Received Date: 19th October, 2022
Accepted Date: 20th March, 2023

Cricket Shots Analysis Using AI

Bhuwan Khatiwada[1*], Bishwa Prakash Subedi[2], Niraj Duwal[3], Rewan Gautam[4] and Suramya Sharma Dahal[5]

[1]Department of Electronics Engineering, Thapathali Campus, E-mail: tha075bei009@tcioe.edu.np
[2]Department of Electronics Engineering, Thapathali Campus, E-mail: tha075bei011@tcioe.edu.np
[3]Department of Electronics Engineering, Thapathali Campus, E-mail: tha075bei028@tcioe.edu.np
[4]Department of Electronics Engineering, Thapathali Campus, E-mail: tha075bei011@tcioe.edu.np
[5]Associate Professor, Department of Electronics Engineering, Kathmandu Engineering College, E-mail: suramya.sharma@kecktm.edu.np

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

Keywords — **AI, CNN, Cricket, ML, Model, Pose, Shots, SVM**

I. INTRODUCTION

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

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

Depending on the position of the batter and the angles at which they use the bat there are several cricket shots such as the cover drive, the straight drive, the pull shot, the reverse sweep, and many more.

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

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

II. RELATED WORKS

Some works have been done on cricket analysis in the last decade using different approaches. A group of researchers from Bangladesh which includes D Karmaker, AZM E Chaudhary, M S U Miah, M A Imran, M H Rahman published a paper "Cricket Shot Classification using Motion Vector" in 2015. It was based on directional optical flow vectors created by body parts from videos and then transferring the vectors to an angle [1]. The paper included 8 cricket shots classified in a frame according to angle ranges for each shot.

* *Corresponding Author*

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

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

The accuracy level of this approach was not satisfactory, the maximum accurate result was obtained on the Off Drive shot which was 63.57%. In another research paper, Foysal, Islam, Karim, and Neehal published a paper entitled "Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots". They used 13 layered Convolutional Neural Networks to extract the feature to classify six categories of cricket shots [2]. The CNN model consisted of three convolution layers, three max-pooling layers, four dropout layers, and two dense layers.

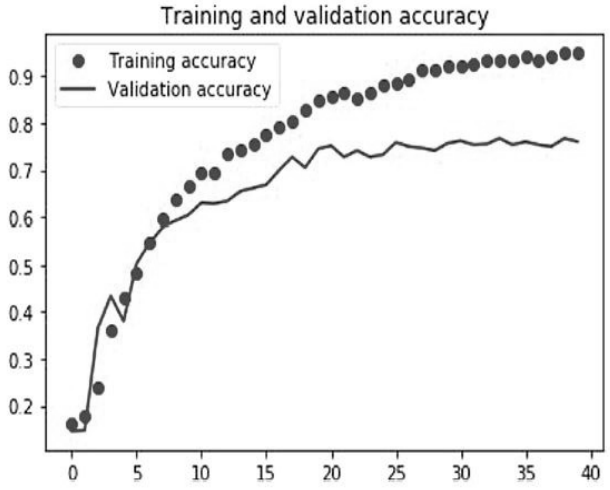


Figure 1: Training and Validation Accuracy [2]

If we analyze the performance of this model, the model seems to be overfitting because there is a huge gap in the accuracy level of training and validation data. We approached solving this with a different kind of technique to classify cricket shots.

Instead of directly feeding the data to the algorithm, we extracted features using pose detection of the batter, filtered the necessary features, and fed it to the machine learning algorithm. Since the pose of the batter highly determines the shot being played, the coordinates of the body parts gave us better results.

III. METHODOLOGY

A. Data Collection and Feature Extraction

The data we require for our tasks is the images of batters playing a particular shot which is not available publicly on the internet. The dataset for shot classification was collected using web scraping and also captured images manually from the ground to increase the data. The types of shots we collected were Cut Shots, Cover Drives, Straight Drives, Leg Glances, Pull shots, and Scoops. There were around 100 images for each shot which was increased to 600 by performing data augmentation through rotating, shifting, changing brightness, and inserting noise. So total dataset for shot classification was 3600. Since the pose of the batter highly impacts the shot the batter is playing, we used it as a feature. We extracted pose features using mediapipe. We were able to extract the coordinates of 33 3D body parts landmarks.

