

CLASSIFICATION OF SHOT SELECTION BY BATSMAN IN CRICKET MATCHES USING DEEP NEURAL NETWORK

by

Afsana Khan

18101464

Fariha Haque Nabila

18101457

Masud Mohiuddin

18101052

Mahadi Mollah

19101040

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2022

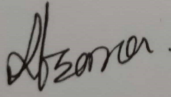
© 2022. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



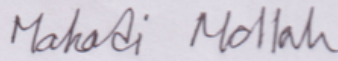
Afsana Khan
18101464

Nabila

Fariha Haque Nabila
18101457



Masud Mohiuddin
18101052



Mahadi Mollah
19101040

Approval

The thesis/project titled “CLASSIFICATION OF SHOT SELECTION BY BATSMAN IN CRICKET MATCHES USING DEEP NEURAL NETWORK” submitted by

1. Afsana Khan (18101464)
2. Fariha Haque Nabila (18101457)
3. Masud Mohiuddin (18101052)
4. Mahadi Mollah (19101040)

Of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 29, 2022.

Examining Committee:

Supervisor:
(Member)



Md. Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Md. Ashraf Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Machine learning (ML) is such a field that focuses on learning based method. It basically leverage data to improve the performance on particular tasks. It creates a model based on training data and makes prediction according to the pattern what it has learnt. Machine learning can be used to classify a certain category of image as it has a successful contribution in image processing. That's why we have used machine learning approach to implement our proposal. Our proposal is basically classification based. As we know cricket is a very popular game in our country. Technological advancement has brought a tremendous change in field of cricket. Such as, projected score prediction, wicket prediction, winning probability, run rate as well as shot detection also it has benefitted the decision making system a lot. Our primary objective is to use Machine learning in the field of Cricket, where we aim to classify the tentative shot selection of batsman. Our primary goal is to automate the broadcast system where cameras can move automatically by identifying the shots and the direction of the shots. As sometimes the shots are delivered so fast, crucial moments can be missed due to lack of fast telecast system. For implementing our proposed model, we have generated our own dataset named "CrickShots" by taking real time photos from various cricket matches. We collected 1800 images of batsman while delivering the shots or to be more specific we have tried to take pictures of the connection moment of the bat and ball. To have an accurate result of classification we have used 'VGG-16' model and 'Inception'. Where we got a better result by using VGG-16. We have used 85% of the total images to train the model first and 15% later on to test the model. The images had to go through several pre-processing methods such as background removal and scaling to be prepared for training the model. At last we got desired accuracy of 95% from VGG-16 and 85% from Inception.

Keywords: Cricket, Batsman, Shot, Camera, Autonomous, Broadcasting, VGG-16, Inception.

Acknowledgement

Firstly, Alhamdulillah for everything. All the praises to the almighty Allah because of whom we have completed our thesis without any major problem.

Secondly, A warm thanks to our Advisor Md Tanzim Reza sir and co-advisor Md Ashrafal Alam, Ph.D sir for their kind and immense support and advice, without their support we may not be able to complete our thesis.

Finally, Thanks to our parents, without their prayer and support we cannot come to the end of our journey in the thesis.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Nomenclature	ix
1 Introduction	1
1.1 The background behind the model	1
1.2 Motive and Objectives	1
1.3 Research Objective	2
2 Problem Statement	3
2.1 Problem Statement	3
3 Literature Review	4
3.1 Literature Review	4
4 Work Plan	8
4.1 Work Plan	8
4.2 Methodology	9
5 Dataset	14
5.1 Dataset	14
5.2 Data pre-processing:	15
5.3 Train test split	16
6 Model Implementation	18
6.1 Workflow Overview	18
6.2 Result	19
7 Future Work	25

8 Conclusion	26
8.1 Conclusion	26
Bibliography	28

List of Figures

4.1	Flowchart of the proposed shot detection model	8
4.2	VGG-16 Network Architecture	11
4.3	Inception V3 Model	12
4.4	Inception working method	13
4.5	Mask R-CNN working architecture	13
5.1	A subset of the dataset collected for training and test	14
5.2	A subset of the dataset with and without background	16
6.1	Confusion matrix of VGG-16	19
6.2	Confusion matrix of Inception-V3	20
6.3	Model Accuracy Graph of VGG-16	21
6.4	Model Loss Graph of VGG-16	21
6.5	Model Accuracy Graph of Inception	22
6.6	Model Loss Graph of Inception	22
6.7	VGG-16 vs Inception-v3 Accuracy comparison	23

List of Tables

6.1	Comparison Table of VGG-16 and Inception-v3	23
6.2	Comparison Table of VGG-16 and Inception-v3 and old paper	24

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CAD Computer-Aided Diseases

CNN Convolutional Neural Network

FPD Fast Pose Distillation

ICC International Cricket Council

ML Machine Learning

MPPE Multiple People Pose Estimation

ODI One Day International

R – CNN Region- based Convolutional Neural Network

SPPE Single People Pose Estimation

T20 Twenty Twenty

VGG Visual Geometric Group

Chapter 1

Introduction

1.1 The background behind the model

As humans we always search for peace and entertainment in this stressful world. One form of entertainment is various kinds of sports. Nowadays, cricket is considered the second most famous athletic game in the world. Around 106 countries currently play in this form of game. To make this form of game more lucrative the broadcasting system has been included in it. Most of the time Broadcasting machines are being operated by humans. Additionally, sometimes human operators cannot detect and capture the exact moment after a ball has been hit by a batsman. As a result, sometimes the exact location of the ball cannot be found in camera in very crucial moments of the game. To solve this issue, we came across this idea of classification of shot selection by a batsman using a neural network where we use various models like VGG-16 and inception-v3 to train our image which was taken from highlights of live cricket matches. Our model will automate the broadcasting as well as capture crucial moments of the game which will eventually make the game of cricket more attractive, enjoyable, and telecast friendly.

1.2 Motive and Objectives

With the rapid advancement of technology in recent era, acquiring information on the inside and out has become very straightforward. As a result of the accessibility of live as well as archived data, Deep Learning is rapidly becoming a prominent trend in sports analysis. In context of Cricket, it has shown tremendous success in projected score prediction, wicket prediction, winning probability, run rate as well as shot detection.

Cricket is one of the sports that has benefited the most from technological advancements. Using technology, the cricket officials like umpire can know if his decision are correct or not in a particular moment. Furthermore, Drones and Spidercam assist the officials to broadcast the game and make it more enjoyable for the fans. However, Drones and Spidercam are controlled by humans right now but if we can further proceed with our research then this tool can autonomously sense the shot of the batsman and move their camera toward that direction.

Our aim is to make a model from which we can classify the shot selection of a batsman by using deep neural networks. Moreover, we are taking a lot of images of different classes of shots and making it understandable by models from which we can

test our model to see how accurately it can classify any shots played by a batsman.

