Develop an accurate Model for Cricket Pose Estimation

1. Introduction

1. Overview

In general, human pose estimation has endless applications in almost every domain. It is a way of identifying and classifying the joints in the human body or capturing a set of coordinates for each joint (arm, head, legs, etc.) which is also known as a key point that can describe a pose of a person.

One of the most apparent dimensions applicable to pose estimation is tracking and measuring human activity and movement. Many architectures like OpenPose, PoseNet, and DensePose are often practiced for action, gesture, or gait recognition. We have used human pose estimation in the field of cricket for the following:

1. Identifying different batting shots
2. Check if a bowler has legal action or not.
3. Cricket Umpire signal Identification.

Identifying different batting shots

Some of the most common cricket shots are drive, cut, pull, flick, defense, and sweep. So we developed a Deep Learning model which will identify and analyze the cricket shot played by the batsman.

Check if a bowler has a legal action or not

In the sport of cricket, throwing, commonly referred to as chucking, is an illegal bowling action that occurs when a bowler bends the elbow joint over 15 degrees while delivering the ball. We developed a model which will

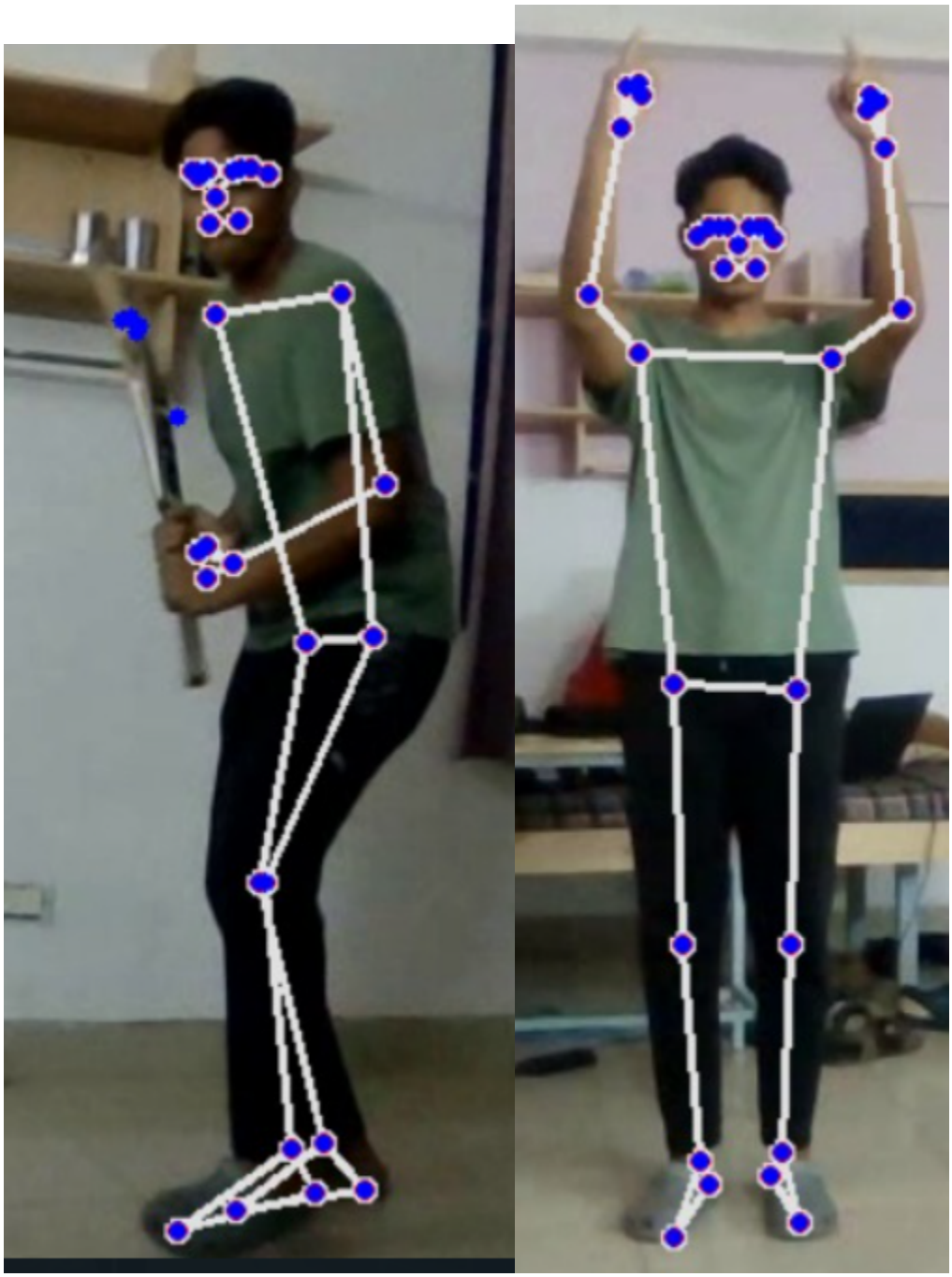
take the coordinates of the shoulder and the wrist, and calculate the angle between them. If the angle exceeds the above-mentioned angle, then it will be classified as an illegal action.