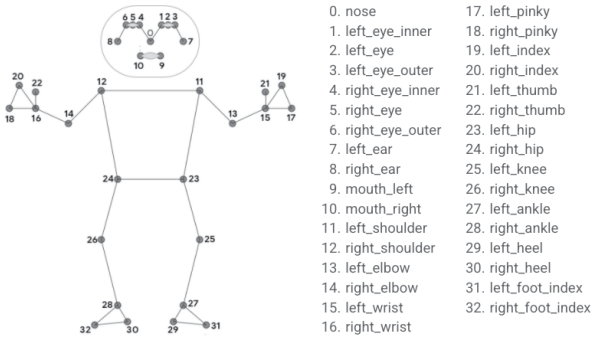


Figure 2: Mediapipe Body Landmarks [3]

Through one body part, we could get the coordinate values for the x, y, and z-axis and visibility.

The sample CSV dataset obtained after preprocessing is tabulated below:

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

Figure 3: Sample Dataset

Now, for feature selection, we use Pearson's correlation. Pearson's correlation attempts to draw a line to best fit through the data of two variables. Pearson's correlation coefficient is the measure of the strength of a linear association between two variables. It indicates how far away all these data points are from this best-fit line. It ranges from -1 to 1. Mathematically it is given by,

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \qquad (1)$$

Filtering the result for only the target variable, we obtained the correlation of the target and other variables.

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

Figure 4: Sample of Pearon's correlation of features with target variable

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

For the detection of bats, 201 images were used which were labeled and annotated using labelImg [4]. Annotation is stored in XML format for each image file item. The XML file contains the image width, height, depth, bounding box coordinates, and the label it belongs to.

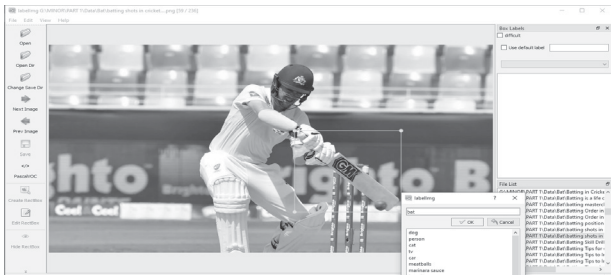


Figure 5: Labeling Interface [5]

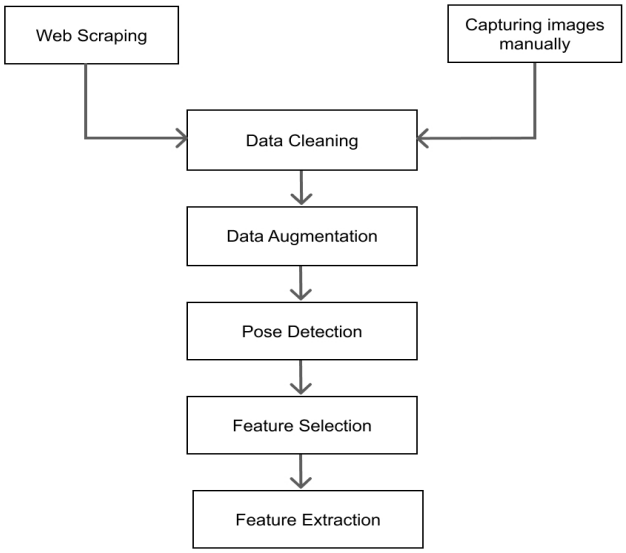


Figure 6: Data Collection and Feature Extraction Block Diagram

B. Shots Classification

For the classification of different cricketing shots, we are using Support Vector Machine (SVM) classifier. SVM is a supervised machine learning algorithm which works by plotting the data items in the space determined by the number of features we have in our dataset and finding the decision boundary i.e a hyper-plane that separates the classes. A hyperplane in an n-dimensional space is a decision boundary of an n-1 dimensional subset of that space that divides the space into required parts. SVM creates a number of hyperplanes, and the best hyperplane is chosen according to the marginal distance between the nearest data items of each class. So, the hyperplane with the maximum marginal distance is considered optimal by SVM.

