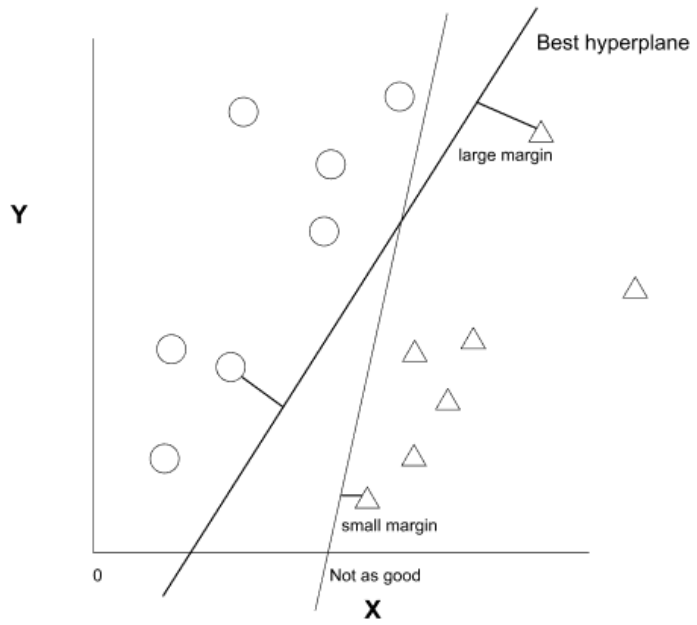


Figure 5-5: Best Hyperplane

Not all data points contribute to the orientation of hyper plane. The data points that are closer to the hyperplane influence the marginal distance and contribute to the orientation of hyper plane and are called support vectors. Due to this reason, SVM algorithm depends on only few data and are well suited for our task. Support vector pass through marginal planes which are parallel to the hyperplane.

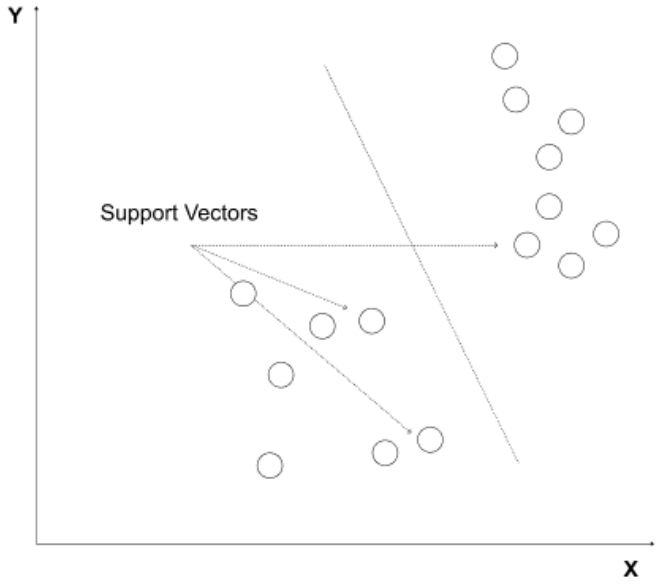


Figure 5-6: Support Vectors Illustration

It is quite easy to separate the data points if they are linearly separable but in many real-world tasks like of ours, the data are non-linear. We can convert it to linearly separable in higher dimensions. We can classify data by adding extra dimensions to it and projecting it back to original dimensions. This can be achieved by using SVM Kernels. Some of the popular SVM kernels are linear, poly, RBF, sigmoid, etc.

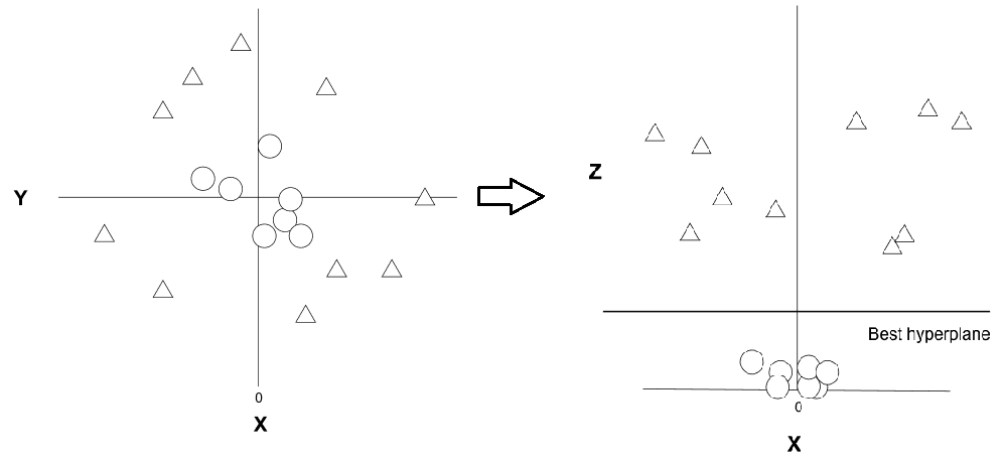


Figure 5-7: Conversion to Higher Dimension

For the multiclass classification, we use 'one -vs-all', approach by creating N SVMs for N classes and choose the class which has highest probability among all classes.

For the linear model, the equation of hyperplane can be formulated as,

$$wx - b = 0 \qquad (5.1)$$

where, $x =$ *input data*

$w =$ *weight vector*

$b =$ *bias vector*

If we have two labels say 1 and -1,

$$w.x_i - b \geq 1 \quad if\ y_i = 1$$
$$w.x_i - b \leq 1 \quad if\ y_i = -1$$

where, $x_i =$ *ith input data*

$y_i =$ *label of ith input data*

We use hinge loss as a cost function in SVM,

$$J = \lambda\|w\|^2 + L$$
$$= \lambda\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(wx_i - b))$$

$$J_i = \begin{cases} \lambda\|w\|^2, & y_i . f(x), \geq 1 \\ \lambda\|w\|^2 + 1 - y_i(wx_i - b), & y_i . f(x), \leq 1 \end{cases} \quad (5.2)$$

where, J = Cost function

L = Hinge loss

n = number of training examples

ƛ = regularization parameter

f(x) = wx_i-b

||w|| = norms of w which is proportional to 1/ (margin size)

The gradients can be calculated by differentiating the cost function with respect to weights and bias as,

$$\frac{dJ_i}{dw} = \begin{cases} 2\lambda w, & y_i . f(x), \geq 1 \\ 2\lambda w - y_i x_i, & y_i . f(x), \leq 1 \end{cases} \quad (5.3)$$

$$\frac{dJ_i}{db} = \begin{cases} 0, & y_i . f(x), \geq 1 \\ y_i, & y_i . f(x), \leq 1 \end{cases} \quad (5.4)$$

