Cricket Umpire Gestures Image Recognition

*

Abhishek Puppala
Department of Information Science
University of North Texas
Texas, USA
abhishekpuppala@my.unt.edu

Jashwanth Kalyan Polavarapu
Department of Information Science
University of North Texas
Texas, USA
JashwanthKalyanPolavarapu@my.unt.edu

Kushal Krishna Vankalapati
Department of Information Science
University of North Texas
Texas, USA
KushalKrishnaVankalapati@my.unt.edu

Manisha Lokasani
Department of Information Science
University of North Texas
Texas, USA
ManishaLokasani@my.unt.edu

***Abstract*—Cricket is a popular sport enjoyed by millions around the world, particularly in countries like India, England. It involves two teams, each with eleven players and three umpires, competing on a large oval field. Our research mainly focuses on the comparison of classification and deep learning model regarding the automatic recognition of cricket umpire gestures in the sport of cricket from images, that are collected over various game plays of cricket. This project mainly handles over the dataset comprising of 2000 images, that are collected from various online sources like kaggle, github. Some of the other amount of images are collected manually by taking snapshots from 100's of videos game play and being partly generated with the help of Generative Adversarial Networks (GANs) algorithm based on the existing images using the various kinds of prompts worked over online platforms such as runwayml and other text to image techniques. The primary methodology deployed is used to recognize and classify the nine distinct umpire gestures, they are six, out, not-out, no-ball, four, byes, leg-byes, wide and no-action class. This study aims to compare the performance of CNN models with other classification models in terms of accuracy by deploying the models on various thresholds and tuning on different parameters to improve the model performance better. By automating the recognition of umpire gestures, the project seeks to facilitate real time updates in cricket scoring systems, by reducing the human readability error and enhance the viewer's experience by providing instant automated game commentary. The anticipated outcomes include not only advancement in sports technology but also potential applications in other areas such as baseball and soccer requiring real-time gesture recognition.**

***Index Terms*—Convolutional Neural Networks (CNNs), Automated Sports Commentary, Image Classification, Deep Learning, Generative Adversarial Networks (GANs), Real-time Image Processing, Classification learning.**

I. Introduction

One of the most watched sports in the world today is cricket, which draws large crowds to arenas, TVs, and internet platforms. The popularity of the sport draws larger and larger crowds of spectators, highlighting the significance of its intricate rules and regulations that uphold the game's fairness and integrity. These rules are necessary to guarantee appropriate behavior in the sport.

The umpire is an essential component of cricket match administration, since they play a vital part in the game's dynamics. As the head official, the umpire's duties include running the game and making wise calls on the field. This duty primarily entails using particular gestures to provide the players with important information. These gestures are essential to the game; they can be used to indicate dismissals or to judge whether a player has broken the rules regarding boundaries, wide balls, and no balls.

The importance of umpire gestures in cricket is examined in this study. It explores the domains of picture recognition and classification, using deep learning and artificial intelligence methods to precisely recognize and categorize these gestures and thereby underscoring their significance in the contemporary digital world of sports.

We apply a deep learning model that is essential to the cricket umpire gesture classification. The approach makes use of Convolutional Neural Networks (CNNs) to extract features from photos of umpire gestures, which allows for accurate classification of different actions and choices. Accurately mapping the retrieved features to the appropriate labels—such as out, not-out, wide, six, four, no-ball, no-action, leg-byes, and byes—requires a rigorous classification method. In real-time situations, these models are very good at identifying and deciphering umpire gestures. By enabling the smooth integration of deep learning technology, this research seeks to transform the interpretation of cricket umpire gestures. Through modern technical solutions, this integration opens the door for creative applications in sports analytics that improve players' comprehension and engagement.

Our primary objective is to accurately recognize umpire gesture images in cricket. We have expanded upon previous research by including a total of nine signals for recognition; whereas earlier studies typically considered only six. Our extended dataset now encompasses three additional gestures: byes, leg byes, and not out, alongside the previously recognized signals — six, four, wide, out, no action, and no ball.

To compile our dataset, we manually collected images by reviewing numerous cricket game plays and augmented this data with images sourced from Google. Further enrichment of our dataset was achieved through data augmentation techniques and the utilization of Generative Adversarial Networks (GANs) to generate additional images. Subsequently, we classified these images into distinct classes with corresponding labels and standardized the images by resizing and rescaling them.

