Proceedings of the 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2023)
IEEE Xplore Part Number: CFP23OSV-ART; ISBN: 979-8-3503-4148-5

2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) | 979-8-3503-4148-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/I-SMAC58438.2023.10290528

Shot Analysis of Batsmen in Cricket Matches using Transfer Learning Techniques

M.Jagadeesh[1]
AP (SG), Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
jagadeesh.m@rajalakshmi.edu.in

Sooraj Nikam.P[2]
UG Student, Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
200701249@rajalakshmi.edu.in

Sudharsan.V[3]
UG Student, Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
200701260@rajalakshmi.edu.in

Vishnu.S[4]
UG Student, Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
vishnuzeno@gmail.com

Rithesh S[5]
UG Student, Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
rithesh.s.2019.cse@rajalakshmi.edu.in

Sagar Y[6]
UG Student, Department of CSE,
Rajalakshmi Engineering College,
Chennai, Tamil Nadu, India,
sagar.y.2019.cse@rajalakshmi.edu.in

***Abstract* - It is now possible to use deep learning techniques in a variety of disciplines thanks to advancements in hardware technology. Deep learning's Convolutional Neural Network (CNN) design has completely changed computer vision. One area where the use of machine vision is flourishing is in sports. Cricket is a challenging sport that involves many various kinds of shots, bowling motions, and other actions. In a cricket match, each batsman uses a unique batting motion. We use this idea to categorize the various kinds of strokes a batsman takes. In this paper, we suggest a CNN model that uses transfer learning to identify seven distinct cricket batting shots based on batting actions. To train the suggested model and assess its accuracy, we used a dataset called Shot-Net dataset, which contains 4900 images of these seven batting strikes. To construct our model, we have used an array of transfer learning techniques that were trained previously on Image-Net dataset. We chose the algorithm that had the greatest efficiency and learning potential. After experimenting with various approaches, we discovered that the greatest level of model accuracy was achieved by eliminating the weights of 17 layers of the VGG-19 network, expanding the architecture, and training the entire developed model. On the test dataset, our approach achieved an accuracy of 98.61% and minimal cross-entropy loss.**

Keyword — Cricket Shots, Shot classification, Inception, Convolutional Neural Network, Deep Learning, Feature Extraction, Transfer Learning, VGG 19, ResNet (Residual Neural Network), Image Processing for CNNs, Performance Metrics for Shot Classification, Fine-tuning, Training Dataset, Data Augmentation

I. INTRODUCTION

Deep learning algorithms adjust the weights and biases of the network as it gains information in order to minimize the difference between expected and actual results. The popularity of deep learning has grown in recent years, particularly due to its ability to produce state-of-the-art results for a variety of applications, including computer vision, word processing nature, language skills, and self-regulation. It's huge because it's possible to learn hierarchical representation of complex data without a textbook architecture. In addition, advances in hardware and software enable large-scale training and implementation of large deep learning models. Computer platforms, including cloud-based systems and mobile devices. Such progress was previously impossible. Cricket is played with sticks and balls on a 22-yard oval in the middle. Each team has eleven players. First appearing in England in the 16th century, the sport has grown worldwide, particularly in countries such as India, Australia, Great Britain, South Africa, and the West Indies. The history of this game dates back to the 16th century. The object of the game is for one team to attack the other team by hitting the ball with a stick and run towards the goal. The sport is played at a very high level of intensity, and it calls for both physical and cerebral abilities, such as batting, bowling, fielding, and strategy. Machine learning and visual algorithms are being used in various analytical aspects of cricket. It is common for technology to be used to assist coaches and provide visual insights in cricket. There are currently no substantial results from recent studies regarding the detection of cricket shots. It has been hypothesized that deep convolutional neural networks (CNNs) could enhance categorization models' accuracy. According to [12], It is essential for a cricketer to be capable of adapting to different circumstances when playing on different cricket grounds, especially in overseas conditions. Therefore, top hitters have not only powerful muscular hitting abilities but also quick reactions, sound features, and sound strategy as well. CNNs are made use of in the process of identifying components of the human body. This is accomplished by employing algorithms that extract features from real-world data. After that, they assign a category to each individual bodily part by applying a variety of action strategies to the model representation. There is a growing interest in identifying strong players for squad selection as a result of the popularity of fantasy leagues and other services of a nature similar to theirs. This interest is a direct result of the rise in popularity of fantasy sports. The detection of batter shots is one of the most laborious and time-consuming manual chores that is capable of being automated. This process entails a number of steps, such as preprocessing,

image normalization, CNN training, and decision-based testing. Despite the fact that these procedures were carried out by a variety of other researchers, this research paper intends to determine which deep learning model is best suited to the tasks we intend to tackle. The paper intends to train our dataset acquired from [12] (Figure 1) by using various transfer learning techniques to figure out the best shot identification model by using various transfer learning techniques. By conducting various experiments and using different transfer learning techniques, we have found the VGG-19 algorithm to be the most efficient and best suited for our problem of identifying various types of cricket shots.