The update rule for the weights and bias to minimize the cost,

$$w = w - \alpha \frac{dJ_i}{dw} \quad (5.5)$$

$$b = b - \alpha \frac{dJ_i}{db} \quad (5.6)$$

Where, α = learning rate

We implement the algorithm creating a SVM class consisting of fit and predict method and the required attributes.

ii. EfficientDet

For the object detection which are bat and ball, we use EfficientDet algorithm. The detection model uses EfficientNet for feature extraction which acts as a backbone. EfficientNet uses compound scaling method. Unlike conventional practice that

arbitrary scales these factors, compound scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

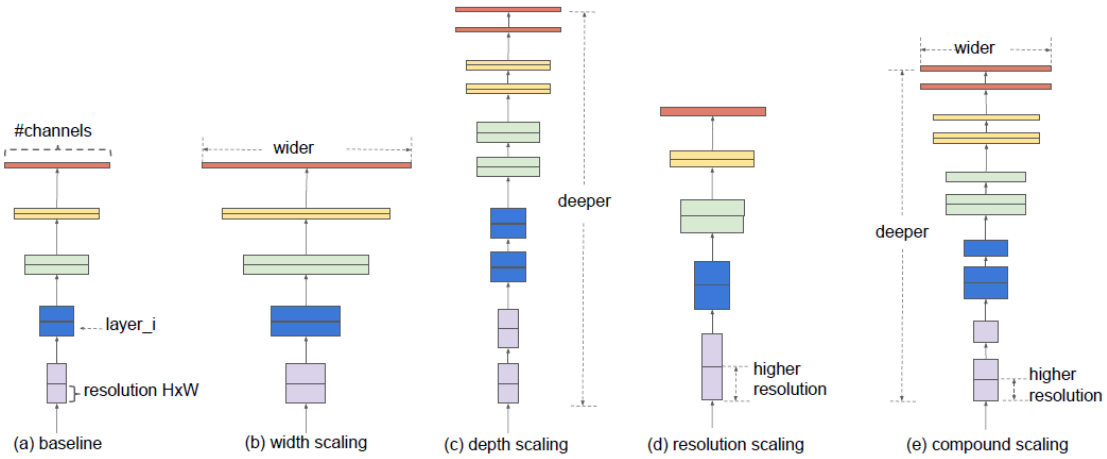


Figure 5-8 Different Scaling Methods [13]

The BiFPN as a shared class/box prediction as well as feature network introduces learnable weights to learn the importance of different input features, while repeatedly applying top-down and bottom-up multi-scale feature fusion. It takes level 3-7 features namely (P3-P7) and they are fed to network one by one. Since, we will be deploying our model in computationally cheaper devices, we need optimized models. EfficientDet in the best fit for our task.

To implement efficientDet model, we are using TensorFlow Lite model maker. The model maker simplifies the process of training a tensorflow lite model using custom dataset. Since, we have less amount of data, tensorflow lite model maker is useful in a sense that it uses transfer learning to train the model

6. IMPLEMENTATION DETAILS

6.1 Software Implementation

6.1.1 Google Colab

We are using Google Colab to write and execute our code in python. Since, it is easier, faster and provides us with free cloud RAM and GPU, it can help us to train our model quickly.

6.1.2 MediaPipe

MediaPipe's pose detection has come handy as we extract features from the pose of the batsman. The input image is passed through MediaPipe's pose processing method which returns the co-ordinates of the body parts in the form of x, y, z axes and a visibility index.

6.1.3 Pandas

The co-ordinates values returned by mediapipe is processed and stored in a csv file through pandas. Now, for feature selection we use Pearson's correlation. Pearson's correlation attempts to draw a line to best fit through the data of two variables. Pearson's correlation coefficient is the measure of the strength of a linear association between two variables. It indicates how far away all these data points are to this best fit line. It ranges from -1 to 1. Mathematically it is given by,

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Using pandas df.corr(method='pearson'), we find how all the features are correlated. Filtering the result for only the target variable, we obtained the correlation of target and other variables.

```
RIGHT_HEEL_z          -0.085503
LEFT_PINKY_x           0.202500
LEFT_FOOT_INDEX_y     -0.016139
LEFT_FOOT_INDEX_z      0.066333
RIGHT_THUMB_vis        0.454975
LEFT_THUMB_z          -0.099236
LEFT_HEEL_x           -0.181415
MOUTH_RIGHT_y          0.170178
LEFT_EAR_vis          -0.092461
RIGHT_ELBOW_z         -0.426689
```

Figure 6-1: Sample of Pearon's correlation of features with target variable

Now, using the threshold of 0.2, we filtered out the less important features.

6.1.4 NumPy

NumPy helps us in dealing with the numerical calculations while implementing the SVM algorithm. Without NumPy, our model would take more lines of code and might not converge quickly. So, the privilege of parallel computing of NumPy is there to deal with it.

6.1.5 Scikit-learn

We split the dataset using train_test_split function of scikit-learn so that 80% of data are used for training and 20% for testing.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_si
ze=0.2, random_state=42)
```

Here, X is the matrix of input features and y is the vector of target feature which is the shots labelled from 0 to 5 as:

Table 6-1: Shot with labels

0	Cut Shot
1	Cover Drive
2	Straight Drive
3	Pull Shot
4	Leg Glance Shot
5	Scoop Shot

test_size = 0.2 means 20% of the data for testing. Using sklearn, we are also able to interpret our model using its evaluation functions like classification_report, f1_score, confusion_matrix.

6.1.6 Matplotlib

Data visualization and analysis is performed using Matplotlib. The accuracy, loss, classes scores heatmaps, confusion matrix reports are generated.

6.1.7 Tensorflow

TensorFlow in our project, implements the efficientDet model for detecting the bat and ball from the footage. After labelling and annotating the images, it loads the data that are stored in Pascal VOC format. Pascal VOC is an XML file which is created for each image which stores the bounding boxes as (xmin, ymin, xmax, ymax). The annotation file and the image file are read by the tensorflow dataloader. The pretrained efficientDet is specified. Now, defining the batch size = 4, epochs = 100, the model gets trained.