1.3 Research Objective

As we know cricket is a part of emotion for most of the nation including us. The comparative advancement of technology in the field of cricket played an essential role to hold this love and compassion together. As it is already in top of the topic among game lovers and being cricket lovers we also decided to work on it to improve the automation system and make the game more enjoyable for the viewers.

1. Understanding the importance of full automation in cricket.
2. Understanding the pros and cons of our proposed model and associated work.
3. Understanding the accuracy of our research result and analysis it for further work.
4. Understanding the impact of deep learning in our proposed paper.
5. Constructing the fast shot detection system using our algorithm.
6. To provide guidance for enhancing the research for the future.

Chapter 2

Problem Statement

2.1 Problem Statement

Cricket has around 2.5 Billion fans and with this fan following it ranks second among all athletic games in the world. There are more than 100 countries associated with this game. The first cricket match that got recognition as the 1st international game was played by the USA and Canada which was held in 1844[1]. This emerging sector of game already brought the attention of tech entrepreneur and they introduce new tool to make it better day by day. In terms of broadcasting it is essential that shots should be clear and zoomed when necessary.

Cricket has not yet reached its peak on its technology advancement side. In cricket most of the works are handled by the officials like checking umpire review, filming the whole match to broadcast etc. Sometimes it becomes time consuming to check umpire review and the result can have flaw compare to machine. Moreover, sometimes the cameraman cannot sense which shot the batsman will play so they miss the position of the ball after the batsman hit the ball due to the lack of staying focus on one thing for a long time. Therefore, it becomes really frustrating for the fans if they miss any important shot in crucial moment.

The current situation can be improved by making the broadcasting system fully automated. Moreover, if we introduce automated system we can easily minimize the error rate successfully which was previously made by human. Our research purpose is to reduce the error by estimating pose via machine which will able to generate clear view of shots during the live session of the match without any delay and also by implementing this the viewers will not miss the crucial moment of the match.

Chapter 3

Literature Review

3.1 Literature Review

Zhang, F., Zhu, X., Ye, M. [2] asserts that current Human Pose Estimation methods frequently look into how to improve the performance of the model instead of enhancing the efficiency. This results in cost-effective and poor scalability models. After evaluating the pose estimation models they introduced a new ‘Fast Pose Distillation (FPD)’ algorithm learning scheme in their research. The FPD model is qualified to execute swiftly with moderate computational cost. A strong teacher network is used to achieve this result. Using datasets MPII Human Pose and Leeds Sports Pose they showed their models superiority over other models. Kowsher et al., [3] affirms that, in the meantime Cricket is considered as one of the most engrossing sports in the world, mainly in South Asia. Since human beings are likely to make mistakes, in most of the matches the umpire makes some mistakes and the third umpire assists him to rectify the mistake. The use of computer vision and artificial intelligence in cricket analysis and decision making have become exoteric recently. Using CNN along with Inception V3 they made a system to autonomously make the decision of the third umpire and keep record of the score.

Islam et al., [4] using CNN model points out the bowling action with 93.3% accuracy. The research used about 8100 images of 18 different bowlers according to their bowling action, where 80% of the images were for training purposes and 20% for test data. The research shows that, along with the VGG16 model and a number of layers on head of it, pre-trained data gives the best result. According to this paper [5] with the advancement of machine learning and pose estimation through computer vision is certifying its importance in the future. Some of the important findings of this article are (i) Human pose estimation has helped the gym freaks to move their gym environment to their home for workout. Using the pose estimation, the trainee can give instructions if the posture is correct or not. (ii) Augmented reality has influenced the digital era decently. Using augmented reality and computer vision virtual objects are being created in real life as well as accurately tracing an object in real-life. (iii) Pose estimation in gaming and animation already show huge impact. Couple of years ago game developers had to create the characters on their own and had to make them animated but due to pose estimation the characters became automating animation as computer vision has the capability to capture motion in real-time. (iv) Most of the top companies are shifting from human labor based to

robot labor based. As humans have some limitations and robots can work for a long span of period without facing any problem. Robots became an exact alternative to humans for the duties but robots also showed some limitations like moving from one place to another. But this problem was solved using pose estimation as robots can now respond to environmental changes on their own. (v) At the moment, mobile phones are also manufactured with built-in pose estimation as one can count foot tracks using ‘crowd tracking’. Article [6] proclaims that human pose estimation can be of two types: Single-person (SPPE) and Multi-person pose estimation (MPPE). However, SPPE is much easier than MPPE as in MPPE we have to deal with multi-person inter-collusion. In each pose estimation algorithm allows upon a body model ahead of time. It enables the algorithm to abstract the problem of estimating body model parameters into that of predicting human posture estimates. The ultimate result of most algorithms is a simple N-joint stiff kinematic skeleton model (N is considered in the middle of 13 and 30).

A present day paper [7] came across some overlooking facts like background effect, color, dress, skin tone and many other tough challenges. This overlooking fact could play a vital role in sign language recognition and other pose styles as well as medical application. Firstly, a deep learning method with the ability to constitute the human body in various models will help to determine different parts of the body and spatial correlation between them. In order to detect the hand, the geometrical details and shape of the hand will be taken from hand contour. Secondly, for color image segmentation problems an unsupervised adaptive method formed on the Voronoi region is mainly used. Due to the small joints of the body part often the identification becomes very tough. Finally, by using box based model pose estimation, optical flow tracking algorithm and Voronoi segmentation this paper reached its desired position including successful use of depending on hand orientation and structure feature and the scope of this work will expand in the future to encompass the development of deep-structure form of models to boxes coupled with the shifting of various body components. This can be accomplished with the help of correlation-based quantification of moving body parts. Another application of this paper is extracted HOG features for the above model. To revamp the accuracy and achieve superior outcomes in the area, this type of feature extraction could be carried out utilizing a neural network-based methodology. As applications focus on pose estimation especially so for videos Convolutional Neural Network (CNN) is a very good approach in order to classify the pose.

Paper [8] about human stance detection regarding cheating in exams has shown some great work. In this paper a single board computer (Jetson Nano) will allow proctor to control students during exams with this device. The device’s video stream was recorded by a web application that allowed the proctor to view it and by processing the image in real time the proctor can catch the culprit without any delay. After training and testing the data set the outcome of this paper was remarkable. NVidia’s Jetson Nano and the case-mounted Jetson Nano AI Camera are used in the prototype assistive monitoring device. The device records a video feed that is transferred to a computer over Wi-Fi for monitoring and cheating detection and the system’s performance was examined using a validation dataset of 30 photographs that were generated and labeled by legitimate proctors. The system has 90% accu-

racy, an f1-score of 89.65%, and an AUROC of 90.32%.