For the recognition and classification of these umpire gesture images, we employed Convolutional Neural Networks (CNNs), which are adept at feature extraction and facilitate accurate predictions. Notably, our research includes a comparative analysis of the accuracy of these models. This approach distinguishes our work from existing literature, where researchers typically focus on either classification methodologies or deep learning techniques without offering comparative insights. Through this innovative approach, our study aims to contribute significantly to the field of sports analytics by enhancing the precision and reliability of umpire gesture recognition in cricket.

II. Literature Review

The automatic scorecard updating in cricket has led to exploring several approaches and methodologies.Firstly it has started with the traditional scoring method and then later on modernized to computerized systems .The challenges in updating automating cricket scorecard introduces machine learning mainly image classification for umpire gestures recognition . The challenges are discussed by L. Bhansali and M. Narvekar, A. Shahjalal et al., and D.T. John et al., discussing their limitations . The limitations and difficulties associated with this approaches are examined and suggest that to make use of support vector machines (SVMs) for gesture detection and the inception v3 image recognition neural networks which is presented by Vaishnavi K. Nair, Raakhi Rachel Jose, Parvathy B. Anil, Minnu Tom, and Lekshmy P.L. It discusses the steps involved in classifier and event detection . It highlights the approach for completely changing the automatic scoring approach with higher accuracy and efficiency. [1]

Kolekar et al., address the challenges of efficiently managing the cricket content in sports videos and organising it effectively.By acknowledging the difficulties associated with generating the highlights from unscripted sports videos may emphasize the special difficulties associated with cricket because of its variety happenings. A. Kokaram and P. Delacourt, Kamesh Namuduri, and Y. Senthil Kumar et al., emphasizing the difficulties involved in the current methods for event detection in the sports videos and it need some more specialized techniques for cricket video analysis .Kolekar et al., has proposed a unique method for that focuses on recognizing on umpire gestures to identify event detection in the cricket videos.It can be observed that the methods works very efficiently in identifying the important occurrences and encouraging outcomes for effective content management and highlight the generation in cricket videos. [2]

In ”Gesture Recognition Technology in Cricket Umpiring: A Vision-Based Approach,” states about how many gesture recognition techniques were employed.It explains about the gestures and the expressive motions of the body that are effectively used for communication , Turk and Pentland (1991) and Müller and Röder (1999) has an solid evidence to to support this.Pavlovic et al. (1997) and Starner and Pentland (1995) examines the use of gestures in human-computer interaction as it is integrated in our everyday devices like TVs, smartphones, and car dashboards. It explores the various gestures recognition technologies in different domains and incorporating ideas from the study of Li and Shah (1998) on assistive technology for the deaf, Cassell et al. (1999) on computer-mediated child interaction, and Picard and Picard (1997) on patient stress monitoring. It highlights the gesture recognition in sports investigated by Müller et al. (2005) in their analysis of martial arts gestures and by Chambers et al. (2012) in their identification of cricket official gestures.Chambers et al. (2012) by highlights the potential for automated methods to expedite the umpiring procedures. [3]

Dhanusha T. John, Kavya S. Kumar, Vaishak T. Nair, P. Visakh, and B. R. Poorna explores more about cricket umpiring and scoreboard management and addressing the challenges that inherit the procedures.They explain how important it is for cricket umpire to maintain the integrity of the game , make crucial decisions and ensure fair game play.They also cover the gender gap in cricket umpiring and the technologies used in umpiring. It examines the current technologies that are used in cricket umpiring ,highlighting the custom of using manual techniques to update scoreboards and make decisions. This manual method not only consumes time but also raise the chance of human error and which may effect the outcome of the match. Against the drawback the author uses the machine learning methods, deep neural networks and image processing to automate the scoreboard updates by the umpire gestures. [4]

John et al. has presented the method for automatically summarizing the cricket videos focusing the highlights for identification of significant events which are signaled by the umpire. Transfer learning is strategy used for umpire pose detection based cricket and generating the cricket highlights based on the gestures of the umpire. And in this process we build classifiers that identify the umpire gestures and categorize their positions after that event based summarization is performed on the video . The empirical findings shows the efficiency of the suggested approach that has achieved higher accuracy in umpire gesture recognition and event classification. The performance of the system is evaluated based on the ability to generate the accurate highlights. [5]