```
43/43 [==============================] - 17s 395ms/step - det_loss: 0.3083 - cls_loss: 0.1843 - box_loss: 0.0025
Epoch 96/100
43/43 [==============================] - 15s 356ms/step - det_loss: 0.2827 - cls_loss: 0.1695 - box_loss: 0.0023
Epoch 97/100
43/43 [==============================] - 16s 371ms/step - det_loss: 0.3411 - cls_loss: 0.2041 - box_loss: 0.0027
Epoch 98/100
43/43 [==============================] - 16s 369ms/step - det_loss: 0.3133 - cls_loss: 0.1849 - box_loss: 0.0026
Epoch 99/100
43/43 [==============================] - 15s 353ms/step - det_loss: 0.3130 - cls_loss: 0.1956 - box_loss: 0.0023
```

Figure 6-2: Model Training

Now the model is exported in tflite extension format because we want to deploy it on computationally cheaper devices whose size is 4.23 MB.

6.1.8 OpenCV

When our model is trained and ready, we implement the model using opencv. Using its rectangle, putText methods we draw the predicted values from the model like bounding boxes and text in the image.

6.1.9 AWS EC2

Through AWS EC2 instance, we got virtual servers i.e IaaS for deploying our model and FastAPI. IaaS (Infrastructure as a Service) is a type of cloud computing service that offers essential compute, storage and networking resources on demand on a pay-as-you-go basis. IaaS allows to bypass the cost and complexity of buying and managing physical servers and datacentre infrastructure. The server used in our project was Ubuntu Server 20.04 LTS. 8 GB storage memory, 2 CPUs and 8 GB RAM was selected as per our need. The inbound rules which control the incoming traffic to the instance were selected of type HTTP, SSH, All TCP.

6.1.10 Flutter and Dart

We are using flutter framework to develop our mobile application. Since flutter framework uses dart programming language, we can access camera and gallery to import the batter footage, fetching data from API (cricbuzz) for latest news, fetching and posting data in database for sample analysis and players profile simply by using different packages of dart/flutter available in pub.dev. Packages we have used in our project are image_picker, http and pie_chart.

- image_picker - Using this package, we import the image from device camera and gallery
- http - To make http request, we use this library
- pie_chart - For displaying the pie chart of map value pie_chart package is used.

6.1.11 SQL

For the database to store players details, analyzed shot details and sample analysis we use SQL. In our database, the entity ‘Player’ stores the attributes of player like name,

age, id, teams, playing role, etc. which are the basic information of the player. The entity ‘shot’ has its own attributes like shot_id, player_id, shot_frequency and efficiency that store the details of shot played by the batter. The player_id is the foreign key here. The relationship among entities ‘Player’ and ‘Shot’ is ‘plays’ as the batter plays shot. Similarly, the entity ‘shot’ has ‘has a’ relationship with the entity ‘shot_profile’ that stores the attributes shot_id, and shot_name. So, to store the data and analyze those values and queries we are using SQL.