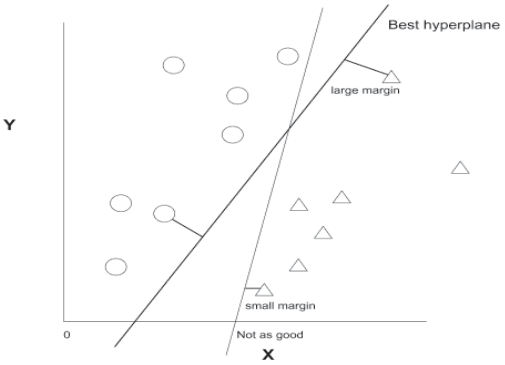


Figure 7: Best Hyperplane

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

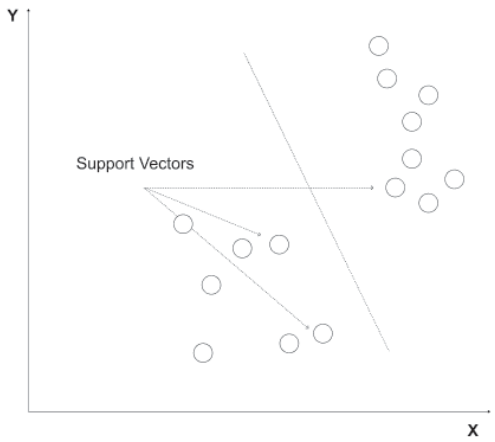


Figure 8: Support Vectors Illustration

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

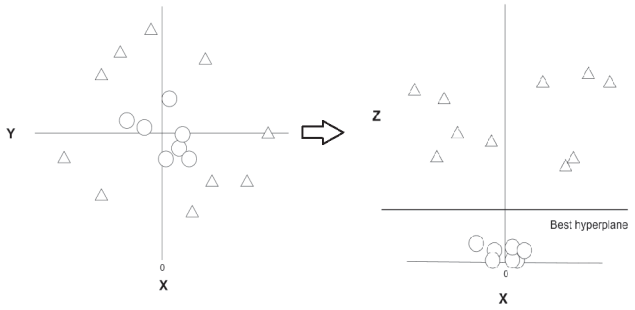


Figure 9: Conversion to Higher Dimension

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

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

$$wx - b = 0 \qquad wx - b = 0 \qquad (2)$$

where, x = input data

w = weight vector

b = bias vector

If we have two labels say 1 and -1,

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

where, x_i = ith input data

y_i = label of ith input data

We use hinge loss as a cost function in SVM,

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

$$J_i = \begin{cases} \lambda\|w\|^2, & y_i.f(x), \geq 1 \\ \lambda\|w\|^2 + 1 - y_i(wx_i - b), & y_i.f(x), \leq 1 \end{cases}$$
$$J_i = \begin{cases} \lambda\|w\|^2, & y_i.f(x), \geq 1 \\ \lambda\|w\|^2 + 1 - y_i(wx_i - b), & y_i.f(x), \leq 1 \end{cases} \qquad (3)$$

where, J = Cost function

L = Hinge loss

n = number of training examples

λ = regularization parameter

f(x) = wx_i-b

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

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

$$\frac{dJ_i}{dw} = \begin{cases} 2\lambda w, & y_i.f(x), \geq 1 \\ 2\lambda w - y_i x_i, & y_i.f(x), \leq 1 \end{cases}$$
$$\frac{dJ_i}{dw} = \begin{cases} 2\lambda w, & y_i.f(x), \geq 1 \\ 2\lambda w - y_i x_i, & y_i.f(x), \leq 1 \end{cases} \qquad (4)$$
$$\frac{dJ_i}{db} = \begin{cases} 0, & y_i.f(x), \geq 1 \\ y_i, & y_i.f(x), \leq 1 \end{cases}$$
$$\frac{dJ_i}{db} = \begin{cases} 0, & y_i.f(x), \geq 1 \\ y_i, & y_i.f(x), \leq 1 \end{cases} \qquad (5)$$