Fig.1. Shot-Net Dataset **[12]**

II. PREVIOUS WORKS

The majority of databases use video content to construct intricate deep learning models. The substantial amount of time and effort required to utilize these videos, coupled with the limitations of the equipment used, pose a significant challenge. In the context of Cricket videos, the analysis of scoring was conducted as elaborated in reference [1].The purpose of these analytical procedures is to provide insightful information. The four most prevalent outcomes in the game are predicted using CNN and LSTM networks with a video frame rate of 30 frames per second. It is also possible to observe predictive modeling in basketball matches, as shown in citation [2]., each video segment lasts between 6 and 12 seconds, covering 180 to 360 frames. Based on contextual information derived from data for the previous 18 deliveries, this LSTM-based neural network was designed to predict scores after 18 deliveries. The accuracy rate for T20 matches was 82.4% over a period of over ten years (2010-2021). Over 2600 matches were collected.

As documented in [3], this study examines commentators, distinguishing between those who are perceived as favorable and unfavorable by individuals with their preferences, such as positivity or negativity, or friendly vs. technical. Cricket ball deliveries are captured in each video, with different labels for each outcome: hit, no ball, wicket, and boundary. Labels indicate the result of specific balls. Labels indicate the result of specific balls. These segments were gathered from two Indian Premier League cricket match broadcasts. In [4], computer vision and machine learning are applied to shot videos derived from the UCF-101 dataset to classify cricket shots. Both 2D CNN and 3D CNN methodologies are used in this endeavor. Eight distinct shot types are included in the dataset, consisting of 800 balanced video segments lasting 5 to 6 seconds at a frame rate of 4 frames per second. An impressive 90% accuracy rate was achieved by the 3D model, while an impressive 80% accuracy rate was achieved by the 2D architecture.

A deep convolutional neural network is used to process cricket videos in the research paper referred to as [5]. A CRBM layer is used for extracting audio features, while an ISA layer is used for extracting video features. In the end, an auto-encoder employing the Softmax function is implemented for classification purposes. This study addresses the challenge faced by broadcasters in rapidly generating a substantial amount of online multimedia content [6]. This challenge presents a significant hurdle for the effective management of sports video content within the research community. Processing, storage, and transmission of extensive collections of available sports video content require substantial time and effort. The main focus of the mentioned document [7] is centered on the creation and distribution of concise versions of videos. The creation of succinct video summaries is a labor-intensive task that consumes significant time and computational resources. The proposed approach builds upon an Alex-Net CNN pre-trained with Image-Net weights. In order to evaluate performance, multiple sports videos from YouTube, including various television shows and lighting conditions, are selected from YouTube. Using the source video, 6800 frames are extracted and labeled into five categories.

The study outlined in reference [8] proposes the utilization of image processing and machine learning techniques to detect and analyze cracks on cricket pitches. The researchers gathered cricket field images through sources like Google Images and their personal cameras. Their machine learning model was devised and assessed using crack images that were labeled manually. The authors assert an accuracy rate of 90% in the identification of fractures on cricket pitches. However, it's worth noting that the accuracy can be influenced by factors such as the size and intricacy of the fractures.

In their study documented in reference [9], a model was developed that considers both elements related to excitement and event occurrences to effectively identify and segment significant moments during a cricket match. These moments are determined by analyzing various cues, such as video footage, audio levels, player reactions, and pitch conditions. In order to differentiate a game video clip from a replay, a commercial, or an active in-game event, a combination of Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) is used. This process yielded an IOU value of 0.731. In a separate study described in article [10], cricket match videos were investigated. Analyzing these videos is a complex and challenging task. To segment the video into discrete time intervals representing individual ball clips, an Optical Character Recognition (OCR) system is used in conjunction with a CNN model. CNN is adept at recognizing different types of batting shots.