Ding, W., Hu, B., Liu, H., Wang, X., Huang, X. [9] Reveal that, in the field of human-computer interaction, the use of skeletal data for human posture detection is a major topic. Initially, in this paper angle features and distance features are defined in a 219-dimensional vector. Particularly, the aspect and length features are fixed in terms of the specific correlation betwixt joints and global structural spot of joints. Using the rule method along with Bagging and random subspace method creates different samples and features for better classification performance. The research finally shows that standard machine learning methods and CNNs have lower estimation accuracy than rule-based learning methods. Samad et al., [10] describes that, the Kinect sensor which uses infrared camera to process a depth image is fit to be allowed to work in some surroundings that have rapidly changing lighting conditions. Furthermore, this research verifies that in various scenarios human detection rate is very apex using this method. Also, the experiment's overall body proportions demonstrate that the accuracy is just about to the golden ratio gain used in this research. However, this method is still inadequate to determine when an individual is physically tied to an object, such as if they are laying down on the ground or leaning in opposition to a wall.

Kress et al., [11] depict pose evolution in real life traffic scenarios. In this paper pose estimation detection done from a moving automobile for roadway traffic application. A dedicated dataset which records real traffic contents including cyclist and pedestrian is allowed a diverse and realistic application. Firstly, an intersection with a wide angle phonograph camera method was used to take measurements for 3D pose with labeled data. Moreover, the accuracy of 2D postures computed in traffic scenes is comparable to state-of-the-art outcomes from other datasets. Furthermore, in inconsistency maps acquired by a phonograph camera, 3D pose estimate formed on unique images exceeds a naive length measurement of single joints, and the achieved accuracy. Finally, the result shows very good accuracy with a PCKh average for = 0:5 of 94.64% for cyclists and 87.89% for pedestrians. Though the research shows bicycles make more errors than pedestrians and that their performance declines as distance increases it will be completely applicable for the transport sector and future purpose.

Paper [12] regarding pose estimation states about monocular footage from TV sports. The development of presumption algorithms and probabilistic models based on learning assessment has been the focus of this problem's research. Without any formal learning on motion collected data, the research proposes a bilateral model-based generative methodology to predict the person's position in 2D from improperly calibrated monocular video in unrestrained sports TV footage. Moreover, with some human interaction and single view point the researcher shows the method which they use in it. Furthermore, the algorithm was used to test the three difficult sports sequences of various sports. Finally, the findings reveal that the suggested framework for generative pose detection is efficient enough to work even in the most difficult unrestrained situations of estimating pose. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton [13] explains that image detection or classification is a very complex task as it cannot be specified with a large amount of dataset like ImageNet. But

CNN has the prior knowledge to compensate for the lacking's as it has models like VGG-16, VGG-19 etc.

Chapter 4

Work Plan

4.1 Work Plan

The purpose of our paper is classifying the shots of the batsman . We tend to train our model with a good amount of training dataset so that we can get our desired accuracy. Firstly, we prepared our dataset. In other words, we pre-processed the dataset for removing unclear and blurry photos to avoid biased results. The work plan diagram is given below-

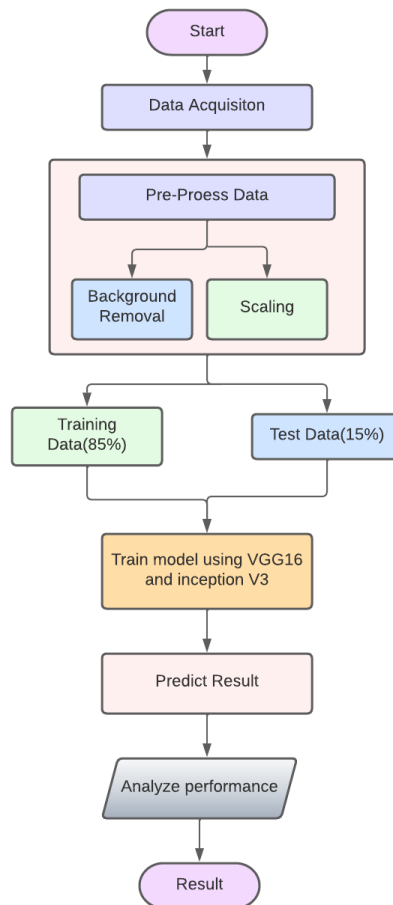


Figure 4.1: Flowchart of the proposed shot detection model

- i) Pre-processing: We have selected clear and high resolution photos and also avoided unclear and blurry photos while choosing our dataset. In our data pre-processing to make our dataset more precise and get enhance accuracy we have used these steps-
- Scaling: In order to fit our dataset in the model we have scaled our raw data size and drive it between 0 and 1 from 0 to 255.
 - Augmentation: We have used 10 percent of the width shift in our data. For width shift floating point number is used and the range is between 0.0 to 0.1.
 - Background Removal: We have used Mask R-CNN in our dataset to remove the background of the image and detect batsman as an object class.
- ii) Splitting: We have split our dataset into two parts. The first part contains 85% of data which is split for the purpose of training. And the remaining dataset are for the purpose of testing data.
- iii) Classification: The training dataset is then passed into the CNN based models VGG-16 and Inception-v3 where data is trained and converged accordingly.
- iv) Result: Here the test dataset is fed to the model and then we can see the accuracy of our proposed model.

4.2 Methodology

In order to fulfill complete research at first, we have chosen our desired topic which was “shot classification using neural network” as this topic is comparatively unique so we decided to choose this topic as we have lot of opportunity in this field for future works.

In our research we have used CNN as our model. We have used CNN for our classification problem because CNN can ensure the high accuracy.

CNN (Convolutional Neural Network):

As our research is focused on picture categorization, we’ve chosen to employ a Convolutional Neural Network (CNN). Lately, CNN based models are dominating the computer vision space for image classification and object detection [14]. An input layer, an output layer, and numerous hidden layers are included in CNN models. Image categorization is the process of segmenting pictures into distinct groups based on their characteristics. The edges of a picture, pixel intensity, pixel value changes, and other features are all examples of features [15]. However, these characteristics differ from one image to the next. The ambiguity of these features is the most difficult aspect of working with images. By evaluating the labeled pixel values, we can determine the desired pattern. However, finding patterns just by observing pixel values, on the other hand, is tough. As a result, we may require a more advanced technique to detect these edges or to discover the underlying pattern of various components of the image that may be utilized to label or categorize these photographs. This is where a more sophisticated strategy, such as CNN, comes into play. CNN stands for convolutional neural network, and it is a sort of deep learning neural network. Consider CNN to be a machine learning system that can take an input image, assign

meaning to distinct parts or objects in the image and distinguish between them in a nutshell. The role of CNN is to compress images into a format that's easier to handle while preserving important properties for accurate prediction. Each convolutional layer in this technique contains certain filters to apply to a picture. The image becomes a filtered image after this filtration, and the technique is known as convolution. When the tensors are maximum pooled and the output is subsampled, a smaller image is returned. In our case, after pre-processing, the dataset is split into training and test data. The 85% training dataset is eventually utilized as a trainer in the CNN method and the remaining 15% data is used as test dataset. It is used to test the CNN once it has been trained.