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

$$w = w - \alpha\frac{dJ_i}{dw}w = w - \alpha\frac{dJ_i}{dw} \qquad (6)$$
$$b = b - \alpha\frac{dJ_i}{db}b = b - \alpha\frac{dJ_i}{db} \qquad (7)$$

Where, α = learning rate

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

C. Shots Efficiency Detection: EfficientDet

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

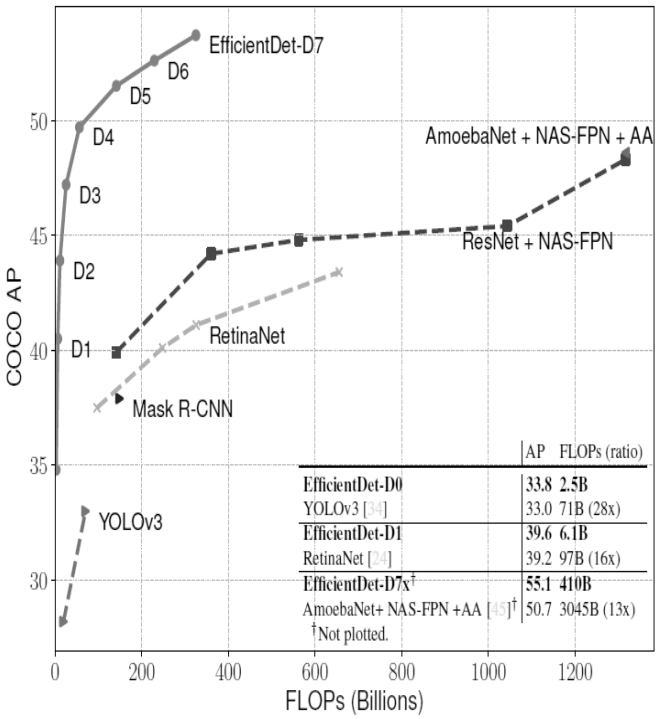


Figure 10: Accuracy vs number of parameters [6]

Efficient Det utilizes several optimizations and backbone tweaks, such as the use of a BiFPN, and a compound scaling method that uniformly scales the resolution, depth and width for all backbones, feature networks, and box/class prediction networks at the same time [6].

Image resolution,

$$R_{input} = 512 + \emptyset.128 R_{input} = 512 + \emptyset.128 \quad (8)$$

BiFPN,

$$W_{bifpn} = 64\left(1.35^{\emptyset}\right)$$
$$W_{bifpn} = 64\left(1.35^{\emptyset}\right) \quad (9)$$

$$D_{bifpn} = 2 + \emptyset D_{bifpn} = 2 + \emptyset \quad (10)$$

Box/class predict network,

$$D_{box} = D_{class} = 3 + \emptyset/3$$
$$D_{box} = D_{class} = 3 + \emptyset/3 \quad (11)$$

For the object detection which is bat and ball, we use the EfficientDet algorithm. The detection model uses EfficientNet for feature extraction which acts as a backbone. EfficientNet uses a compound scaling method. Unlike conventional practices that arbitrarily scale these factors, the compound scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

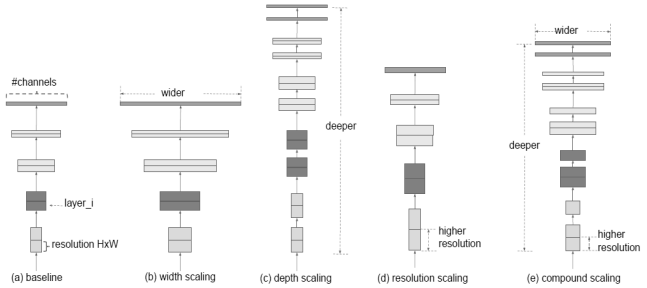


Figure 11: Depth Scaling Methods [7]

The BiFPN as a shared class/box prediction as well as feature network introduces learnable weights to learn the importance of different input features, while repeatedly applying top-down and bottom-up multi-scale feature fusion. It takes level 3-7 features namely (P3-P7) and they are fed to the network one by one. We needed optimized models for deploying it to edge devices i.e smartphones in our case. So, EfficientDet was the best fit for our task.