The process includes identifying and annotating changes in runs or wickets and detecting the end of each ball using OCR. The calculated average IOU for eight innings across four games was 0.829, with the highest and lowest IOUs being 0.944 and 0.633, respectively.

Their publication [11] introduced an approach aimed at identifying bowlers within bowling frames. Our belief is that this system will provide substantial benefits to team management, scorers, and broadcasters. This method will eliminate the need for broadcasters to manually track bowlers. With each over, the bowler's identity is updated, allowing the number of deliveries carried out by a specific bowler to be tracked. To classify distinct bowlers, this research uses the VGG 16 algorithm and grayscale representations of players. As a result of their efforts, their performance against the test dataset was impressively accurate at 93.3%.

As a batsman, you need to adjust your approach when playing cricket on different grounds, especially in distant countries. A convolutional neural network (CNN) is used in [12] to determine a player's position and the type of stroke used by the bat. This is accomplished by applying an algorithm that extracts distinctive features from actual data. Within the model framework, distinct strategies are employed to classify each body part. We curated a dataset of 3600 images to facilitate this. Each of the six categories of cricket shots is represented by 600 photographs. The most common types of shots are cover shots, cutaway shots, full shots, straight shots, leg glare shots, and scoop shots. A rate of 80% accuracy was achieved by the classifier.

According to [13], there's a growing desire in cricket to improve performance and success, especially at the best levels along with the importance of understanding specific playing patterns through thorough analysis and machine learning applications. We suggest two novel approaches to deal with this problem. Initially, a CNN-based feature encoding approach is proposed for cricket match prediction. There are around 4,000 parameters in a shallow CNN architecture, compared to the millions in deep CNN architectures used in this study. It is noteworthy, nevertheless, that several essential variables—like strike rates and player rankings—are left out of the analysis.

[14] States those similarities between several shots make it difficult to manually extract features from video frames. The CricShot10 dataset features variable shot lengths and lighting. The accuracy rate of these two models on the CricShot10 dataset was 93%. An approach based on deep learning to analyze multimodal data is proposed in the paper [15]. The proposed model integrates different types of data into one deep learning model using a multi-modal fusion approach. In order to process video and statistical data, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are used. Using a fusion layer, each match's output is combined to create a single representation.

III. PROPOSED MODEL

Using different transfer learning techniques such as Inception V3, ResNet-26, ResNet-50, and VGG to train the model on the Shot-Net dataset, we propose a transfer learning model for identifying cricket batting shots based on their batting actions. We then employ the model with the highest efficiency for our final result. A model trained on one task or dataset is used as a beginning point for another task or dataset in the machine learning and deep learning techniques known as transfer learning. Transfer learning enables us to use the information and feature representations learned from a related task or dataset to fine-tune the model on the new task or dataset rather than training a model from the beginning. Transfer learning can be especially helpful in deep learning, where training large models from scratch can take a long period and require a lot of data. We can frequently accomplish good performance with less training data and in less time by starting with a pre-trained model and fine-tuning it on a new task. Transfer learning comes in two major features:
The pre-trained model is employed in this method as a set feature extractor. A new classifier that is trained on the new job is fed the output of the final layer of the pre-trained model. When the pre-trained model's features are broad enough to be helpful for the new job, this approach performs well. By changing the weights of some or all of the layers, the pre-trained model is further trained on the new job in this method of fine- tuning. When the new task is similar to the job that was previously trained and calls for similar features, fine-tuning may be more effective. Transfer learning has been effectively used in a variety of tasks, including speech recognition, object detection, object classification, and natural language processing. Transfer learning is frequently employed in our work and uses pre-trained models like VGG, ResNet, and Inception V3. A preprocessing step in the proposed approach involves resizing input images (e.g., 224x224 pixels), normalizing pixel values to a standard range, and augmenting the dataset to increase generalization with techniques such as random cropping and flipping. Furthermore, irrelevant metadata is removed, and labels for the target cricket batting shot categories are encoded. Preprocessing ensures that the input data is uniform, well-structured, and suitable for training deep learning models like Inception V3, ResNet-26, ResNet-50, and VGG.

IV. TECHNOLOGY USED

A. Basics Of Convolutional Neural Network (CNN) Convolutional Neural Networks, or CNN, is one of the most potent networks in the Deep Learning space. This is a feed-forward artificial neural network, or ANN. Information travels immediately over a "feed-forward" network. CNNs actually function like the biological visual cortex. One of the best models for categorizing images is CNN. Compared to other conventional image classification techniques, CNN has higher classification accuracy. Feature selection is not necessary for CNNs, but it is for other image classification techniques. In CNN, a variety of shifts are employed.