Cricket umpire signal identification

In the game of cricket, the umpire is the person with the authority to make important decisions about events on the field. The umpire signals these events using hand signals, poses, and gestures. Therefore, we developed a Deep Learning model to detect unique signals such as "six", "no ball", "out", and "wide".

With the help of these models, cricket association boards can use this for the betterment of analyzing the pose of cricketers while playing the game to identify the different poses such as cut, sweep, drive, bowling action, etc.

Below are the screenshots of cricket pose estimation by our model.



2. Purpose

- The innovation we propose in our solution is a special feature of commentary. After the model has classified the shot, it will display the name of the shot as well as pronounce the name of the shot. For example, if a batsman hits a straight drive, the output will be displayed as "straight

drive" and also a voice note would be played stating that "It is a straight drive". Similarly, this would also be applied to other models which will identify whether the action of the bowler is legal, and the signal of the umpire. This innovation would be a boon for visually impaired people who would like to know which shot is being played or what is the signal of the umpire.

- With the help of these models, cricket association boards can use this for the betterment of analyzing the pose of cricketers while playing the game to identify the different poses such as cut, sweep, drive, bowling action, etc.

2. Literature Survey

1. Existing Problem

The solution to this cricket pose estimation problem already exists, many have used Deep Learning models to solve this problem. One of the common solutions to this problem is to use a CNN network and classify images for pose detection, but this solution of image classification is not able to perform well on video data and real-time data. LSTM is one of the good solutions to this problem, the idea is to pass a single image frame sequence from the video as an input one by one up to 32 LSTM units (can have 32 or more or less than 32 LSTM (recurrent) units), and classify the sequence as per the input frames poses. Since 3D CNNs run one order of magnitude faster than both types of LSTM, their use is preferable. We argue that 3D CNNs' speed, early predictive power, and robustness should pave the way for their application in process outcome prediction.

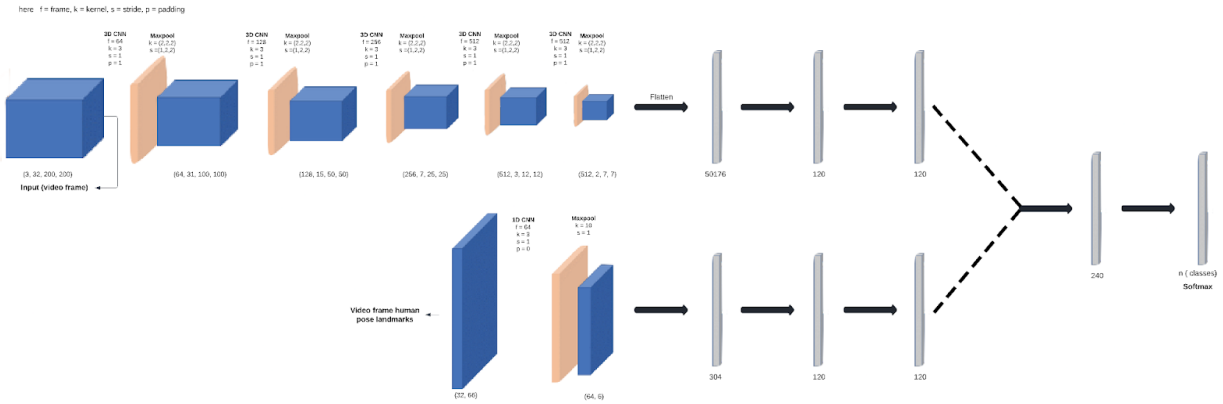
2. Proposed Solution

We created a hybrid model that contains a 3D Convolutional Neural network and a 1D Convolutional Neural network, 3D CNN consists of five 3D convolutional layers with kernel size 3 and with padding, and five Maxpooling layers without padding. It will take input as a video frame to

process, and the video will contain 32 frames, the size of the input video frame will be (3, 32, 200, 200). During the process, the size of each next output layer is half of the previous output layer, further, the layer will be flattened and then the flattened layer will be passed to the next three feed-forward layers.

The 1D CNN consists of one 1D Convolutional layer with kernel size 3 and without padding followed by one Maxpool layer without padding. It will take input as a 2D list of human body landmarks extracted from each frame of the video input. The size of the input will be (32, 66) where 66 are the separate x y pair of 33 landmarks which will be extracted with the help of the MediaPipe library. After passing the input through the 1D CNN layer, further, the layer will be flattened and the flattened layer will be passed to the next three feed-forward layers. The last layer of both networks will be of size 120, both layers will be concatenated and passed to the next feed-forward layer for classification.