Video -based system is used for detecting main events in cricket and focusing the recognizing the umpire gestures and identifying the events like six, four, out , no ball and wide. Using 3D convolutional neural networks for event detection, the system extracts features from the umpire frames by using the computer vision algorithms. Previously the methods used for event detection are wearable sensors , image processing and CNNs . some methods like intensity projection profiles for gesture recognition and also hierarchical hidden markov models also. Umpire gesture identification has also made use of transfer learning methods like inception V3 and VGG19 which were pre-trained on imagenet. For summarizing videos we use visuals like bowling and score change detectors. Image -based systems mostly fail to capture the motion accurately this can lead to misclassifications. Video-based approaches have better performance when compared to image-based methods. [6]

Cricket is world wide globally recognized sport and there are challenges in analyzing the cricket videos due to their complexity , and there are some methodologies for automatically identifying the umpire frames in the cricket videos. The current approaches highlights the difficulties involved in analyzing the cricket videos including recording techniques and several objects and regions of interest. Multi-step procedure includes pre-processing the images and segmenting the colour images using the support vector machine (SVM) classification and extracting features using histogram oriented descriptors . This approach optimizes the input images and captures the unique visual characteristics of the umpire and successfully distinguishes between the umpire frames and non umpire frames. [7]

Recognizing the umpire gestures in cricket matches using pose estimation techniques has been developed. It highlights the umpire gestures and important moments in the game like wickets and penalties. Umpire gesture identification has depended on the attributes of the picture and straight forward classification algorithms , but they fail to take into account some body parts like orientation which are very essential for posture detection. Artificial neural networks and pose estimation techniques are used to recognize the umpire gestures By extracting the important key points from the images and giving them to ANN classifiers the proposed approach needs to improve the accuracy for the umpire gesture recognition. They also used min-max normalization to ensure consistent scaling of the keypoints and ANN classifier design with dense , batch normalization , activation and dropout layers. The approach suggest that experimental results has an 87% of classification accuracy for umpire posses. [8]

III. Objectives of Study

Our study's main goal is to improve cricket umpire gesture identification using sophisticated image processing methods. Our work aims to broaden the range of gestures that can be used in existing approaches. In the past, assessments have concentrated on a small group of six signals; yet, we understand the value of a more thorough approach. As a result, we included three more motions to our study: byes, leg byes, and not out. This extension seeks to enhance the precision and usefulness of gesture detection systems in live sporting situations in addition to offering a more thorough comprehension of umpire communications. By doing this, we want to close the gaps left by earlier studies and provide a more reliable system that responds to the complex requirements of cricket umpiring.

Furthermore, assessing and contrasting the performance of various machine learning models in relation to umpire gesture recognition is a key objective of this project. Our research suggests a comparison analysis to identify which approaches produce the most accurate results, whereas other studies have usually used either classification methods or deep learning techniques in isolation. Because convolutional neural networks (CNNs) are so good at classifying images and extracting features, we shall make use of them in this investigation. In addition to making theoretical contributions to the field of sports analytics, our work also improves the practical applications of the rules of sports by evaluating the performance of these models against one another.

IV. Data Collection

To process and deploy our applications, we need the image data for recognizing the gestures by umpires. We are using the data collected from the various sources such as kaggle, github and generated the images using the online platforms using GAN algorithms such as runway and manually collected various gestures from 100's of game plays.

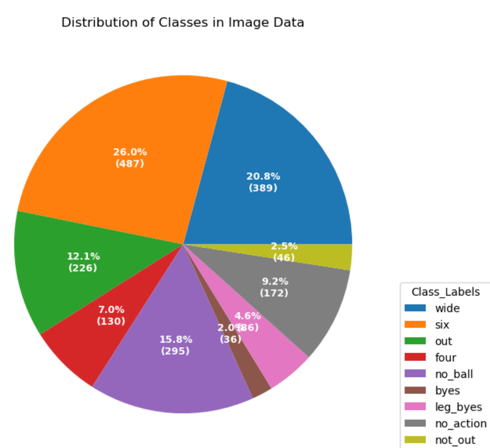

Fig. 1. Distribution of image classes for our data

The above **Fig. 1** displays the distribution of our image classes data.