```
CREATE TABLE `Player` (
  `id` int(10) NOT NULL,
  `name` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `src` varchar(300) COLLATE utf8_unicode_ci NOT NULL,
  `age` int(2) NOT NULL,
  `battingstyle` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `bowlingstyle` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `playingrole` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
```

Figure 6-3: Creating Player Table

6.1.12 PHP

PHP creates a json format API which access data of SQL and using query on it we can easily access the SQL database.

6.1.13 000webhost

To create an API and access data of SQL using PHP from a mobile application, we need a server for hosting. For hosting we have used 000webhost because it is free for limited storage and a package of flutter can easily access its domain name.

6.2 Hardware Implementation

Any normal PC is preferable for training when using Google Colab but if we want to train it on our own system, high CPU processors, high GPU or TPU and RAM are necessary. This is because deep learning must deal with lots of data and would cost us time. For the deployment, we need a mobile phone for the app that we have built to predict the shots on.

A PC with minimum specification of 4GB RAM, 3GB of memory was used to implement our model. For using mobile application, a mobile phone with 1GB RAM and 14MB of memory was used.

7. RESULT AND ANALYSIS

7.1 Shots Classification

The main objective of our project is to classify the shot played by the batsman.

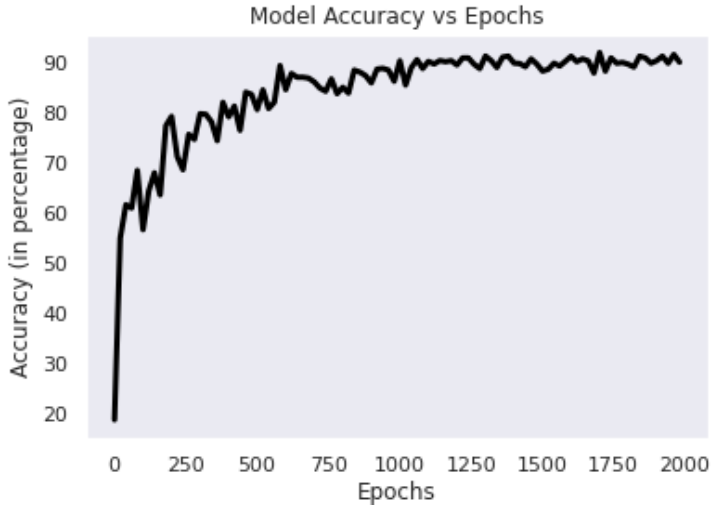


Figure 7-1: Model Accuracy vs Epochs

Training the SVC model for 2000 epochs, it was able to achieve a F1 Score of 92%.

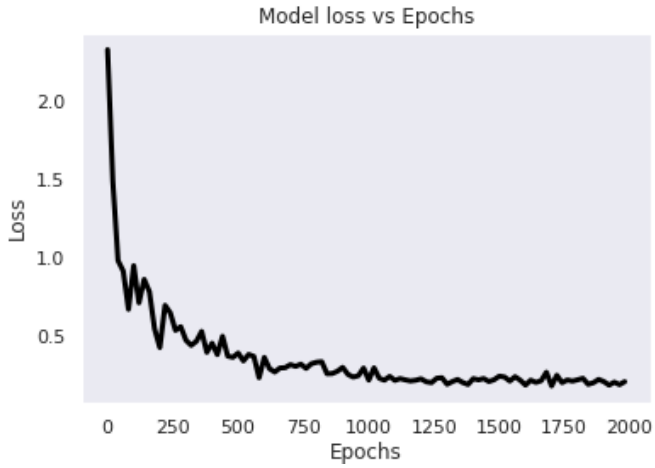


Figure 7-2: Model loss vs Epochs

The Hinge Loss observed was 0.2.

After providing the input image to the trained model, the following results were obtained:

Figure 7-3: Predicted Pull Shot [14]

Figure 7-4: Predicted Cut Shot [15]

Figure 7-5: Predicted Scoop Shot [16]

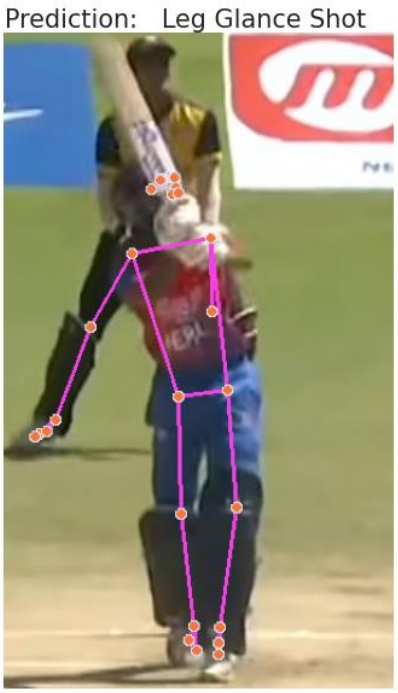


Figure 7-6: Predicted Leg Glance Shot [17]

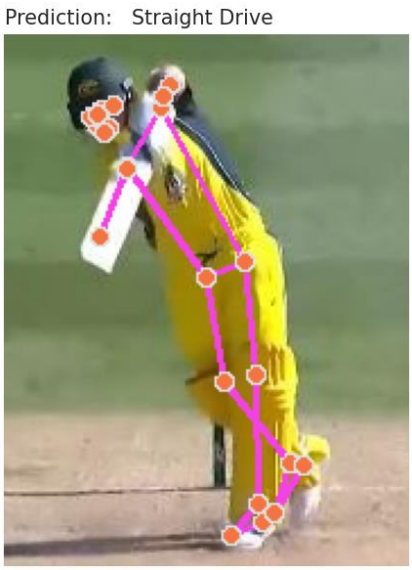
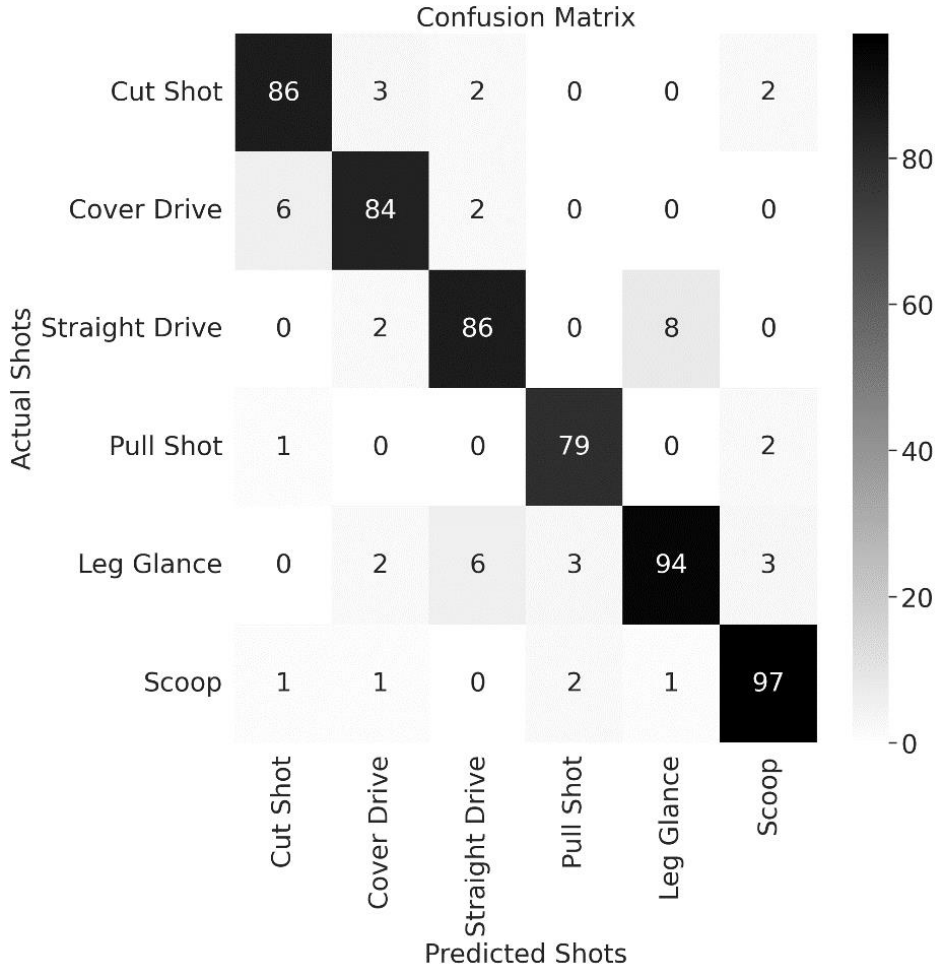


Figure 7-7: Predicted Straight Drive [18]

Figure 7-8: Predicted Cover Drive

A confusion matrix is the matrix that summarizes the predicted results and actual results. The confusion matrix of different shots is illustrated below. Every row of the matrix represents the instances of actual shots whereas every column represents the instances of predicted shots. Here, there were many shots that were predicted correctly for Scoop shot which was 97 in this case. We can also see that there were 8 shots that were predicted as leg glance but they were straight drive.

Table 7-1: Confusion Matrix



Precision talks about how precise/accurate the model is out of those predicted positive.

$$Precision = \frac{TP}{TP + FP}$$

The highest precision was obtained for Pull Shot (0.94).

Recall calculates how many of the actual positives our model captured through labeling it as positive (True Positive).

$$Recall = \frac{TP}{TP + FN}$$