A picture is passed through a moving filter, or kernel, in the convolutional layer. Typically, you loop through a 2D matrix (an image representation) and extract specific portions, multiply them by points, and then store them in another matrix. The equation below can be used to calculate the dimensions of the output matrix.

Convolutional Layer: The input image or feature map is subjected to a convolution, a mathematical process, by the convolutional layer. This procedure involves computing the dot products between small weighted matrixes called the kernel or filter and local patches of the input data. Each dot product produces a unique output value that creates a fresh output feature map.

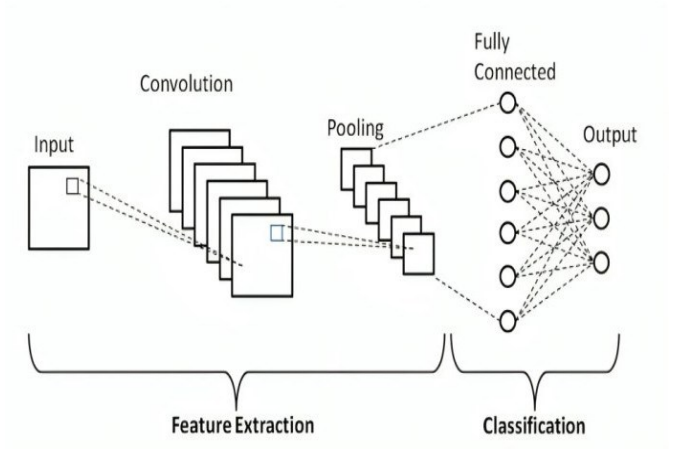

Fig.2. Basic CNN Architecture [16]

The convolutional layer usually applies multiple kernels in parallel to the input data to produce multiple feature maps as output. A particular feature learned by the layer, such as edges, corners, or other visual patterns, is represented by each output feature map. After that, nonlinear activation functions like the Rectified Linear Unit (ReLU) are applied to the output feature maps in order to add nonlinearity to the model and boost its expressive capacity. In addition to introducing nonlinearity to neural networks, ReLU mitigates the vanishing gradient problem, making training more efficient. As a result of its simplicity and computational efficiency, it is a popular choice for activation functions in deep learning, making it more efficient at learning complex representations.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{l-1} \qquad (1)$$

Max Pooling Layer: The fundamental concept behind max pooling is to output the maximum value found within each rectangular area after dividing the input feature map into non-overlapping regions. In doing so, the max pooling layer can reduce the dimensionality of the feature maps, provide some translation invariance, and capture the most important feature in each area. In addition to dimensionality reduction and translational invariance, Max Pooling Layers also improve computational efficiency by reducing the number of parameters required for training. In addition to focusing on important features, they can help reduce noise in the data to mitigate overfitting. For instance, we would obtain a 2x2 output feature map if we applied a 2x2 max pooling with a cadence of 2 to a 4x4 feature map. The maximum value of each 2x2 area in the input feature map that is not overlapped would be taken by the max pooling operation. By reducing the number of factors in the model and offering a kind of regularization, max pooling can aid in preventing overfitting.

Activation Function: An activation function is a statistical function that governs the output of each neuron in a neural network. Neurons' output can be made nonlinear by adding nonlinearity to the activation function, which adds complexity to the relationship between inputs and outputs. Without an activation function, a neuron's output would be a linear function of its inputs, which would restrict the neural network's creative potential. An activation function can change a neuron's output into a nonlinear range, like [0,1] or [-1,1], allowing the neural network to accurately simulate any continuous function. The job at hand and the neural network's structure influence the choice of activation function.

$$Y = \text{Activation function } \Sigma \text{ (Weights*Input+ Bias)} \qquad (2)$$

B. Transfer Learning Techniques Used

Inception V3

Image categorization and recognition are the two primary applications for Google's Inception v3, a convolutional neural network architecture that was developed by the company's researchers. It is an extension of the earlier Inception v1 and v2 models, and it was intended to improve the accuracy of image recognition while simultaneously reducing the number of parameters and the complexity of the computational process. The use of "Inception modules," which are building blocks that incorporate numerous parallel convolutional operations at different scales, is the defining characteristic of the Inception v3 architecture. These modules are known as "Inception modules." Because of these modules, the network is able to capture both local and global features of the image that it is given as input, which results in enhanced accuracy. In addition, Inception v3 makes use of other methods like group normalization and factorization in order to lessen the effects of over-fitting and to enhance generalization. The network is trained on a large dataset such as ImageNet, and it is then able to be fine-tuned for particular tasks such as the detection of objects and the segmentation of images.