Upon initially loading our image dataset, it became evident that the volume of data available was insufficient for optimal model training. To address this limitation and enhance the robustness of our training dataset, we employed data augmentation techniques. This approach enabled us to generate an additional 1,500 images, thereby enriching our dataset. As a result, the augmented dataset comprised a total of 3,500 images. This expansion significantly increased the diversity and quantity of the training data, which is critical for improving the accuracy and generalization of our deep learning model.

Data Augmentation : A technique used to create new data from existing data, which is especially useful for training machine learning (ML) models. These models perform best when they have access to large and diverse datasets, but obtaining such datasets can often be difficult due to issues like data privacy laws and other barriers. To overcome this, data augmentation makes slight modifications to the existing data to produce additional, unique data points. Recently, advanced generative artificial intelligence (AI) technologies have been developed to perform this data augmentation more efficiently and effectively, making it a valuable tool across different fields.

Up on accessing the link **Umpire gestures dataset** leads to the data, we have collected for total of 9 gestures.

Fig. 2. Distribution of image classes for our data

The above **Fig. 2** displays the sample classes of our image data.

In our classification models, we initially loaded the data and utilized the **Histogram of Oriented Gradients (HOG)** to extract features from the images, and passed these features and class labels as input to our models, where HOG is a sophisticated feature descriptor employed in computer vision and image processing for object detection, similar to other techniques such as the Canny Edge Detector and SIFT (Scale Invariant Feature Transform). This method involves counting the occurrences of gradient orientations within localized sections of an image, focusing primarily on the structure or shape of an object. Unlike simpler edge descriptors, HOG incorporates both the magnitude and direction of the gradient to compute more detailed and robust features. It generates histograms for image regions based on these gradients, thereby enhancing the effectiveness of object detection in our models.

V. Methodology

The process of recognizing umpire gestures is conducted using a dataset comprised of various umpire gesture images collected from different cricket matches. In this study, we employed four distinct algorithms to categorize the gestures into class labels and to evaluate the performance of each model. This structured approach allows for a systematic comparison of how effectively each algorithm performs in accurately classifying these gestures:

TABLE I
Umpire Gestures

Sno	Class Label
1	wide
2	six
3	four
4	not out
5	out
6	no ball
7	byes
8	leg byes
9	no action

The **Table. 1** displays the classes of our image data. After collecting the whole data, for our target variable, we have converted our labeled data to numerical data using the label encoding.

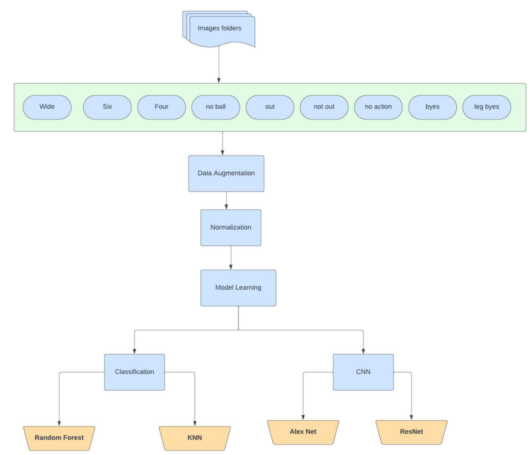

Fig. 3. Umpire Gesture Image Recognition Architecture

This section outlines a method for examining different gestures in cricket gameplay using a data-driven approach. We've developed a framework that incorporates image data to predict labels for cricket game images (as shown in Fig-3). The analysis process involves five main stages:

1) Load the image data
2) Data Augmentation
3) convert the labels into numeric using label encoding
4) Scaling the data
5) Apply or train Deep Learning and Classification models

The four models, we tried to deploy our application are:

1) **Convolutional Neural Networks**:
   a) AlexNet
   b) ResNet
2) **Classification models**:
   a) Random Forest
   b) K Nearest Neighbors

A. *AlexNet*

For our gestures images data, we tried to deploy or train our model using the AlexNet architecture. This model is most basic and famous one used in the ILSVRC contest, in which it totally predicted over total of 1000 classes. In specific, it has total of 65000 neurons per each hidden layer.