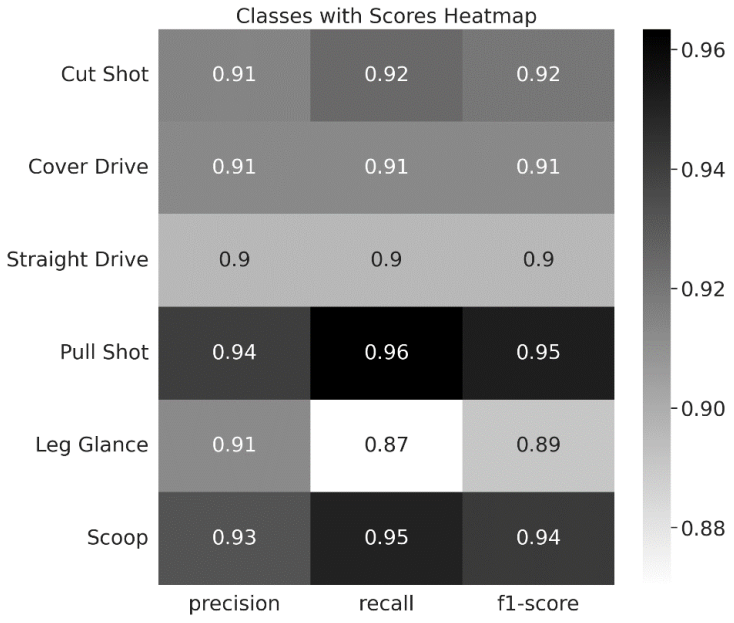
The highest recall was obtained for Pull Shot (0.96).

The F1 score is defined as the harmonic mean of precision and recall. It is a better measure to seek a balance between Precision and Recall.

$$F1 - score = \frac{2 * precision * recall}{precision + recall}$$

The pull shot had better F1 score (0.95).

Table 7-2: Evaluation Metrics



Since, the pull shot has the highest F1-Score, the model was able to classify pull shot better than any other shots whereas leg glance shot was hard to predict (0.89).

7.2 Bat Detection

Now, for the detection of bat, the loss observed for classifying the bat and detecting the bounding box is shown below:

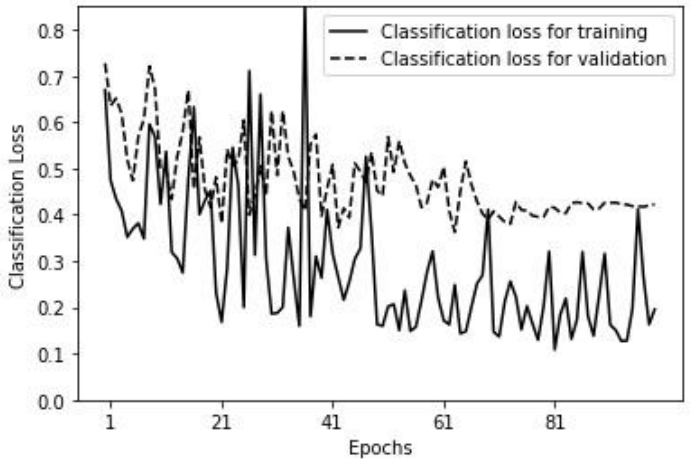


Figure 7-9: Classification Loss

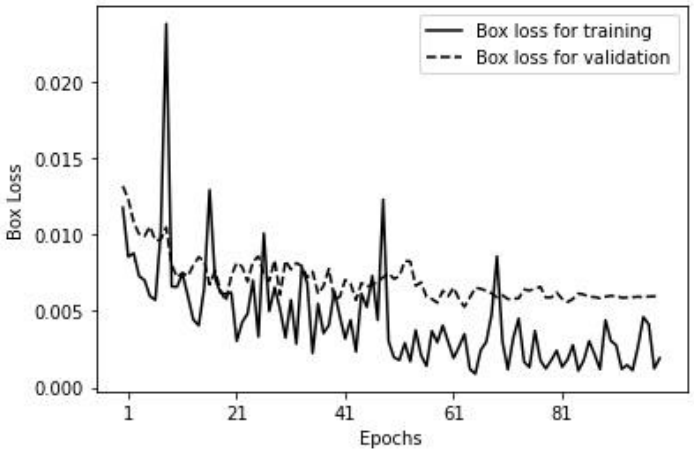


Figure 7-10: Box Loss

While testing the model with input image, the following results were obtained:

Figure 7-11: Bat Detection Far Angle [19]

Figure 7-12: Bat Detection Close Angle

Figure 7-13: Bat Detection on Focused Image [20]

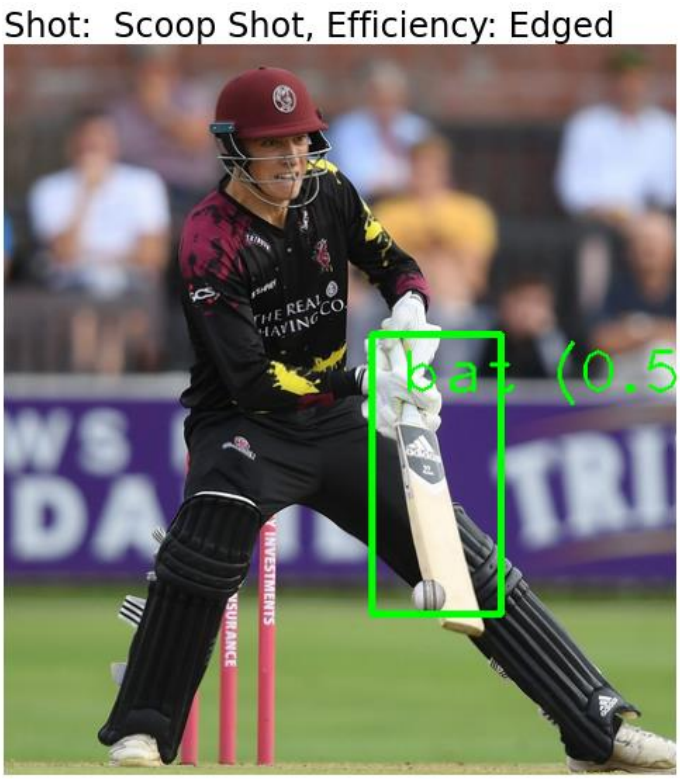


Figure 7-14: Shot and Efficiency Prediction [21]

Figure 7-15: Shot and Efficiency Prediction (Wrong Result)

As the frontend of our project, we have mobile application developed in flutter. The various screenshots of the application are shown below:

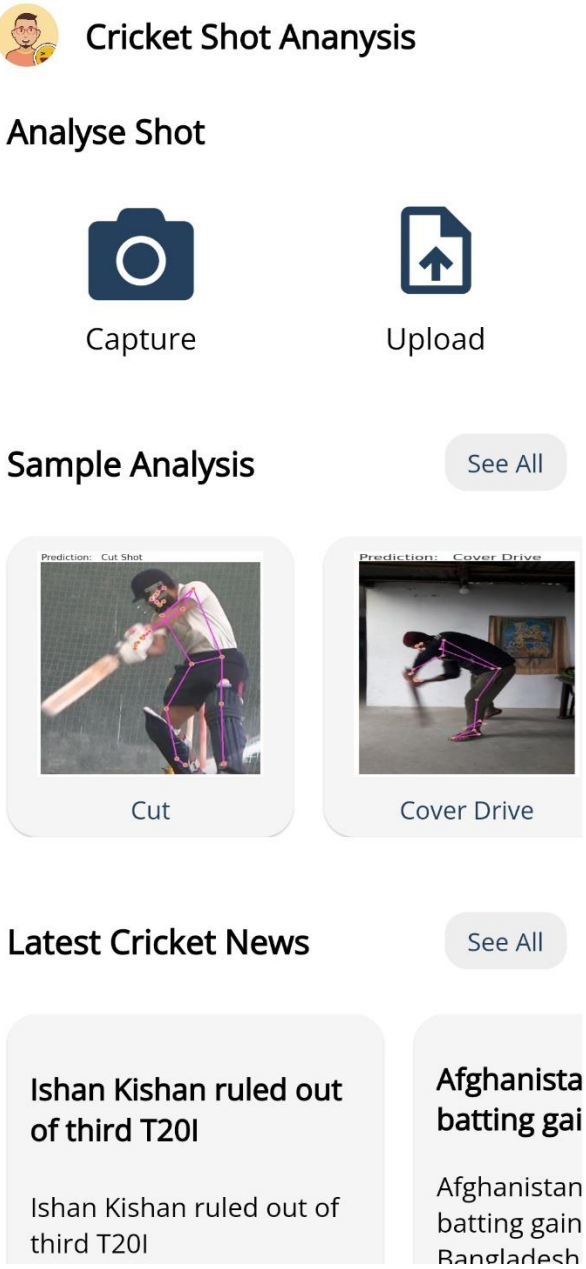


Figure 7-16: Mobile Application User Interface

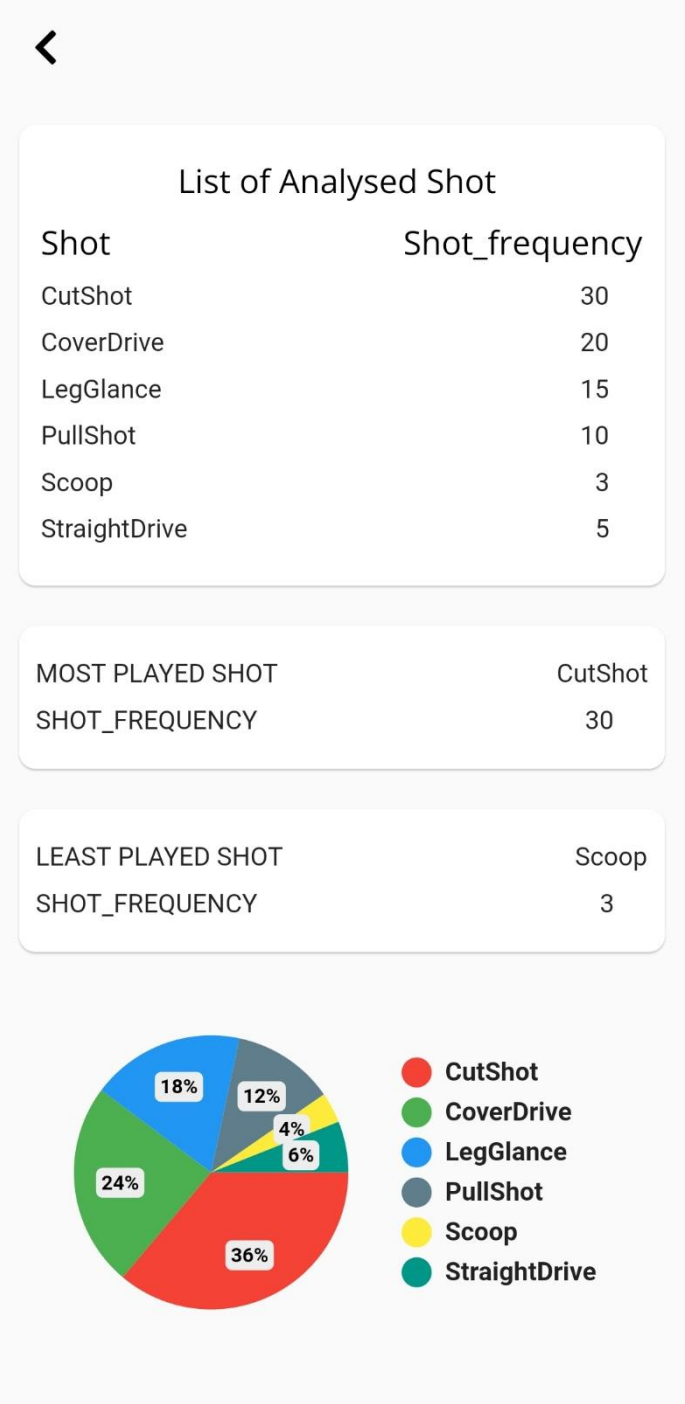


Figure 7-17: Shots Analysis in Mobile Application

Here, each player's profile was created, and their data is stored in a SQL database. The app classifies the shot played by the batter and updates the information in the database and the user interface. The app displays the number of shots played by the batter with the most played shot and a pie chart.

8. FURTHER ENHANCEMENTS

Deep learning is a new field in research and everyday research papers are published related to it. So, there are always areas to improve. The algorithms which we implemented considering it as a state of the art for our problem among others, might not be true in the coming days. So, we can enhance our model using some new techniques or architectures in the future.

The presented system(project) in its current form can classify only six cricketing shots as the trained model is unaware of other various cricket shot. This project can be further improved by increase the number and quantity of dataset including data of other shots as well so that the model can classify wider array of shots. Further using data from multiple camera angles and training the model on those additional data, the efficiency and accuracy of results can be further improved.

As of now, we can only provide images as input to this model. We know that videos are just sequential images displayed in continuous manner, so the project can also be improved to take videos as input. Detecting the frame in which the ball is about to make contact with the bat, thus detected frame can be used to classify the shot played and calculate the efficiency.