After labeling and annotating the images, the data was loaded and stored in Pascal VOC format. Pascal VOC is an XML file that is created for each image which stores the bounding boxes as (xmin, ymin, xmax, ymax). The annotation file and the image file are read by the TensorFlow data loader. Defining the batch size = 4, epochs = 100, the model was trained.

D. System Working

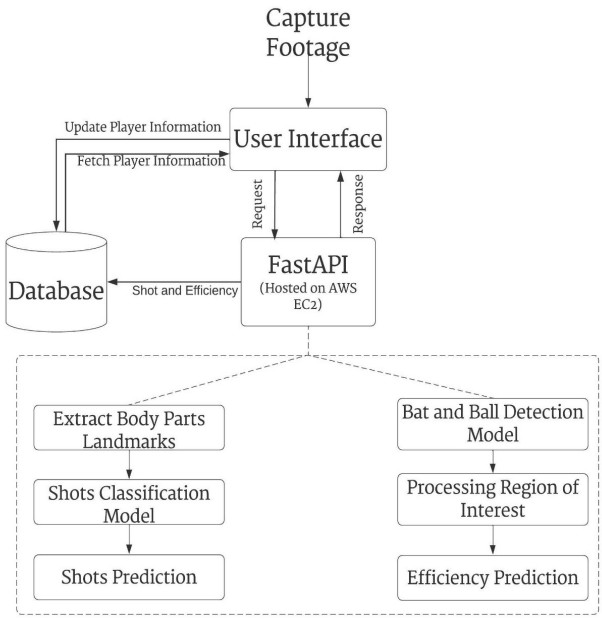


Figure 12: Working System Block Diagram

The above figure shows the system block diagram. Through the mobile application, the the input image can be captured using the device's camera or an already captured image can be uploaded as input. This can be performed with the assistance of the user interface. The user interface has

features like uploading or capturing images, adding player profiles for new players, and a portal for recent cricketing news and articles. The player profile includes information like the name of the player, age, batting style, bowling style, playing role, and team. Through UI, we store this information in a database and fetch them from the same database whenever required.

We obtain the required input through the UI. The FastAPI which is hosted on AWS EC2 instance server loads the model and helps process the given input. The processing steps can be seen in the given expanded view. For shot classification, firstly the body part landmarks are extracted from the input images, and thus extracted features are fed to the shot classification SVM model which finally predicts the shot played. And for the shot efficiency, we feed the image to the bat detection EfficientDet model, the model processes the image to determine the region of interest i.e., the area of the bat. Then upon detecting the ball, depending on whether the ball lies within the region of interest or not, the efficiency of the shot is predicted.