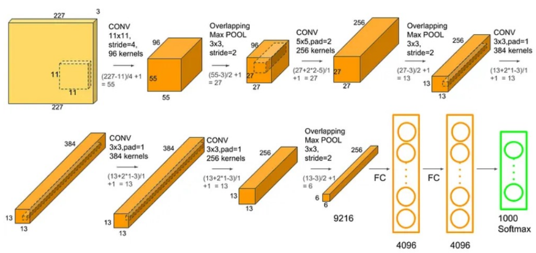

Fig. 4. AlexNet Architecture

The above Fig. 4 displays our AlexNet model architecture, which has total of 24 layers, to be precise, 5 convolutionary layers, 3 dense layers, 1 Flatten layer and Dropout to avoid the overfitting of our data.

Here, we started by providing the input as shape of (50,50,3) to our first layer with a total of 96 filters and kernel with size of (11,11) and strides of (4,4) and with the same width of padding. Followed by the second layer with 256 filters and (5,5) as kernel size. Then up on third and fourth layer, with 384 filters. At last, ending with the final fifth layer with 256 filters.

In every layer, we have used the ReLU(Rectified Linear Unit) as the activation function. One of the main advantage of the ReLu is, it does not activate all the neurons at the same time. It is mostly used for modern ANNs such as MLP and CNNs.

$$\text{ReLu}(x) = \max(0, x)$$

After applying the activation function, we have applied the batch normalization for every layer to maintain the uniform among the images data. Then followed by, applying the pooling to the layer, which reduces the overfitting, where even there is small invariance in previous data, it does not effect the output of the layers. For each of our inputs, there are certain weights w, applied for each input x.

$$\hat{Y} = \phi\left(b + \sum_i w_{ij} x_i\right)$$

For the output dense layer with 9 units(labels), we have applied the SoftMax activation function, which is commonly being used for Multi Class classification.

$$y = \frac{e^x}{\sum e^x}$$

For our model, we have applied the adam as optimizer, sparse categorical crossentropy as loss metric and accuracy as the performance metric.

Adam: is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iteratively based on training data. It combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

sparse categorical crossentropy: a loss function used when the training labels are provided as integers. It is particularly useful for multi-class classification problems where each output is mapped to a single class, significantly reducing the memory footprint when compared to standard categorical crossentropy, which requires labels to be provided as one-hot encoded vectors

Here, we have trained our model on batch size of 64 and total of 40 number of epochs. Up on training our model, we have performed the K cross Validation using the Early Stopping methodology to avoid over fitting process and with validation split of 10%.

B. *ResNet*

According to the analysis, we can see that, up on increasing the hidden layers, the accuracy needs to be increased, but for all the other models, we can see that, they have been continuously fluctuating over a period of time. Our ResNet model follows different approach, which is Skip/Residual connection, where for a single layer, it takes the output of the (n-1)th layer and passes up on to the (n+1)th layer as the input, which skips the layer to train,

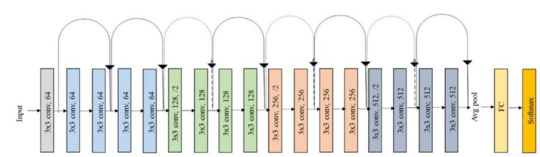

Fig. 5. ResNet Architecture

The above Fig. 5 displays the resnet architecture, which follows the Residual/Skip Connection methodology. It has a total of 27 layers, with 7 convolutional layers followed by Batch Normalization and ReLu as activation functionality with Pooling functionality as GlobalAveragePooling and 2 dense layers and output layer with 9 class labels. The input passed with the dimensions as (50,50,3) to the input layer of ResNet. Up on training the model with same metrics, such as Adam as the optimizer, sparse categorical crossentropy as loss and accuracy as metric to measure its performance. Followed by K cross validation using the Early Stopping as to avoid the over fitting process.

C. *Random Forest*

It is an ensemble learning technique for classification, regression, and other problems. It works by building a large number of decision trees during training and producing the class that represents the mean forecast of the trees, or the majority vote of the individual trees, in the case of regression. The fundamental idea behind Random Forest is to integrate the predictions of several decision trees to generate a prediction that is more reliable and accurate than it could be from any one tree alone. This is accomplished by training each tree in the forest on a random portion of the data and dividing the nodes using a random subset of characteristics. This brings diversity into the predictions and reduces the possibility of overfitting.

For our model, we have deployed, we trained with 42 as random state and n jobs to be run is 1. We have also tuned our parameters for the model better performance.