With this project sky is the limit, adding features like ball tracking we can develop this into a full-fledged cricketing software. The ball tracking features can include tracking of line of the ball, detecting where it pitches (length of the ball) and calculating the speed at which the ball is bowled. Tracking the ball through all the frames of the video, we could predict the trajectory of the ball and develop a cheaper working version of Hawkeye. Finally, in a grand scale, we could develop a complete cricket software package that could be implemented in stadiums where we could take inputs from multiple angles so we can track the batter as well as shot played so we can provide a complete Wagon Wheel and all the information of the delivery and shot played.

9. CONCLUSION

The game of cricket involves numerous planning and execution of strategies. Every professional cricket team has a cricket analyst supporting them. In this project, we purposed a system to automate some of the tasks of the analyst.

Using SVM model we were able to classify the shots into six categories (Cover Drive, Straight Drive, Scoop, Cut, Pull, Leg Glance) and the efficiency of the shot were predicted (Missed, Edged, Perfect) using EfficientDet model architecture. The mobile application was developed to make our system accessible to anyone at any time.

10. APPENDICES

10.1 Appendix A: Project Budget

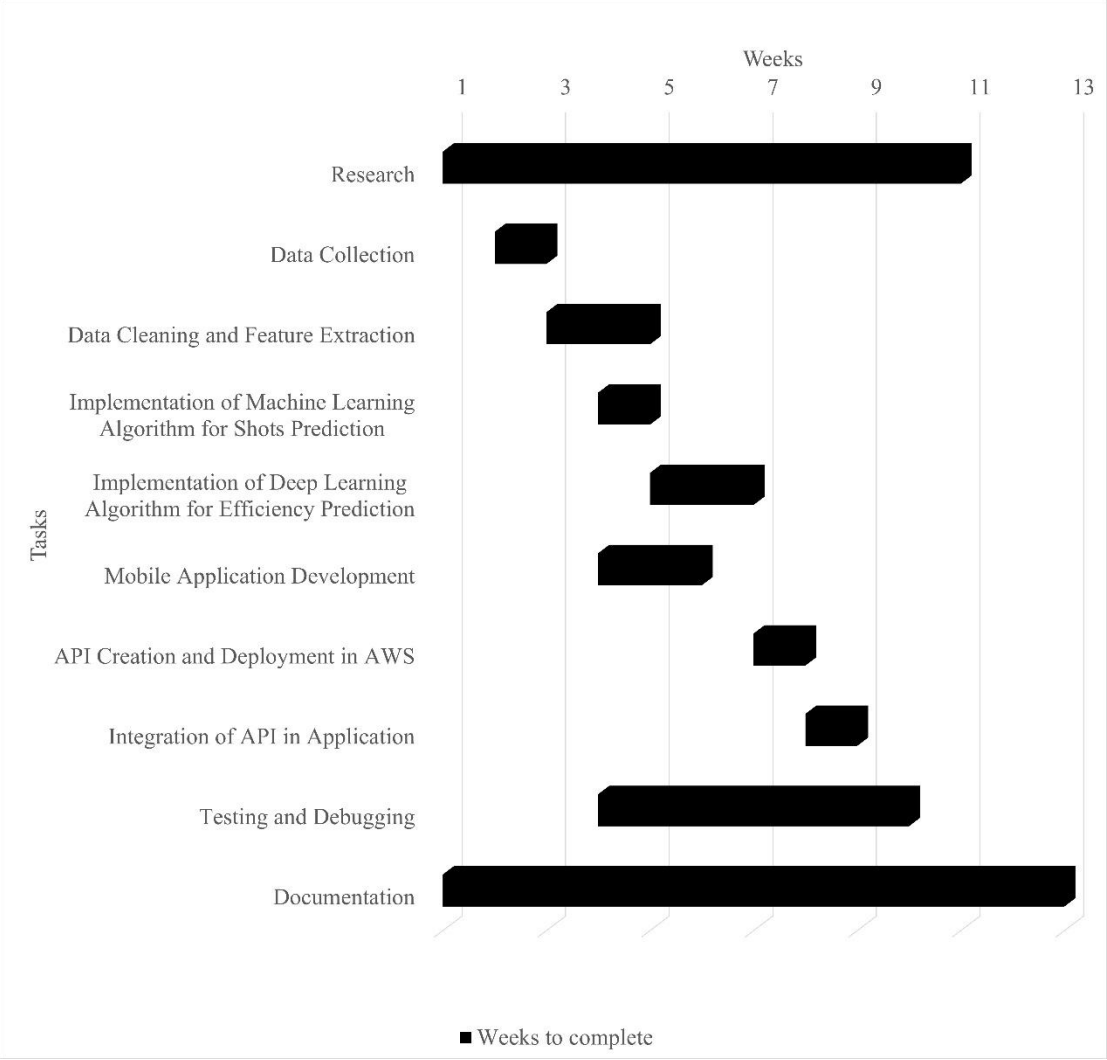
Table 10-1: Project Budget

ITEMS	PRICE (in Rs.)
AWS Computing Resources	3,000.00
Miscellaneous	500.00
Total	3,500.00

We completed the project within a budget of Rs. 3,500.

10.2 Appendix B: Project Timeline

Table 10-2: Gantt Chart



The Gantt chart above gives the visualization of our task and the time to complete it. The plan was to complete the project within two months. In the initial stage, we researched on how the project could be done by reading research papers, books, and tutorials on the internet. Data is the crucial part of this project, so we invested time for this and extracted necessary features out of it. Afterward, we tried to work on building deep learning and machine learning models aiming higher accuracy and speed. The mobile application development was carried out concurrently. The API was created and deployed using AWS. The mobile application was integrated with the API. Several testing and debugging were performed throughout the project.

10.3 Appendix C: Used Commands

- %%capture: Cell magic to suppress unwanted output
- cd data: Changes the current directory to the data directory
- mkdir fold: Creates a folder named fold
- ls: List files in current directory
- nohup python3 -m uvicorn fast_api:app –reload: Not to stop a command once it has started
- pip install package_name: To install python package package_name
- pip freeze --local > requirements.txt: To create requirements.txt file which consist of packages installed within the environment with it’s version
- virtualenv env: Creating a new virtual environment env in the current directory

10.4 Appendix D: Code Snippets

- **Feature Extraction**