ResNet

The term "ResNet", short for "Residual Network", refers to a family of deep neural network architectures first introduced in 2015 by Microsoft researchers. It is easier for the network to learn residual matching than full base matching, which is the central idea of ResNet. The difference between the intended output and input to a particular layer constitutes a residual mapping, and this difference is learned by a residual block

composed of two or more convolutional layers. The ResNet family includes different architectures such as ResNet-34, ResNet-101, ResNet-18, ResNet-152 and ResNet-50. Each of these architectures has a different number of layers than the other architectures in the family. Compared to other architectures that are deeper and more powerful, ResNet-18 and ResNet-34 have architectures that can be described as relatively shallow. For many computer vision tasks, the effectiveness of ResNet-based architectures has reached modern levels.

VGG

In 2014, a research group at the University of Oxford came up with the concept of VGG, which stands for Visual Geometry Group, a design for deep neural networks. VGG networks are known for their ease of performing image distribution tasks with high accuracy. The VGG network has many convolutional layers, followed by the maximum pooling layer, then all layers, and finally for classification. Because the convolutional technique uses small 3x3 filters and step 1, the network can capture detailed information at different scales. Each max pooling layer has a 2-step 2x2 window that both reduces the dimensionality of the spatial feature map and introduces some degree of translational invariance. There are many different versions of the VGG network, each with multiple layers. The first iteration of the VGG network has 16 or 19 layers (depending on version) and is trained on the ImageNet dataset containing over 1 million images divided into 1000 groups. When the ImageNet dataset was published, the VGG network achieved state-of-the-art data performance with an error rate of 7.3% in the validation process of the VGG-16 network. This is the least common error. Since then, the VGG architecture has been widely accepted as the basis for deep learning models for computer vision. Layers detect patterns in the image using 3x3 filters with the ReLU process, while layers select the maxima of 2x2 non-overlapping regions, minimizing the spatial dimensionality of the feature maps. Both layers are part of a neural network. This will help minimize all parameters in the model and prevent the model from being too accurate. After multi-level convolution and pooling, output feature maps are sent through all multi-linked layers.

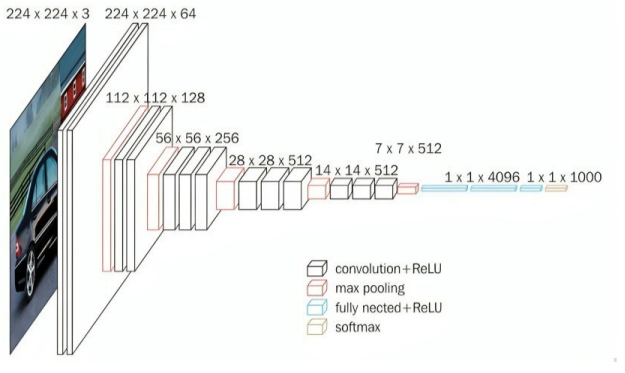

Fig.3. VGG 19 Architecture

The VGG-19 model is a deep convolutional neural network with 19 layers, some of which are layered, some are fully connected, and some are convolutional. The first image is processed by several layers of convolutional and pooling layers, each layer extracting features of the image at a different level of abstraction from the previous layer. All these layers separate the extracted features according to the classification results. To calculate the conversion errors of the input data, the fully coupled operation uses a function called ReLU and the output operation uses a function called softmax to generate the distribution to the different units.

C. Dataset Used

We have used a dataset of 4900 images consisting of 7 types of cricket shots, which are taken from various YouTube videos based on the perfect cricket shots of various cricket batsmen. In their study, Foysal and colleagues utilized a dataset that they devised and manipulated themselves [12]. In order to meet our specific limitations, we selected this dataset after meticulous evaluation. In cricket, players are assessed based on a variety of shots. Our study's objective was to classify five particular shots. There are 600 photographs in each shot category, which includes cover drive, straight drive, scoop shot, cut shot, and pull shot. Foysal partitioned a dataset into two subsets. The testing set has one portion for evaluating accuracy and classification, and the training set has four sections utilized for training.. To counteract the risk of overfitting, they employed artificial augmentation techniques to expand the dataset's content, effectively boosting the volume of pertinent data. These augmentation methods encompassed a range of techniques including the addition of salt and pepper effects, image rotation, and shearing.