VGG-16:

VGG is nothing but the convolutional neural network architecture which was first introduced in 2014. It is also known as VGGnet. Basically, VGG refers to a visual geometric group. The image's input form is $224 * 224 * 3$, with a fixed $3*3$ filter size and a maximum pooling layer size of 5 and a size of $2*2$ across the network [16]. VGG, a convolutional neural network architecture, contains multiple layers. VGG has two types among them VGG-16 contains 16 layers on the other hand VGG-19 deals with 19 layers [17]. Among various kinds of image recognition architecture, VGG is widely used. A. Zisserman and K. Simonyan from Oxford University were the first to present VGG. After that, it exhibits a fantastic result of 92.7 percent accuracy, which is among the top 5 accuracy in imageNet, which is a dataset of 14 million photos separated into 1000 classes. This image net dataset comprises pictures with a fixed size of $224*224$ pixels with RGB channels. So we have a tensor flow of as input $(224,224,3)$. This Models take the input which was taken as an image and after processing it give us output as a vector of 1000 values. For example:

$$\hat{y} = \begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \hat{y}_{999} \end{bmatrix}$$

This vector contains classification probability for corresponding classes. It contains 1000 classification probability. Suppose we have output layer 1.3 and we pass it through the softmax activation function which basically converts the vector number into a probability vector and the probability will be 0.02. Like this for 5.1, 2.2, 0.7, 1.1 the corresponding probabilities will be 0.90, 0.05, 0.01, 0.02.

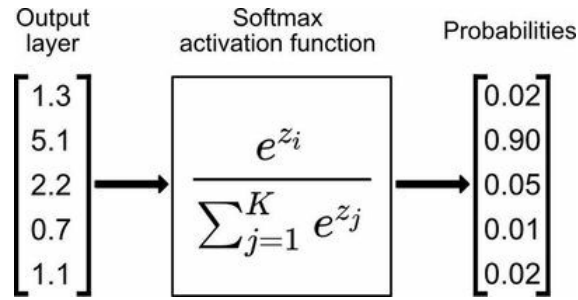
Here,

$\sigma = softmax$

$z = vectorinput$

$K = classcount$

$e^{z^i} = vectorinputexp.function$



$e^{z^j} = \text{vectoroutputexp.Function}$

After the operation we can calculate the ground truth vector from given classes and from that ground truth vector we can also calculate the loss function.

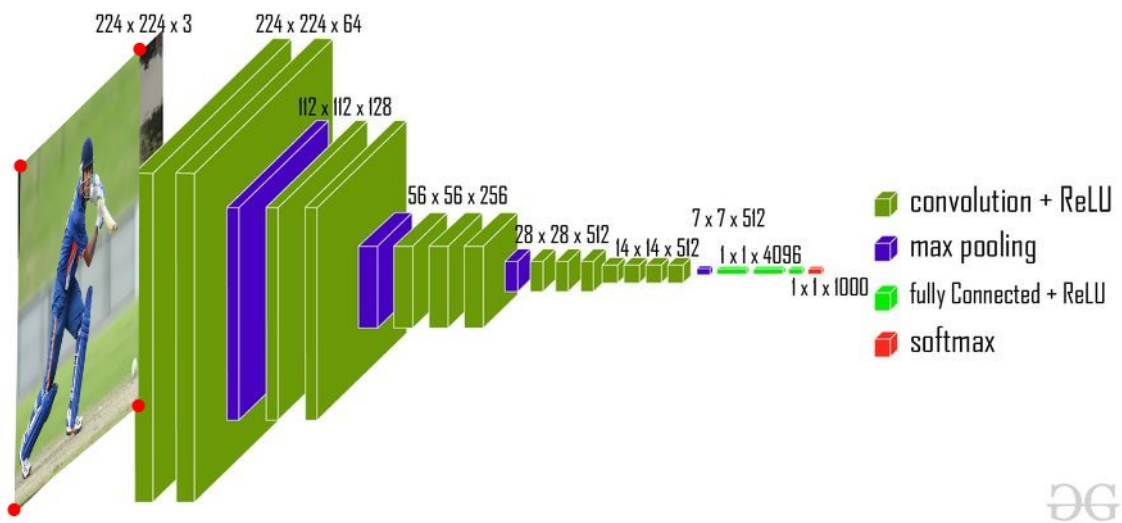


Figure 4.2: VGG-16 Network Architecture

We have used VGG for different image classification problems. Though we can use VGG for different purposes it has few disadvantages as well. One of its disadvantages is depth and multiple fully connected nodes which requires a lot of memory spaces. As a result, the model works a bit slow for training the data set, but we still use this model because of its easy implementation in image classification.

Inception:

Inception is a convolutional neural network. This convolutional neural network is mostly used for object recognition and picture analysis. It was initially released as a GoogLeNet module. The development of CNN classifiers relies heavily on Inception. Inception v3 is an image recognition model that has been shown to attain greater than 78.1 percent accuracy on the ImageNet dataset. Inception v1 and Inception v2 are two variants of this network. Inception 2 and Inception 3 are two sequels to the original Inception. Inception v4 and Inception-ResNet are two different versions of the same software. For our picture categorization, we employed Inception V3. For reducing computational power, Inception V3 was released. This enhances processing performance by reducing the amount of parameters in a network. It also

keeps an eye on the network's efficiency. Inception V3 usually reduces the parameter of networked. The process usually happens by decomposing a large convolution to a small one by decomposing it which actually reduces parameter number and computational expense. Moreover, it replaces the size of the network and makes it smaller. Furthermore, the training speed of this model is up to the mark. The 1×1 convolutional kernel makes this model faster in case of training speed [18]. This 1×1 convolutional kernel also helps to reduce features channel numbers. We have used inception V3 for our image classification. We have taken 1800 photos and divided them into 10 classes then we have trained our data and as well as we tested it and after testing we have achieved 85% of the accuracy. As V3 has deeper network compared to V1 and V2 and also maintained its speed this model helps a lot in case of efficiency and computational expense for our image classifications. For this reason, we have used this outstanding model for our image classifications.