```python
def create_features(image_path, target):

    data = [] # List to add columns
    idx = 0 # Index
    mpPose = mp.solutions.pose
    pose = mpPose.Pose()
    mpDraw = mp.solutions.drawing_utils # For drawing keypoints
    points = mpPose.PoseLandmark # Landmarks

    for p in points:
        body_part = str(p)[13:] # For extracting name of the body part
        data.append(body_part + "_x") # X co-ordinate
        data.append(body_part + "_y") # Y co-ordinate
        data.append(body_part + "_z") # Z co-ordinate
        data.append(body_part + "_vis") # Visibility
    data.append('target') # Target
    data = pd.DataFrame(columns = data) # DataFrame with only columns (empty)

    for img in os.listdir(image_path):
        temp = []
        img = cv2.imread(image_path + "/" + img)
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # OpenCV = BGR, Mediapipe = RGB
        results = pose.process(imgRGB) # Pose detection

        if results.pose_landmarks:
            landmarks = results.pose_landmarks.landmark
            for p in landmarks:
                temp = temp + [p.x, p.y, p.z, p.visibility] # Append x, y, z, vis of each part
            temp.append(target)
            data.loc[idx] = temp
            idx += 1
    data.to_csv(f"dataset{target}.csv", index=False) # save the data as a csv file
```

- **Training**