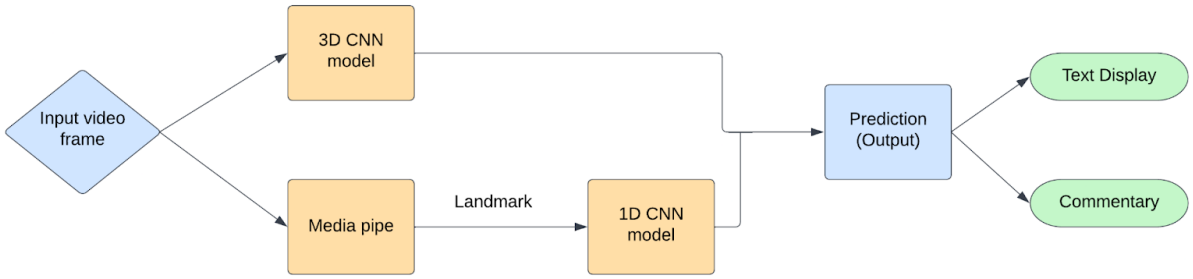
The final softmax layer is of the size of the number of classes, we are using 9 classes for both Batting and Umpiring. We are using Adam optimization which will make the algorithm converge towards the minima faster, and the Categorical cross-entropy loss function for multiclass classification.



Architectural Flow

3. **Theoretical Analysis**

1. Block Diagram



Overview of our Architecture

We have designed an architecture that is a hybrid model consisting of a 3D CNN and 1D CNN for real-time cricket pose detection. The idea is to create a 3D CNN model which will take input as a video frame, then further concatenate the last feed-forward layer of the 3D CNN model with the last feed-forward layer of the 1D CNN model which will take input as the list of human body landmarks of each input video frame. The model will take 32 frames for a single prediction. The addition of landmarks to the 3D CNN model will perform better than the LSTM model. This model can do cricket pose estimation for real-time data i.e. if you play a shot or an umpire gives a signal in front of the webcam then our model will be able to perform real-time detection of the shot or the signal respectively.

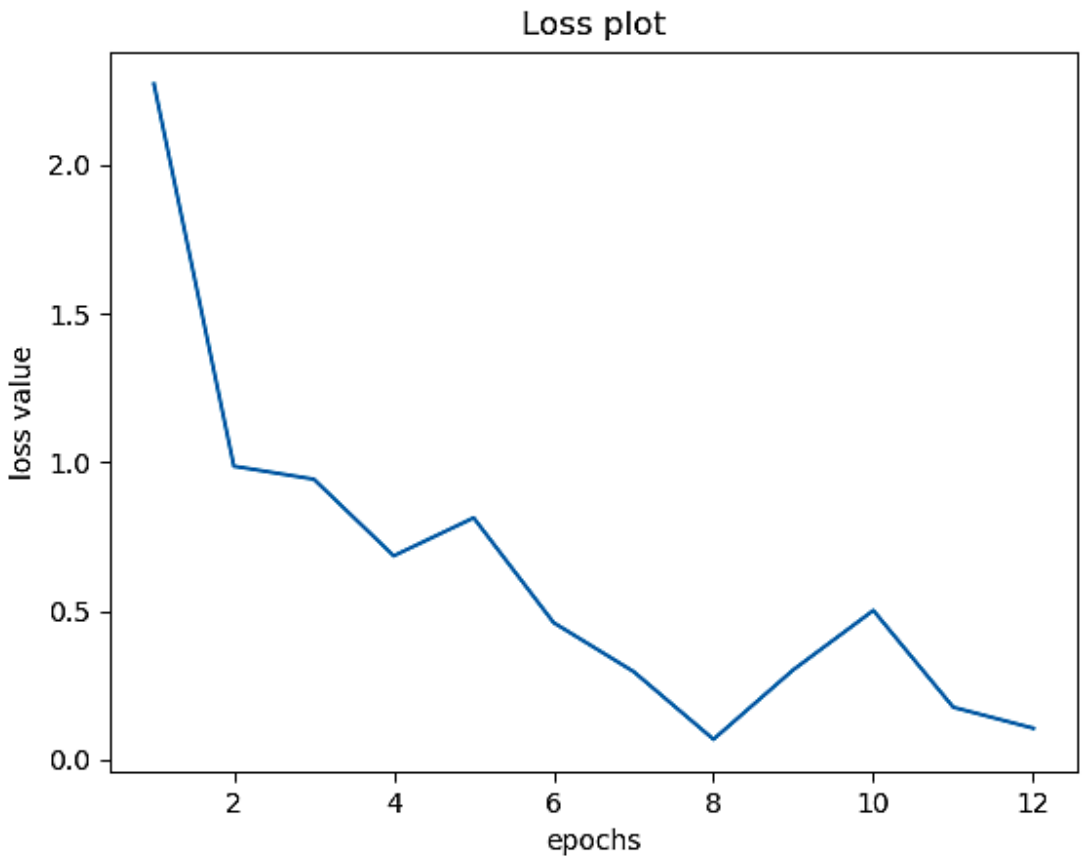
The innovation we propose in our solution is a special feature of commentary. After the model has classified the shot, it will display the name of the shot as well as pronounce the name of the shot. for example, if a batsman hits a straight drive, the output will be displayed as "straight drive" and also a voice note would be played stating that "It is a straight