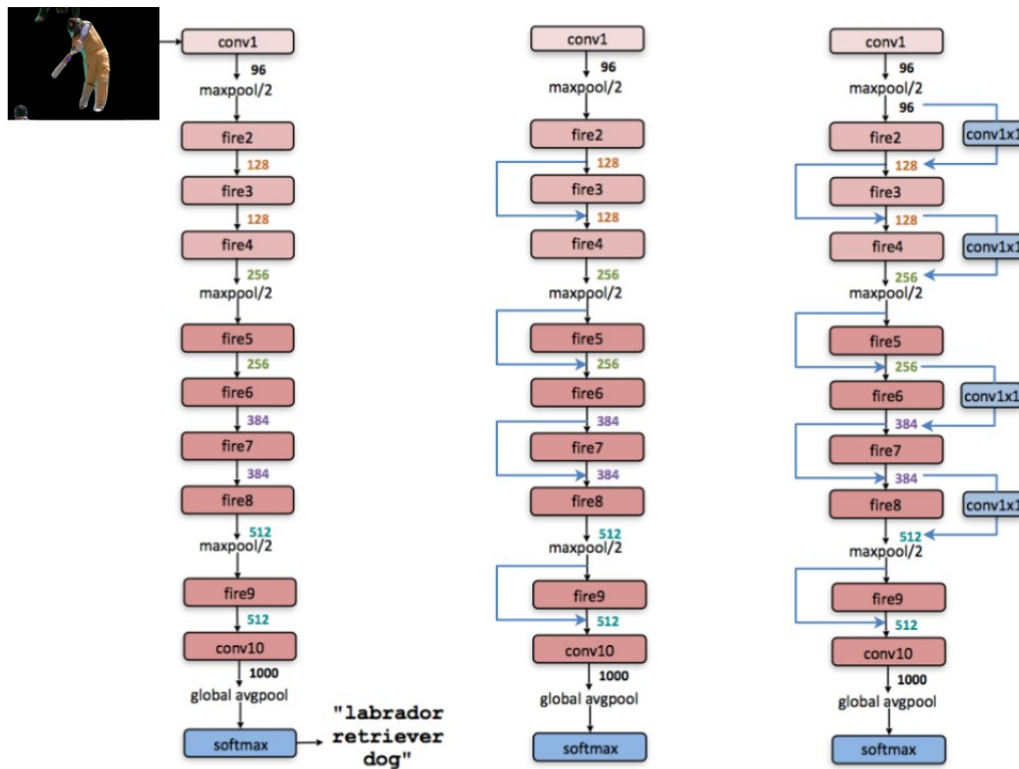


Figure 4.3: Inception V3 Model

Inception is a neural network which deals with a huge number of arrays of images. It also brings lots of variations in the feature image content. These tons of images really need to be designed correctly.

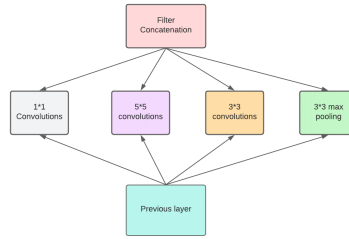


Figure 4.4: Inception working method

Mask R-CNN:

Object detection is a computer vision approach that detects the presence of objects in an image or video. In 2017, a group of AI researchers created the 'Mask R-CNN' model to recognize objects using deep convolutional neural networks. Detecting objects is a challenging problem since the model must categorize the proper object class. This model was created with the Python programming language. Before the invention of Mask R-CNN, the CNN family already had three object detection models: R-CNN, Fast R-CNN, and Faster R-CNN. Mask R-CNN is based on the Faster R-CNN model and can recognize objects as well as segment instances pixel by pixel. Instance segmentation is a challenging task as it requires to correctly detect every object in an image as well as segment every instance [19]. This model is different from its predecessors in terms of outputting the object mask, the mask is the spatial layout of the input object.

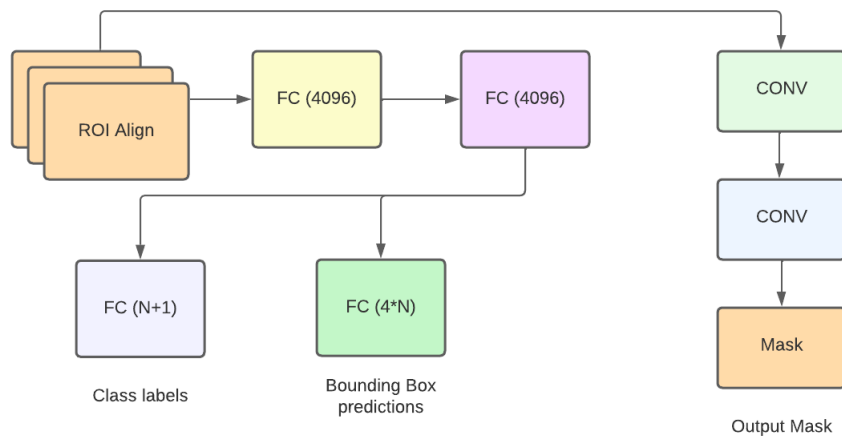


Figure 4.5: Mask R-CNN working architecture

This model can run in real time using GPU and can rise to 5-8 FPS, but it cannot run in real time using CPU. This model surmounts every other existing object detection model in terms of simplicity, performance, efficiency and flexibility. This model is the most successful one in terms of object detection based on instance segmentation[20].

Chapter 5

Dataset

5.1 Dataset

As our topic is comparatively unique, we have generated our own dataset “Crick-Shot” for both training and testing purposes. At first, we collected images from various matches and then we split it into 10 classes such as cover drive, cut drive, off drive, on drive, leg glide, hook, pull etc. our main objective was to train our model accurately so that it can classify the batsman shot position.

Our dataset related works consist of live cricket matches video where we took real time images of the batsman while they were playing the shots. Additionally, when we were taking our images we considered all formats of cricket starting from ODI to T-20 as well as we also considered all the cricket playing nations who are currently playing the cricket under the rules of ICC. Firstly, we took images from live highlights of a match frame by frame by using various kinds of snipping tools. Moreover, while taking images we have faced some challenges such as capturing the exact moment of shots as well as capturing the exact direction of the ball after the shot has been played. In order to solve this problem, we took all our images with a standard frame from the video of live matches. On top of that, to make our dataset efficient we just worked with right handed batsman.



Figure 5.1: A subset of the dataset collected for training and test

Initially, we took around 1800 photos. From those photos we have proposed 85% of photos for training purposes and 15% for testing. While taking all these photos we

were very careful about image clarification. Furthermore, we have always tried to avoid blurring photos in order to overcome biased results.

5.2 Data pre-processing:

Scaling image:

Data pre-processing is a process that transfers raw data to processed data which can clearly be understood by machine. Raw data cannot be directly used in a machine when we are concerned about accuracy so in order to get good results we have to pre-process our raw data to a processed form. Firstly, we have used the pre-processing system for the train data only. We rescaled our data. Previously, our data size was from 0 to 255 and after rescaling the data we brought the value in between 0 to 1.

Augmentation:

We have used 10 percent of the width shift in our data. For width shift floating point number is used and the range is between 0.0 to 0.1. It specifies the upper bound of the fraction of the total width. Width shift is used to shift the image either left or right. Furthermore, we have used 10 percent of the height shift in our data. It is basically the vertical shift of the data. Flipping means rotating an image. We can flip the image both horizontally and vertically. Horizontal flip flops data from left to right and it will not change the pixel information and dimension of the layer. Here in our code we have set the horizontal flip value true. The reason we have used scaling is because it helps us to make the scattered image of the real world more focused. Scaling makes the data points more generalized.