```python
class SVM:
    def __init__(self, learning_rate=0.001, lambda_p=0.01, epochs=2000):
        self.lr = learning_rate
        self.lambda_p = lambda_p
        self.epochs = epochs
        self.w = None
        self.b = None

    def fit(self, X, y):
        samples_size, features_size = X.shape

        y_ = np.where(y <= 0, -1, 1) # Since 1 and -1 are two classes. If 0 or less, make it -1

        self.w = np.zeros(features_size)
        self.b = 0

        for i in range(self.epochs):
            for idx, x_i in enumerate(X):
                cond_gt_one = y_[idx] * (np.dot(x_i, self.w) - self.b) >= 1
                if cond_gt_one:
                    self.w -= self.lr * (2 * self.lambda_p * self.w)
                else:
                    self.w -= self.lr * (
                        2 * self.lambda_p * self.w - np.dot(x_i, y_[idx])
                    )
                    self.b -= self.lr * y_[idx]

            if i % 100 == 0: # Print hinge loss every 100 epochs
                cost = self.lambda_p * np.linalg.norm(self.w)**2 + 1 / samples_size * \
                    np.max(np.c_[np.zeros(samples_size), 1 - y_ * (np.dot(X, self.w) - self.b)])  # hinge loss
function
                print(f"Iteration {i:<10} Loss: {cost}")

    def predict(self, X):
        approx = np.dot(X, self.w) - self.b
        return np.sign(approx) # Since 1 and -1 are the two classes
```

- **Prediction**

```python
def predict_shot(path):

    mpPose = mp.solutions.pose
    pose = mpPose.Pose()
    mpDraw = mp.solutions.drawing_utils # For drawing keypoints
    points = mpPose.PoseLandmark # Landmarks

    data = []
    img = cv2.imread(path)
    imageWidth, imageHeight = img.shape[:2]
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = pose.process(imgRGB)

    # Run this only when landmarks are detected
    if results.pose_landmarks:
        mpDraw.draw_landmarks(imgRGB, results.pose_landmarks, mpPose.POSE_CONNECTIONS,
                              mpDraw.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=2),
                              mpDraw.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2))
        landmarks = results.pose_landmarks.landmark
        for i,j in zip(points,landmarks):
            data = data + [j.x, j.y, j.z, j.visibility]
    data = [data[i] for i in idx_features]
    result = model.predict([data])[0]
    plt.figure(figsize=(10, 10))
    # Remove the text class number
    text = f"Prediction:{re.sub('[^a-zA-Z]', ' ', classes_list[result])}"
    plt.text(0, -5, text, size='xx-large', weight=500)
    plt.imshow(imgRGB)
    plt.grid(False)
    plt.axis(False)
```

- **Image upload, predict shot and efficiency in mobile application**

```
final String _fastapi ="http://ec2-3-133-156-185.us-east-2.compute.amazonaws.com:8000";
 Future predictShot({required String filePath}) async {
   try {
     String url = "$_fastapi/files/";
     var headers = {
       'accept': 'application/json',
       'Content-Type': 'multipart/form-data'
     };
     String filename = filePath.split('/').last;
     FormData formData = FormData.fromMap({
       "file": await MultipartFile.fromFile(
         filePath,
         filename: filename,
         contentType: MediaType('image', 'jpeg'),
       ),
       "type": 'image/jpeg'
     });
     Response response = await dio.post(url,
         data: formData,
         options: Options(
           headers: headers,
         ));
     if (response.statusCode == 200) {
       return response.data;
     } else {
       return null;
     }
   } catch (e) {
     return null;
   }
 }
```

REFERENCES

[1] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET),* Antalya, 2017, pp. 1-6

[2] D. Karmaker, A. Chowdhury, M. Miah, M. Imran and M. Rahman, "Cricket shot classification using motion vector", *2015 Second International Conference on Computing Technology and Information Management (ICCTIM),* 2015. Available: 10.1109/icctim.2015.7224605 [Accessed 11 December 2021].

[3] M. Foysal, M. Islam, A. Karim and N. Neehal, "Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots", *Communications in Computer and Information Science*, pp. 111-120, 2019. Available: 10.1007/978-981-13-9181-1_10 [Accessed 13 December 2021].

[4] M. Tan, R. Pang and Q. Le, "EfficientDet: Scalable and Efficient Object Detection", *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. Available: 10.1109/cvpr42600.2020.01079 [Accessed 21 January 2022].

[5] Python.org. 2021. Welcome to Python.Org. [online] Available at: <https://www.python.org/> [Accessed 14 December 2021].

[6] TensorFlow. 2021. Tensorflow. [online] Available at: <https://www.tensorflow.org/> [Accessed 14 December 2021].

[7] H. Partnership, "OpenCV", Opencv.org, 2021. [Online]. Available: https://opencv.org/. [Accessed: 15- Dec- 2021].

[8] Tutorialspoint.com. 2021. SQL Tutorial - Tutorialspoint. [online] Available at: <https://www.tutorialspoint.com/sql/index.htm> [Accessed 25 January 2022].

[9] "Amazon Web Services", aws.amazon.com, 2022. [Online]. Available: https://aws.amazon.com/. [Accessed: 17- Feb- 2022].

[10] "FastAPI", Fastapi.tiangolo.com, 2022. [Online]. Available: https://fastapi.tiangolo.com/. [Accessed: 15- Feb- 2022].

[11] Mediapipe, 2020. *Pose Landmarks*. [image] Available at: < https://google.github.io/mediapipe/solutions/pose.html/> [Accessed 5 January 2022].

[12] LabelImg, 2020. *LabelImg User Interface*. [image] Available at: < https://github.com/tzutalin/labelImg/> [Accessed 1 January 2022].

[13] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" *36th International Conference on Machine Learning* [Accessed 21 January 2022].

[14] HamroKhelKud.com, 2021. *Pull Shot*. [image] Available at: <https://hamrokhelkud.com/> [Accessed 26 January 2022].

[15] CricketingNepal.com, 2021. *Cut Shot Shot*. [image] Available at: <https://cricketingnepal.com/> [Accessed 22 January 2022].

[16] CricFit, 2021. *Scoop Shot*. [image] Available at: <https://cricfit.com/> [Accessed 10 January 2022].

[17] Cricket Association of Nepal, 2021. *Leg Glance Shot*. [image] Available at: <https://cricketnepal.org.np/> [Accessed 21 January 2022].

[18] Cricket Australia, 2021. *Straight Drive Shot*. [image] Available at: <https://www.cricketaustralia.com.au/> [Accessed 20 January 2022].

[19] GameOn Esports, 2020. *Test Cricket*. [image] Available at: < https://gameon-esports.com/live-each-moment-from-the-complement-cricket-news.html/> [Accessed 29 January 2022].

[20] The National, 2019. *Paras Khadka at ICC Global Academy in Dubai*. [image] Available at: <https://www.thenationalnews.com/sport/cricket/paras-khadka-becomes-first-nepal-batsman-to-score-t20-international-century-1.916481/>[Accessed 30 January 2022].

[21] "Tom Banton Masterclass: The Scoop Shot - Somerset County Cricket Club", *Somerset County Cricket Club*, 2022. [Online]. Available: https://www.somersetcountycc.co.uk/news/first-xi/tom-banton-masterclass-the-scoop-shot/. [Accessed: 25- Feb- 2022]