IV. Results and Analysis

A. Shots Classification

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

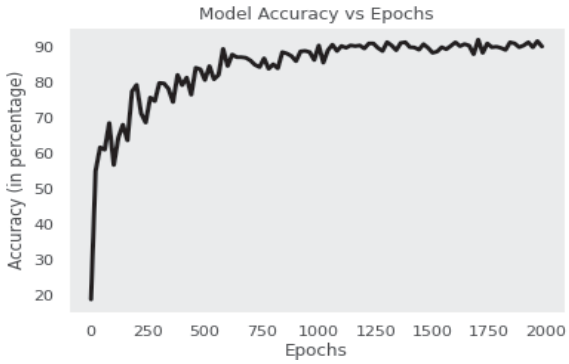


Figure 13: Model Accuracy vs Epochs

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

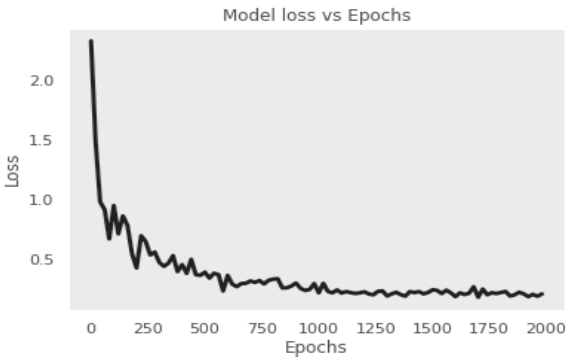


Figure 14: Model loss vs Epochs

The Hinge Loss observed was 0.2.

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

Figure 15: Predicted Pull Shot [8]

Figure 16: Predicted Scoop Shot [9]

Figure 17: Predicted Cut Shot [10]

Figure 18: Predicted Leg Glance Shot [11]

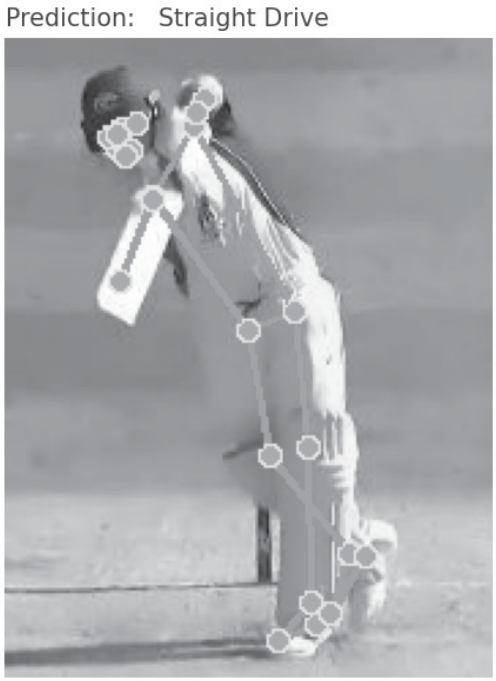
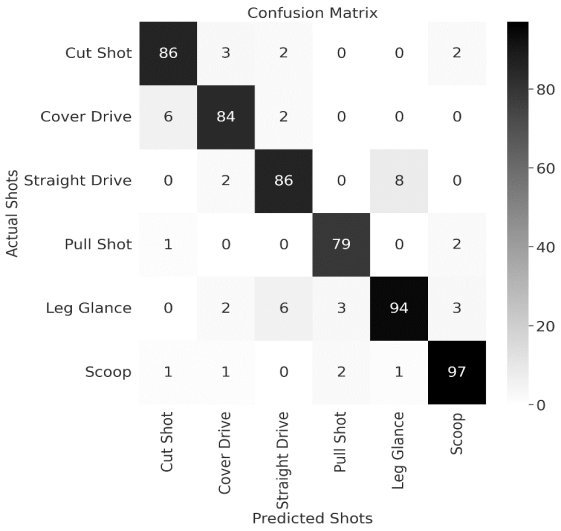


Figure 19: Predicted Straight Drive [12]

Figure 20: Predicted Cover Drive

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

Table 2: Confusion Matrix



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

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

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

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

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

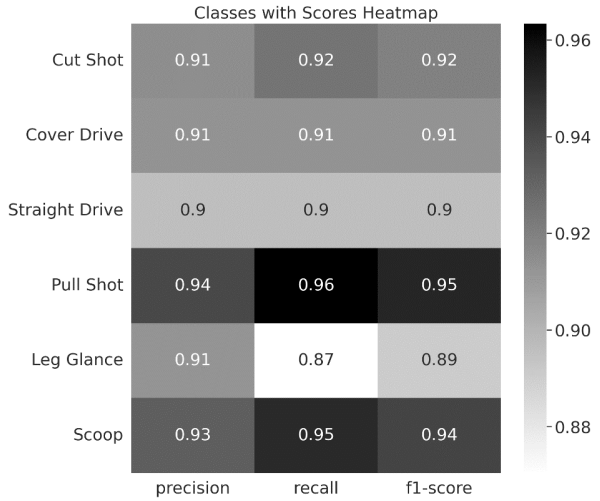
The highest recall was obtained for Pull Shot (0.96).