Our dataset contains images with unique dimensions, including size, height, and other attributes. To align with our chosen model's prerequisites, which necessitate uniform pixel sizes for both training and testing, we resized the images to dimensions of 100 x 100 pixels. Additionally, we transformed the images into grayscale format, a decision driven by the computational capacity of our computer's GPU. The dataset's compilation involved sourcing images from diverse cricket matches, encompassing World Cups, ODI matches, and test plays. Our compilation strategy involved leveraging YouTube videos to collect images from numerous matches, a process that significantly expedited data gathering in comparison to manual collection methods.

D. Data Preprocessing and Augmentation Techniques

Using modified versions of the original images, image augmentation methods are used to artificially increase the size of a dataset. This can enhance machine learning model efficiency by decreasing over-fitting and boosting generalization. Here are a few typical picture enhancement methods along with their descriptions and equations:

Flipping: Flipping an image creates a new version of the original image that can be used to train models that should be independent of the orientation of the object.

Rotation: By varying the angle at which an image is rotated, new versions of the original picture can be produced. These new versions can be used to train the model to recognize objects from various angles.

Fig.4. Dataset Grayscale Conversion

Scaling: A model can learn to recognize objects at various scales by scaling a picture to produce new versions of the original image in various sizes.

Shearing: By creating new, skewed versions of the original picture through the process of shearing, a model can practice identifying objects in various poses.

Translation: The model can learn to recognize things in various positions by translating an image to produce new versions of the original image that are shifted in various directions.

E. Training the model

In order to achieve a higher level of precision, the end layer of the VGG-19 model was supplemented with an additional five extra dense layers, three of which contained 1024 nodes and two of which contained 512 nodes each, as well as an output layer. This was accomplished through a series of experiments involving trial and error as well as the fine-tuning of parameters. Training was performed on the remaining levels of the general model while the weights of the first 17 layers remained unchanged. The Softmax activation protocol was implemented in the very top stratum of our network. Each of the dense layers has a 10% dropout added to them to keep them from getting too tight. In order to prevent the model from becoming too dependent on the colors of the players' jerseys, each of the training pictures was first transformed into a grayscale image with a single color channel. This was done at the beginning of the process. The grayscale photos were transformed to three channel grayscale before being handed to the model for training because the VGG-19 pre-trained architecture required images to have three color channels. This was done before the model was fed the images. After attempting a number of different optimizers on the model in an effort to bring the cross entropy down to the lowest possible value, the Adam optimizer and categorical cross entropy were ultimately utilized. In this configuration, a group size of 32 was used when feeding the model both the training data and the test data that it was supposed to be analyzing. The model's training consisted of a total of 150 iterations. A short summary of the various attributes of our model is provided in Table II. Using the Adam optimizer and the categorical cross entropy loss function with the aforementioned learning rate, we trained the dataset for numerous transfer learning models. The Table I list the various methods we employed as well as the consistency of each model with the test set.

Table I: Accuracy Table

S.no	Algorithm Used	Optimizer	Loss Function	Accuracy
1	Inception V3	RMSProp	Categorical Cross Entropy	77.25
2	ResNet 26	RMSProp	Categorical Cross Entropy	55.87
3	ResNet 50	Adam	Categorical Cross Entropy	62.35
4	VGG 16	Adam	Categorical Cross Entropy	93.37
5	VGG 19	Adam	Categorical Cross Entropy	98.61

Table II: Techniques Performed on the Model

Properties	Used functionalities
Transfer Learning Model	VGG 19
Weights	The weights of the first and thee last layers are not considered
Activation Function	ReLU for the hidden layers and Softmax for the output layer.
Loss Function	Categorical Cross Entropy
Output Classes	7

Regularization	,Data Augmentation using width shift range of 0.3, height shift range of 0.3 and zoom range of 0.2
Accuracy	98.61

V. RESULTS AND DISCUSSION

Thus, the model has been successfully trained and the efficiency of the model that was performed on the test set has reached almost a hundred percent. On the model, we have applied a variety of evaluation methods. The accuracy chart of a deep learning model is a visual depiction of the model's performance in terms of accuracy on a particular task or dataset. This performance is assessed by how well the model can categorize the data. The frequency with which a model predicts the class of a given input is known as accuracy. For instance, in a classification problem, accuracy refers to how often the model correctly predicts the class. In most cases, the accuracy chart displays, while the model is being trained, the level of precision it achieves on a validation dataset. The accuracy chart of our model is depicted in Figure 5.