Background removal:

In recent years, CNN has had a significant impact on computer vision. But the accuracy of CNN models varies from dataset to dataset if the images highly vary [21]. Images on a dataset may contain other objects along with the target object, this type of diversion is the main cause of lower accuracy. Background removal is a process where we isolate an image by blurring the surroundings of that particular image and detect object instances in that image. Background removal can be done manually using Photoshop but is time lengthy as well as very complex job. As manually it becomes very difficult to remove background of a large dataset. Also, when this type of task is done manually it is prone to do mistakes.

To overcome this problem, CNN based models are used to remove background and detect objects automatically as well as define their object class.

For example, segmentation, we utilized the Mask R-CNN model, which helps to recognize each item in an image and categorize them as a single entity. The goal of this object detection is to determine the location and class of every object in the image if it includes any. Also, the goal or target for instance segmentation is to detect all the pixels that is the property of that object [22]. To remove background accurately

first segmentation of the image is done then the background removal algorithm is applied. As, if we use the background removal algorithm before segmentation the model may thin our desired object as background and remove that instead of other none essential objects.



Figure 5.2: A subset of the dataset with and without background

From the figure what we have found is that the Mask R-CNN detected each person and removed other objects from the image as well as blurred the other objects like pitch, stamp, scoreboard etc. Mask R-CNN is right now one of the leading object detection model for its accuracy.

5.3 Train test split

Our featured data has been separated into two parts, train and test data. We used 85 percent of them for training and 15 percent for testing reasons. All of this featured data was extracted by the VGG-16 model. Initially, we took 1800 photos and

85% which means 1530 photos were used to train our model and the rest of the photos were kept to test how much our model learned from training. For this paper we have made our own dataset from scratch. We manually took images from highlights of the live cricket matches and then we divided our images into 10 classes. As we have made our data set from our own so we also have randomly chosen the ratio.

Background removal in Training:

As a disadvantage of taking photos manually from various highlights of live cricket matches we continuously failed to capture the batsman only. Along with Batsman the surroundings are also captured by camera which may cause problems for training data set to identify batsman correctly. In order to solve this issue, we have used background removal techniques which eventually helps us to remove unnecessary surroundings from the pictures.

Chapter 6

Model Implementation

This section basically discusses about the implementation part of our model for classification of shots selected by the batsman. This implementation part has to go through several phases to be completed. Such as, input data, feature extraction, classification, and prediction.

As we have used our self-created dataset which was prepared by taking images from cricket matches, we had to prepare our data very carefully to be used as input for the classification. Moreover, we have used some data pre-processing techniques to feed the data to the model such as, scaling, background removal. For implementing our proposed model, we have used two previously trained models VGG16 and Inception-v3 separately to compare the rate of the accuracy.

This chapter also discusses the final results of the proposed model's implementation for classifying the cricket shots selection by the batsman.

6.1 Workflow Overview

As mentioned earlier we have generated our own dataset. There is a sequence of steps which is maintained by us to build the best model for classifying the shots. This section describes the overview of our research:

- First of all, we removed the background of the images. We utilized the Mask R-CNN model for instance segmentation.
- Then we applied scaling in our dataset. We resize the background removed images from (1920 x 1080) to (224 x 244).
- We used a width and height shift of 10% and also flipped the images horizontally to avoid overfitting. Also we rescaled the pixel values of the images from 0 to 255 to 0 to 1.
- During model initialization we have disabled the top layer so that we can use our own customized layer and we have selected the weights of our image from imageNet

as these models have been trained with the data from imageNet.

- After that we added two dense layers and a dropout layer. The last dense layer's activation function is softmax.
- The last layer of our model will give us the output from 10 classes that we have given as input.
- The batch size of VGG-16, we kept as 16 and epoch size is 10 and for Inception the batch size is 32 and the epoch is 55.
- We have used 85% data for training purposes and 15% for testing.
- At last we calculated the accuracy of our system using the testing data. Also we have compared the performance of each model by the accuracy we got.

6.2 Result

After training we predicted the result for our test dataset using the VGG16 model and Inception-v3. The best accuracy is obtained from the VGG-16. We got 95% accuracy by using VGG16 and 85% accuracy by using Inception-v3 for our proposed model. We got lower accuracy in Inception-v3 compared to VGG16.

The following figure is the confusion matrix of VGG-16 and Inception-v3 so that we can provide a clear idea about the accuracy.

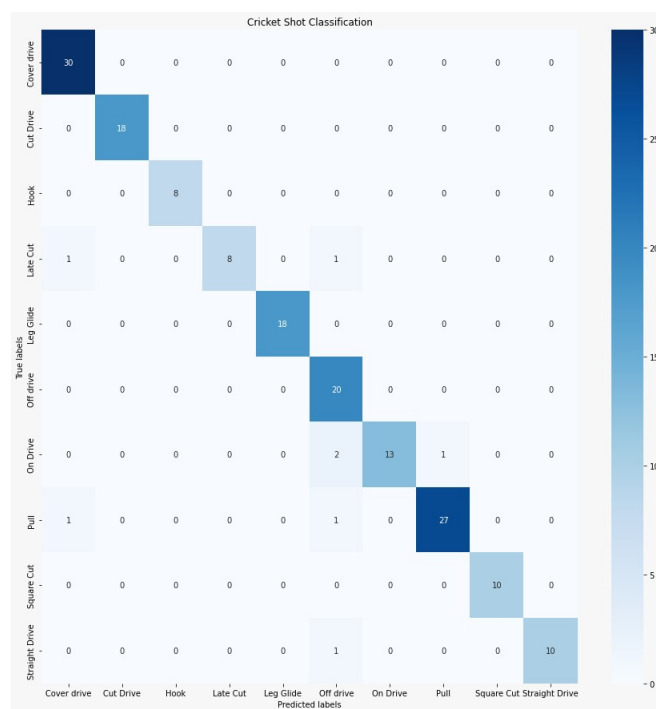


Figure 6.1: Confusion matrix of VGG-16

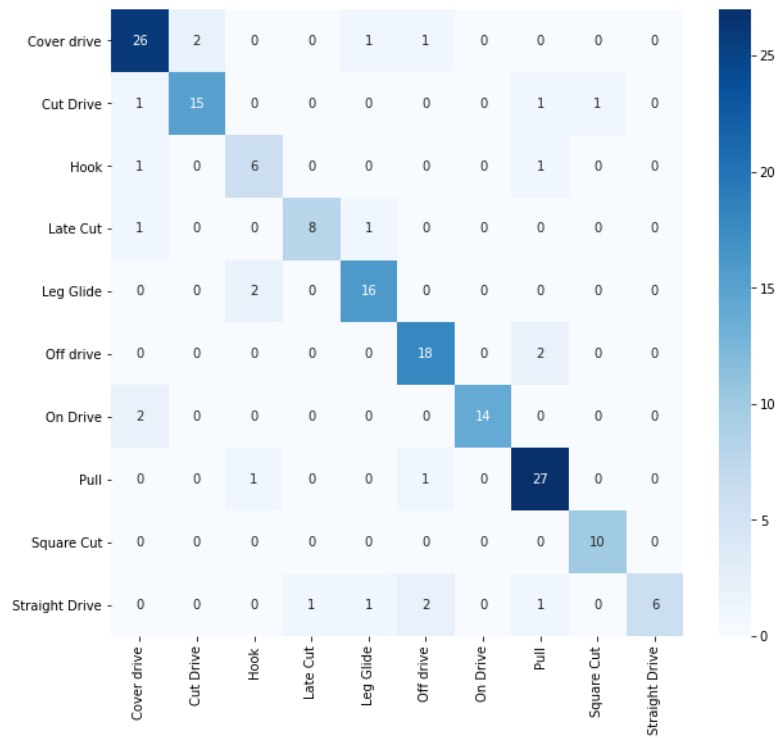


Figure 6.2: Confusion matrix of Inception-V3

The following graphs are model accuracy and model loss graphs of VGG-16:

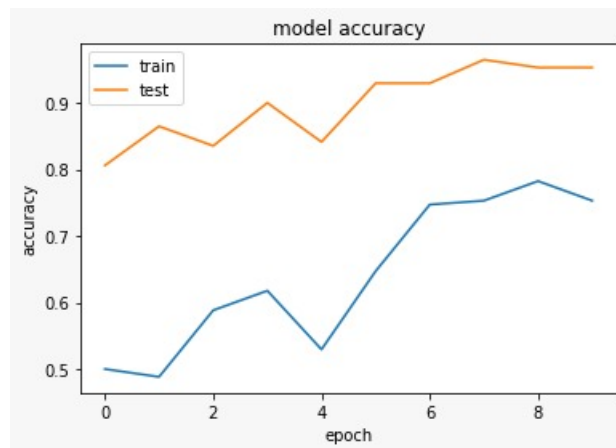


Figure 6.3: Model Accuracy Graph of VGG-16

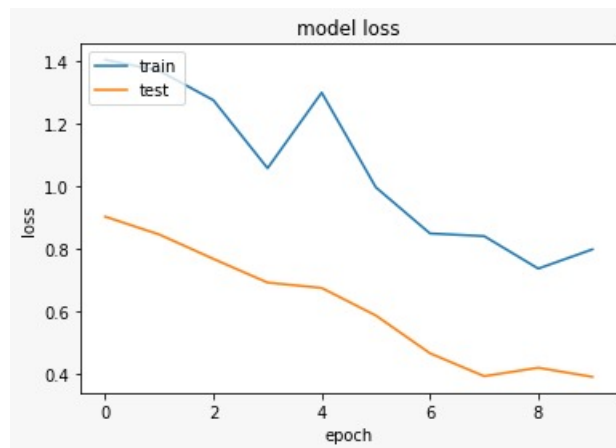


Figure 6.4: Model Loss Graph of VGG-16

The following graphs are model accuracy and model loss graphs of Inception-v3:

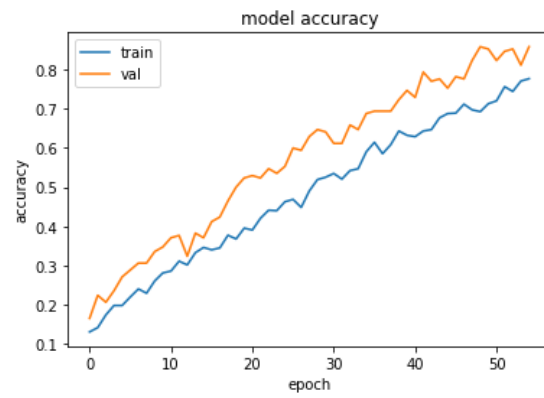


Figure 6.5: Model Accuracy Graph of Inception

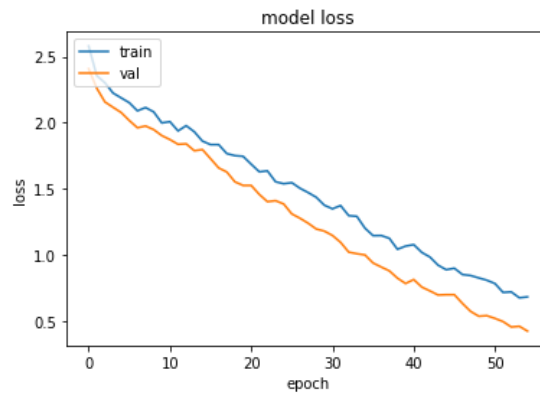


Figure 6.6: Model Loss Graph of Inception

The following table is generated based on our result derived from VGG16 and Inception-v3 model implementation. We have shown the comparison in the below table:

Table 6.1: Comparison Table of VGG-16 and Inception-v3

Shots	Precision (VGG-16)	Re-Call (VGG-16)	F-1 Score (VGG-16)	Precision (InceptionV3)	Re-call (InceptionV3)	F-1 Score (InceptionV3)
Cover Drive	0.94	1.00	0.97	0.84	0.87	0.85
Cut Drive	1.00	1.00	1.00	0.88	0.83	0.86
Hook	1.00	1.00	1.00	0.67	0.75	0.71
Late Cut	1.00	0.80	0.89	0.89	0.80	0.84
Leg Glide	1.00	1.00	1.00	0.84	0.89	0.86
Off Drive	0.80	1.00	0.89	0.82	0.90	0.86
Pull	0.96	0.93	0.95	0.84	0.93	0.89
Square Cut	1.00	1.00	1.00	0.91	1.00	0.95
Straight Drive	1.00	0.91	0.95	1.00	0.55	0.71
On Drive	1.00	0.81	0.90	1.00	0.88	0.93
Weighted Average	0.96	0.95	0.95	0.87	0.86	0.86

Comparison Between VGG-16 and Inception-v3:

The best accuracy is obtained from the VGG16. We got 95% accuracy by using VGG16 and 85% accuracy by using Inception-v3 for our proposed model. We got 10% lower accuracy in Inception-v3 compared to VGG16.

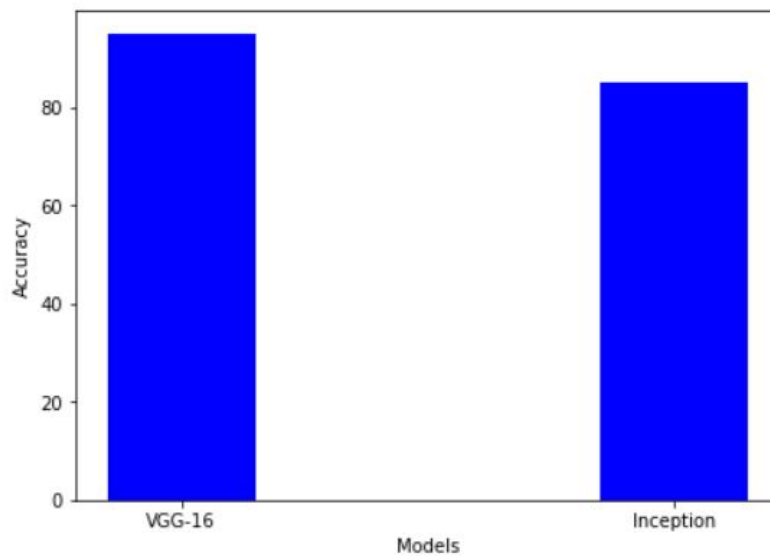


Figure 6.7: VGG-16 vs Inception-v3 Accuracy comparison

For evaluation of the performance of our approach, we have compared our result with a previous paper[23]. They have used CNN for classifying the shots. They have used 6 classes and we have used 10 classes which means they have worked on 6 cricket shots and we have worked on 10 cricket shots. In the next section we are showing the comparison of the results among the similar shots:

Table 6.2: Comparison Table of VGG-16 and Inception-v3 and old paper

Shots	Precision (VGG-16)	Re-Call (VGG-16)	F-1 Score (VGG-16)	Precision (InceptionV3)	Re-call (InceptionV3)	F-1 Score (InceptionV3)	Precision (Previous)	Re-Call (Previous)	F-1 Score (Previous)
Cover Drive	0.94	1.00	0.97	0.84	0.87	0.85	0.74	0.78	0.76
Cut Drive	1.00	1.00	1.00	0.88	0.83	0.86	0.69	0.76	0.72
Straight Drive	1.00	0.91	0.95	1.00	0.55	0.71	0.78	0.83	0.81
Pull Shot	0.96	0.93	0.95	0.84	0.93	0.89	0.89	0.77	0.83

The following table shows that both the models used by us in this research shows better accuracy than the previous model.

Chapter 7

Future Work

There is still a lot of scope left to do research on cricket. In our research, we have classified the shots of different right-handed batsman's. In the near future, we also want to do the same for left-handed batsmen. Moreover, we want to increase our dataset for more precise results. We are hoping to use video clips instead of still images to compare with the result we got from this research so that we can work with the best result. Furthermore, we want to work with the prediction of the direction the ball will go once a shot is played. With our present dataset and features we will use some other features like speed of ball, bat position, bat and ball connection moment, stroke power etc. to determine the direction. Since the field can be divided into some positions like cover, mid-on, mid-off, square leg, third man etc. Also, each position of the field has some sub-positions for instance- square leg position has backward square leg, forward square leg, deep backward square leg, deep square leg and deep forward square leg.

Chapter 8

Conclusion

8.1 Conclusion

Cricket is the world's most popular and widely played sport, having the largest fan base. The influence of technology has not reached all aspects of the game. As most of the tasks are human operated, it leads towards errors. Moreover, it becomes time consuming sometimes. For example, the cameraman sometimes falls back to capture major shots due to lack of attention or fast delivery. Furthermore, checking umpire reviews can take a long time, and the outcome can be flawed. As a result, supporters become quite disappointed if they miss a vital shot at a vital moment. As it has a large fan base, the result of the game should be error free as well as the broadcasting system should be more clear and fast. As a result, this study aims to address the problem by providing a system that will generate a clear picture of shots throughout the live session of the match quickly, ensuring that audience do not miss any vital moment of the match.

Bibliography

- [1] A. Ghani, “Digital cricket training via gamification,” 2021.
- [2] F. Zhang, X. Zhu, and M. Ye, “Fast human pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3517–3526.
- [3] M. Kowsher, M. A. Alam, M. J. Uddin, F. Ahmed, M. W. Ullah, and M. R. Islam, “Detecting third umpire decisions & automated scoring system of cricket,” in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, IEEE, 2019, pp. 1–8.
- [4] M. N. Al Islam, T. B. Hassan, and S. K. Khan, “A cnn-based approach to classify cricket bowlers based on their bowling actions,” in *2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON)*, IEEE, 2019, pp. 130–134.
- [5] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele, “Articulated people detection and pose estimation: Reshaping the future,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3178–3185.
- [6] I. Gregory and S. M. Tedjojuwono, “Implementation of computer vision in detecting human poses,” in *2020 International Conference on Information Management and Technology (ICIMTech)*, IEEE, 2020, pp. 271–276.
- [7] P. R. G. S. A. Reddy, “Human pose estimation in images and videos,” *International Journal of Engineering Technology*, vol. 7, no. 3, p. 27, 2018.
- [8] G. E. V. Bancud and E. V. Palconit, “Human pose estimation using machine learning for cheating detection,”
- [9] W. Ding, B. Hu, H. Liu, X. Wang, and X. Huang, “Human posture recognition based on multiple features and rule learning,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 11, pp. 2529–2540, 2020.
- [10] R. Samad, L. W. Yan, M. Mustafa, N. R. H. Abdullah, and D. Pebrianti, “Multiple human body postures detection using kinect,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 2, pp. 528–536, 2018.
- [11] V. Kress, J. Jung, S. Zernetsch, K. Doll, and B. Sick, “Human pose estimation in real traffic scenes,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2018, pp. 518–523.
- [12] M. Fastovets, J.-Y. Guillemaut, and A. Hilton, “Athlete pose estimation from monocular tv sports footage,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 1048–1054.

- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [14] M. Hussain, J. J. Bird, and D. R. Faria, “A study on cnn transfer learning for image classification,” in *UK Workshop on computational Intelligence*, Springer, 2018, pp. 191–202.
- [15] S. Issa and A. R. Khaled, “Knee abnormality diagnosis based on electromyography signals,” in *International Conference on Soft Computing and Pattern Recognition*, Springer, 2021, pp. 146–155.
- [16] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in mri image using convolutional neural network,” *Math. Biosci. Eng.*, vol. 17, no. 5, pp. 6203–6216, 2020.
- [17] H. Qassim, A. Verma, and D. Feinzimer, “Compressed residual-vgg16 cnn model for big data places image recognition,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2018, pp. 169–175.
- [18] C. Lin, L. Li, W. Luo, K. C. Wang, and J. Guo, “Transfer learning based traffic sign recognition using inception-v3 model,” *Periodica Polytechnica Transportation Engineering*, vol. 47, no. 3, pp. 242–250, 2019.
- [19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [20] T. Cheng, X. Wang, L. Huang, and W. Liu, “Boundary-preserving mask r-cnn,” in *European conference on computer vision*, Springer, 2020, pp. 660–676.
- [21] K. Kc, Z. Yin, D. Li, and Z. Wu, “Impacts of background removal on convolutional neural networks for plant disease classification in-situ,” *Agriculture*, vol. 11, no. 9, p. 827, 2021.
- [22] A.-A. Dalal, Y. Shao, A. Alalimi, and A. Abdu, “Mask r-cnn for geospatial object detection,” *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 12, no. 5, pp. 63–72, 2020.
- [23] M. Foysal, F. Ahmed, M. S. Islam, A. Karim, and N. Neehal, “Shot-net: A convolutional neural network for classifying different cricket shots,” in *International Conference on Recent Trends in Image Processing and Pattern Recognition*, Springer, 2018, pp. 111–120.