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

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \quad (14)$$

The pull shot had better F1 score (0.95).

Table 3: Evaluation Metrics



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

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR (True Positive Rate) against FPR (False Positive Rate) at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

In a ROC curve, a higher X-axis value indicates a higher number of False positives than True negatives. While a higher Y-axis value indicates a higher number of True positives than False negatives. So, the choice of the threshold depends on the ability to balance between False positives and False negatives.

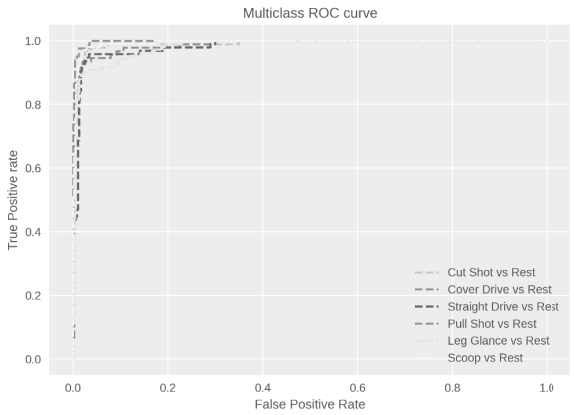


Figure 21: Multiclass ROC Curve

B. Bat Detection

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

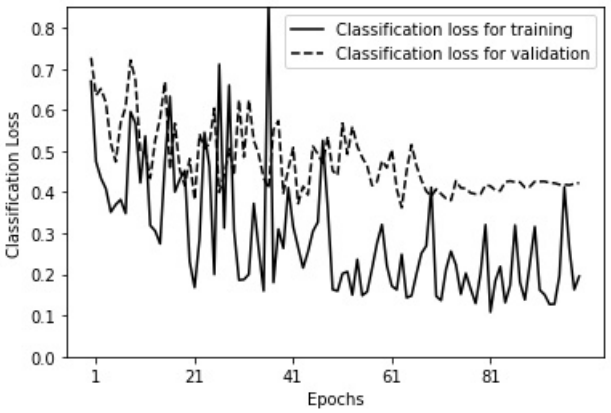


Figure 22: Classification Loss

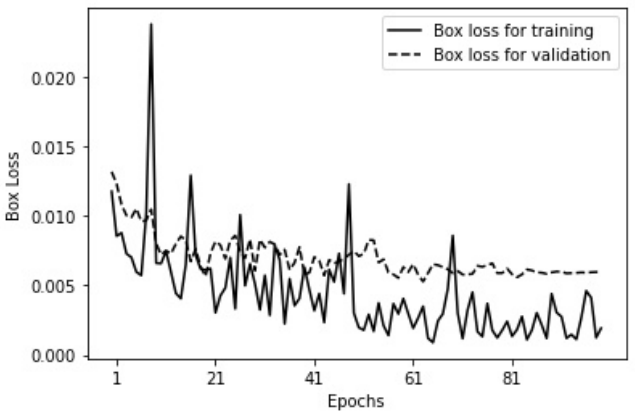


Figure 23: Box Loss

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

Figure 24: Bat Detection Far Angle [4]

Figure 25: Bat Detection Close Angle

Figure 26: Bat Detection on Focused Image [13]

Figure 27: Shot and Efficiency Prediction [14]