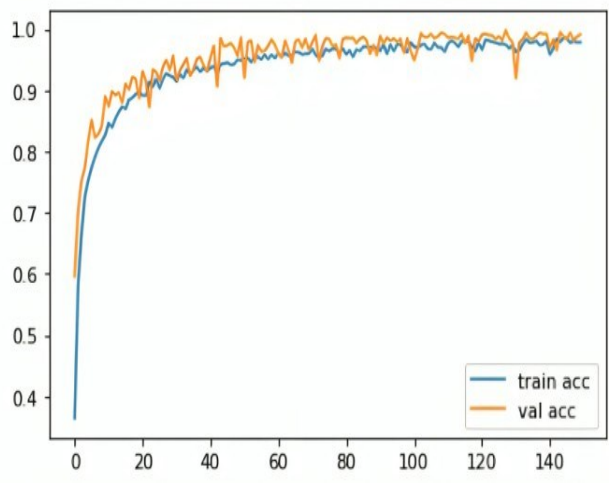

Fig 5: Accuracy Chart

A table that summarizes the performance of a deep learning model on a classification assignment is called a confusion matrix. This table is part of a deep learning model. It demonstrates the amount of accurate positive predictions, accurate negative predictions, false positive predictions, and false negative predictions that the model made. It is helpful to evaluate the performance of a classification model using the confusion matrix, particularly in cases where the classifications do not have an equal number of instances. It gives information about the model's accuracy, precision, recall, and F1-score, all of which are measures that are typically utilized to evaluate the performance of a classification model. Calculating these metrics, which can help determine the strengths and weaknesses of the model as well as tune its parameters, can be done with the confusion matrix [Fig.6], which can be used to do the calculations.

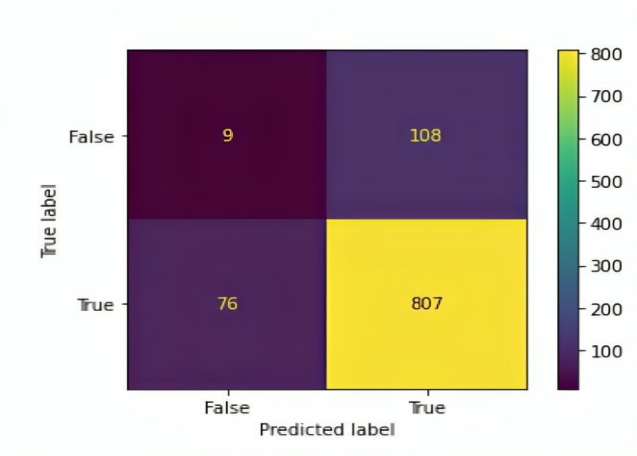

Fig 6: Confusion Matrix

Precision is the proportion of related instances among retrieved instances. In other words, it assesses the accuracy of accurate forecasts, in other words. Accuracy formula:

Accuracy = true positive / (true positive false positive) (3)

Recall is the percentage of relevant instances that are retrieved. Positive predictions are measured by how complete they are Feedback Formula:

Feedback = true positive / (true positive false negative) (4)

The F1 score is the harmonic average of accuracy and recall combining the two metrics to produce a single balanced score. . These metrics are calculated from the model and are used in Table III below.

Table III: Precision, Recall and F-Score

Output Class	Precision	Recall	F-score
Pull Shot	0.93	0.97	0.91
Scoop Shot	0.89	0.99	0.94
Cut Shot	0.97	0.95	0.92
Leg Glance	0.99	0.87	0.92
Straight Shot	0.95	0.97	0.86
Cover Drive	0.88	0.93	0.97
Sweep Shot	0.83	0.97	0.93
Average	0.91	0.95	0.92

VI. CONCLUSIONS AND RECOMMENDATIONS

We have successfully developed and trained the VGG – 19 model to classify cricket shot images and the model has achieved a significantly higher efficiency. In our future work, we intend to train video based datasets using CNN and LSTM techniques to improve the shot recognition system. We also intend to provide a flutter application for the model so that the users can find benefit in this application by applying it to real-world scenarios. This model can help coaches and batsmen to

improve their performance by giving them insights on the shots and suggestions for playing the perfect cricket shot. Furthermore, to improve the model's performance, we can incorporate various techniques like transfer learning with the existing larger models (VGG-16, ResNet-50) with advanced architectures and by experimenting with many hyperparameters and optimizers. In addition to that, the training data could be made more diverse by the application of advanced data augmentation techniques.