1) **max-depth**: 2,3,5,10,20
2) **max-features**: auto, sqrt
3) **min sample leaf**: 5,10,20,50,100,200
4) **n estimators**: 10,25,30,50,100,200

With the help of Grid Search CV, we have found the best parameters for training our model and then trained our model based on the best parameters and generating the classification report.

D. K Nearest Neighbors

It is a simple, yet powerful machine learning technique used for both classification and regression. It operates on a very straightforward principle: it predicts the label of a data point by looking at the 'k' closest labeled data points and taking a majority vote in the case of classification, or an average in the case of regression. This proximity is typically measured using a distance metric such as Euclidean distance. KNN is a type of instance-based or lazy learning algorithm where the function is only approximated locally and all computation is deferred until classification.

For our model, we have deployed, we trained with 42 as random state. We have also tuned our parameters for the model better performance.

1) **n-neighbors**: 3,5,7
2) **weights**: uniform, distance
3) **algorithm**: ball-tree, kd-tree, brute, auto
4) **leaf size**: 10,30,50
5) **metric**: minkowski, euclidean, manhattan
6) **p**: 1,2

1) Minkowski Distance:

$$d(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{1}$$

2) Euclidean Distance:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

3) Manhattan Distance:

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{3}$$

E. Tools and Techniques

For the umpire gesture image recognition process, we have used various modules, such as cv2 to import the images or load into the input data and followed by deep learning based tools such as tensorflow and keras for processing and training our models and matplotlib for various visualizations. For classification models, we used the sklearn module, which consists of Random Forest and KNN classifier models to train. At end, while loading the images as input, we have used the numpy, to convert our data into arrays, for the easy training process, where numpy follows numerical python methodology.

After training our model and measuring its performance, for the testing purpose, to deploy in our local machine, we used the DJango framework to display our front end working model to the audience.

VI. Results

Based on comparison of all the models, we can see that, ResNet model has performed better as compared with all other models.

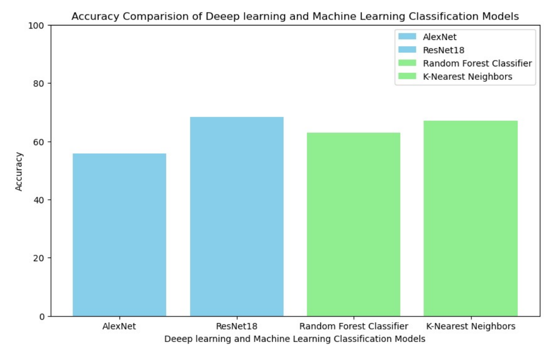

Fig. 6. Model comparison results

The above Fig. 6 displays the accuracy results of all our models, where the blue indicates the deep learning models and green indicates the classification models, we can see that, ResNet has out performed all other models with accuracy of 67%, followed by KNN with 63% and then Random Forest with 61% and at last, alex net with 53%. This image states that, deep learning models has out performed the classification models for predicting the image process.

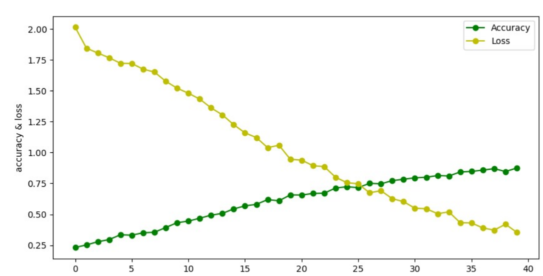

Fig. 7. AlexNet Accuracy v/s Loss

The above Fig. 7 indicates the comparison between the accuracy and loss for AlexNet model, which displays that accuracy is increasing according to the increasing number of epochs. We can see, there are some number of steps, between 15 - 20, where the loss is increasing and then decreased. In this region, where we need to analyze and re deploy our model according to the average epochs.

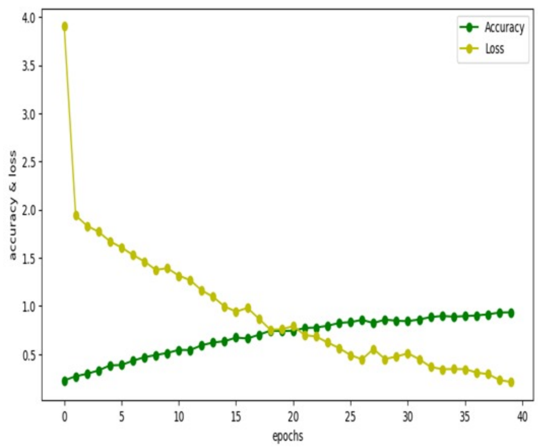

Fig. 8. ResNet Accuracy v/s Loss

The above Fig. 8 indicates the same accuracy v/s loss comparison for the ResNet architecture.