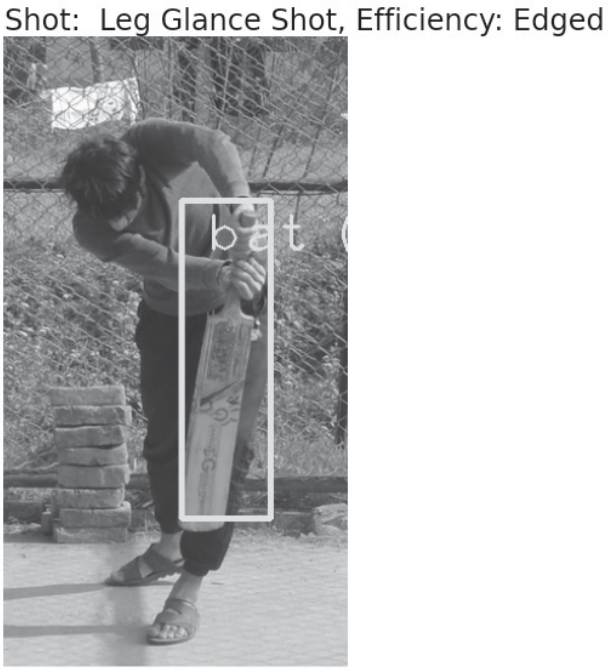


Figure 28: Shot and Efficiency Prediction (Wrong Result)

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

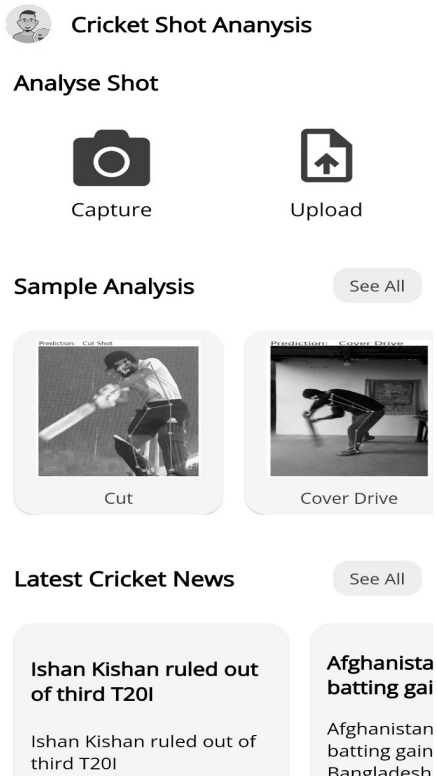


Figure 29: User Interface (1)

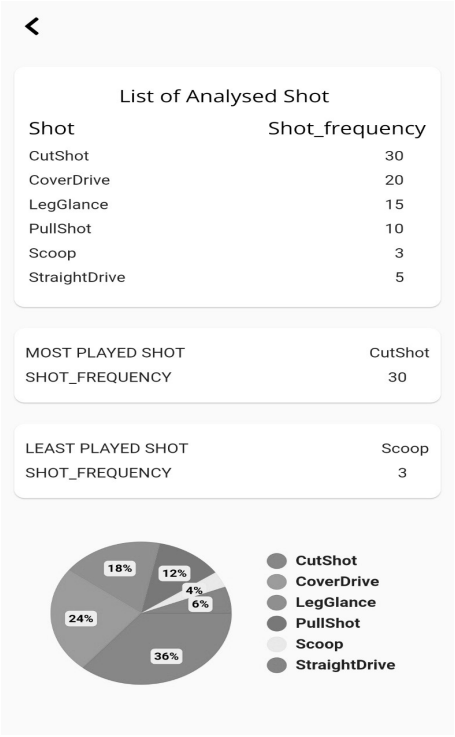


Figure 30: User Interface (2)

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

V. Conclusion

The game of cricket involves numerous planning and execution of strategies. Every professional cricket team has a cricket analyst supporting them. In this project, we purposed a system to automate some of the tasks of the analyst. Using the SVM model we were able to classify the shots into six categories (Cover Drive, Straight Drive, Scoop, Cut, Pull, Leg Glance), and the efficiency of the shot was predicted (Missed, Edged, Perfect) using the EfficientDet model architecture. The mobile application was developed to make our system accessible to anyone at any time.

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

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

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

VI. Future Enhancements

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

as shot played so we can provide a complete Wagon Wheel and all the information of the delivery and shot played.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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