REFERENCES

[1]. Kumar, Rohit, D. Santhadevi, and Janet Barnabas. "Outcome classification in cricket using deep learning." In *2019 IEEE international conference on cloud computing in emerging markets (CCEM)*, pp. 55-58. IEEE, 2019.

[2]. Prateek Gupta, Navya Sanjna Joshi, Raghuvansh Tahlan, Darpan Gupta, & Saakshi Agrawal. (2021). "Cricket Score Forecasting using Neural Networks". International Journal of Engineering and Advanced Technology (IJEAT), 10(5), 366–369.

[3]. Dixit, Kalpit, and Anusha Balakrishnan. "Deep learning using cnn for ball-by-ball outcome classification in sports." *Report on the Course: Convolutional Neural Networks for Visual Recognition; Stanford University: Stanford, CA, USA* (2016).

[4]. Khan, Muhammad Zeeshan, Muhammad A. Hassan, Ammarah Farooq, and Muhammad Usman Ghanni Khan. "Deep cnn based data-driven recognition of cricket batting shots." In *2018 International Conference on Applied and Engineering Mathematics (ICAEM)*, pp. 67-71. IEEE, 2018.

[5]. Sachan, Devendra Singh, Umesh Tekwani, and Amit Sethi. "Sports video classification from multimodal information using deep neural networks." In *2013 AAAI Fall Symposium Series*. 2013.

[6]. Nasir, M., Ali Javed, Aun Irtaza, Hafiz Malik, and M. Mahmood. "Event detection and summarization of cricket videos." *Journal of Image and Graphics* 6, no. 1 (2018): pp. 27-32.

[7]. Rafiq, Muhammad, Ghazala Rafiq, Rockson Agyeman, Gyu Sang Choi, and Seong-Il Jin. "Scene classification for sports video summarization using transfer learning." *Sensors* 20, no. 6 (2020): 1702.

[8]. KANHAIYA, KISHAN, RAJAT GUPTA, and ARPIT KUMAR SHARMA. "Cracked cricket pitch analysis (CCPA) using image processing and machine learning." *Global Journal on Application of Data Science and Internet of Things [ISSN: 2581-4370 (online)]* 3, no. 1 (2019).

[9]. Shukla, Pushkar, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. "Automatic cricket highlight generation using event-driven and excitement-based features." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1800-1808. 2018.

[10]. Agarwal, Sanchit, Nikhil Kumar Singh, and Prashant Giridhar Shambharkar. "Automatic Annotation of Events and Highlights Generation of Cricket Match Videos." *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-8 Issue-11, September 2019

[11]. Al Islam, Md Nafee, Tanzil Bin Hassan, and Siamul Karim 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)*, pp. 130-134. IEEE, 2019.

[12]. Foysal, Md Ferdouse Ahmed, Mohammad Shakirul Islam, Asif Karim, and Nafis Neehal. "Shot-Net: A convolutional neural network for classifying different cricket shots." In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part I 2*, pp. 111-120. Springer Singapore, 2019.

[13]. Manivannan, Siyamalan, and Mogan Kausik. "Convolutional neural network and feature encoding for predicting the outcome of cricket matches." In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pp. 344-349. IEEE, 2019.

[14]. Sen, Anik, Kaushik Deb, Pranab Kumar Dhar, and Takeshi Koshiba. "Cricshow Classify: an approach to classifying batting shots from cricket videos using a convolutional neural network and gated recurrent unit." *Sensors* 21, no. 8 (2021): 2846.

[15]. Alaka, Souridas, Rishikesh Sreekumar, and Hrithwik Shalu. "Efficient Feature Representations for Cricket Data Analysis using Deep Learning based Multi-Modal Fusion Model." *arXiv preprint arXiv:2108.07139* (2021).

[16]. Recent Advances in Video Analytics for Rail Network Surveillance for Security, Trespass and Suicide Prevention—A Survey https://doi.org/10.3390/s22124324