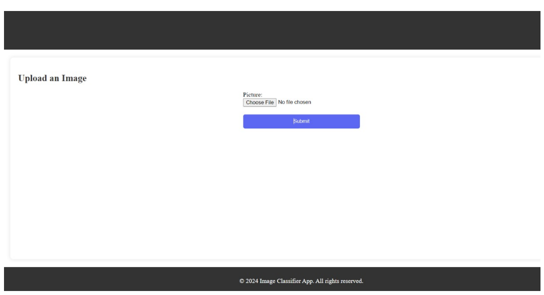

Fig. 9. Model Deployment

The above Fig. 9 displays our user interface model deployment in local for uploading the image to predict.

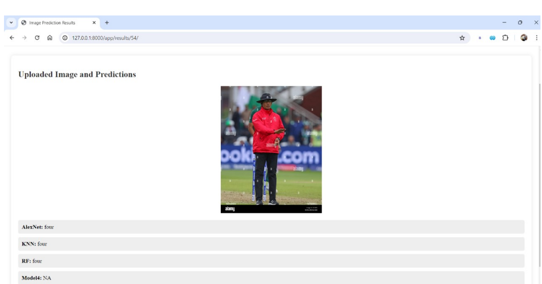

Fig. 10. Model Deployment Prediction results

The above Fig. 10 displays our various model predictions for the image input, we have provided.

VII. Conclusion

Our research talks mainly about the extensive use of convolutional neural networks(CNN) for better use of umpire gesture detection to reduce manual work, with models like ResNet and AlexNet models to show better results compared to general classification models. The CNN models performed better than classification models at processing complicated visual data, responding to differences in gesture expression and multiple variables as accuracy scores of CNN models are better compared to that of classification models. This ability provides better scope This difference underscores the ability of deep learning approaches to transform analyzing sports parameters in real-time contexts where fast and accurate decision-making is critical.

Not only the advantageous, but also, there are some of limitations to be applicable, By using different models we can have extensive scope of better accuracy while we are limited to only few models like RNN, decision Tree and naive bayes can't be used.

For the future work, we can make the video parsing and make the automated commentary system by using inception V3, and converting the video into various image frames, and up on predicting the labels and converting the output into speech using python text to speech algorithm.

References

[1] Nair, V. K., Jose, R. R., Anil, P. B., Tom, M., & PL, L. (2020). Automation of Cricket Scoreboard by Recognizing Umpire Gestures. International Journal of Innovative Science and Modern Engineering, 6, 1-7.

[2] Hari, R., & Wilscy, M. (2014, December). Event detection in cricket videos using intensity projection profile of Umpire gestures. In 2014 Annual IEEE India Conference (INDICON) (pp. 1-6). IEEE.

[3] Shahjalal, M. A., Ahmad, Z., Rayan, R., & Alam, L. (2017, December). An approach to automate the scorecard in cricket with computer vision and machine learning. In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-6). IEEE.

[4] John, D. T., Kumar, K. S., Nair, V. T., Visakh, P., & Poorna, B. R. (2019). Cricket scoreboard automation using umpire gestures. International Journal of Research in Engineering, Science and Management, 2(7).

[5] Ravi, A., Venugopal, H., Paul, S., & Tizhoosh, H. R. (2018, November). A dataset and preliminary results for umpire pose detection using SVM classification of deep features. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1396-1402). IEEE.

[6] Sakib, S., Mridha, M. A. H., Hasan, N., Habib, M. T., & Mahbub, M. S. (2022, December). Event Detection from Cricket Videos Using Video-Based CNN Classification of Umpire Signals. In 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-5). IEEE.

[7] Nandyal, S., & Kattimani, S. L. (2021, February). Vision based umpire detection method for event extraction in cricket video based on hog and color image segmentation. In 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-6). IEEE.

[8] Mili, R., Das, N. R., Tandon, A., Mokhtar, S., Mukherjee, I., & Paul, G. (2022, December). Pose Recognition in Cricket using Keypoints. In 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-5). IEEE.