drive". Similarly, this would also be applied to other models which will identify whether the action of the bowler is legal, and the signal of the umpire. This innovation would be a boon for visually impaired people who would like to know which shot is being played or what is the signal of the umpire.

2. Software Designing

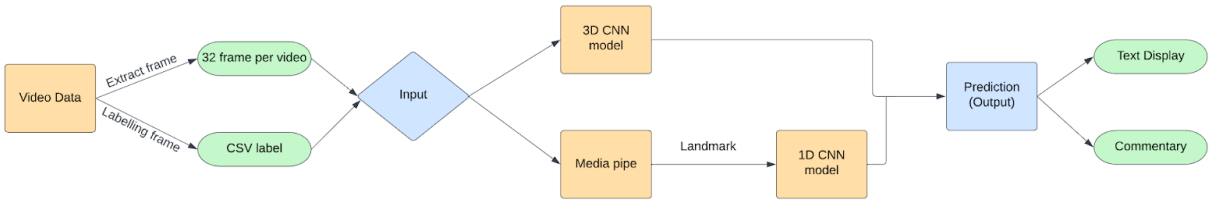
Programming Language - Python
Libraries/Frameworks - Pytorch, Numpy, OpenCV, MediaPipe, Pandas
Working platforms - IBM Cloud, Deep Learning-IBM Watson Studio
Dataset - Videos of various cricket shots and umpire signals

4. Experimental Investigations



The loss plot for the training.

5. **FlowChart**



Flowchart

Note: Define the scope of work to be implemented in the project with modules etc.

The proposed solution required development of the following modules :

1. **Data Extraction** - We recorded videos of cricket shots and umpiring actions for the dataset. Captured videos of different poses of different persons and store the data for further processing. The following modules were implemented for the data extraction.

   capture_data.py - This module is used to record the live video and store it in a specific folder. We did this with the help of OpenCV Library.

   extract frame.py - This module breaks down the captured video into 32-frame videos. Each video will be stored in a specific folder with a CSV file that will contain the path of every video.

   data.csv - This CSV file contains a video path along with the class label corresponding to each video. We labeled every video manually for creating our dataset.

2. **Data Preprocessing** - We used Pytorch for data processing. The video should be structured in a proper format for model training. The following modules were implemented for the data preprocessing.

Data_lit.py - This module contains a video_get() function which will convert the video data into a Pytorch tensor of size (no channels, frame, width, height), the default size is (3, 32, 200, 200), and it also extracts human body landmarks with the help of MediaPipe library which will be further converted into Pytorch tensor of size (frame, no of landmarks). This module will return the video tensor, landmark tensor, and labels corresponding to every video.

3. **Model** - We implemented a Deep Learning hybrid model consisting of the combination of 3D CNN and 1D CNN models.

   OneD_CNN.py - This is a 1D CNN model which will take human body landmarks as an input of size (frames, total landmarks). This model consists of one 1D CNN layer along with the maxpool layer, and three linear layers. The output of the model will be size 120. The output of this model will be used further in the 3D CNN module.

   ThreeD_CNN.py - This is a 3D CNN model which will take video as an input of size (no channels, frames, width, height). This model consists of five 3D CNN and maxpool layers along with the Batch Normalization and three linear layers. The output of this model will be concatenated with the output of the 1D CNN model, the size of the concatenated layer will be 240. The concatenated output will be passed through the final softmax layer of size 9 (number of classes) for classification.

4. **Training** - We combined all the modules for training the model. We used Pytorch for implementing and training the model. The following module we built for the training.

   Train.py - This consists of a DataLoader function which will be used for the data processing with the help data_lit.py module. The data will be

processed further into the CNN model and a Categorical Cross Entropy loss function will be used to optimize the Adam optimization function. We trained the model for 51 epochs. The model checkpoint will be saved during the training. Batch size of 1 is used for the training data.

5. **Testing** - We used a video for testing the model both for cricket shots and umpiring actions. The following module we implemented for the testing.

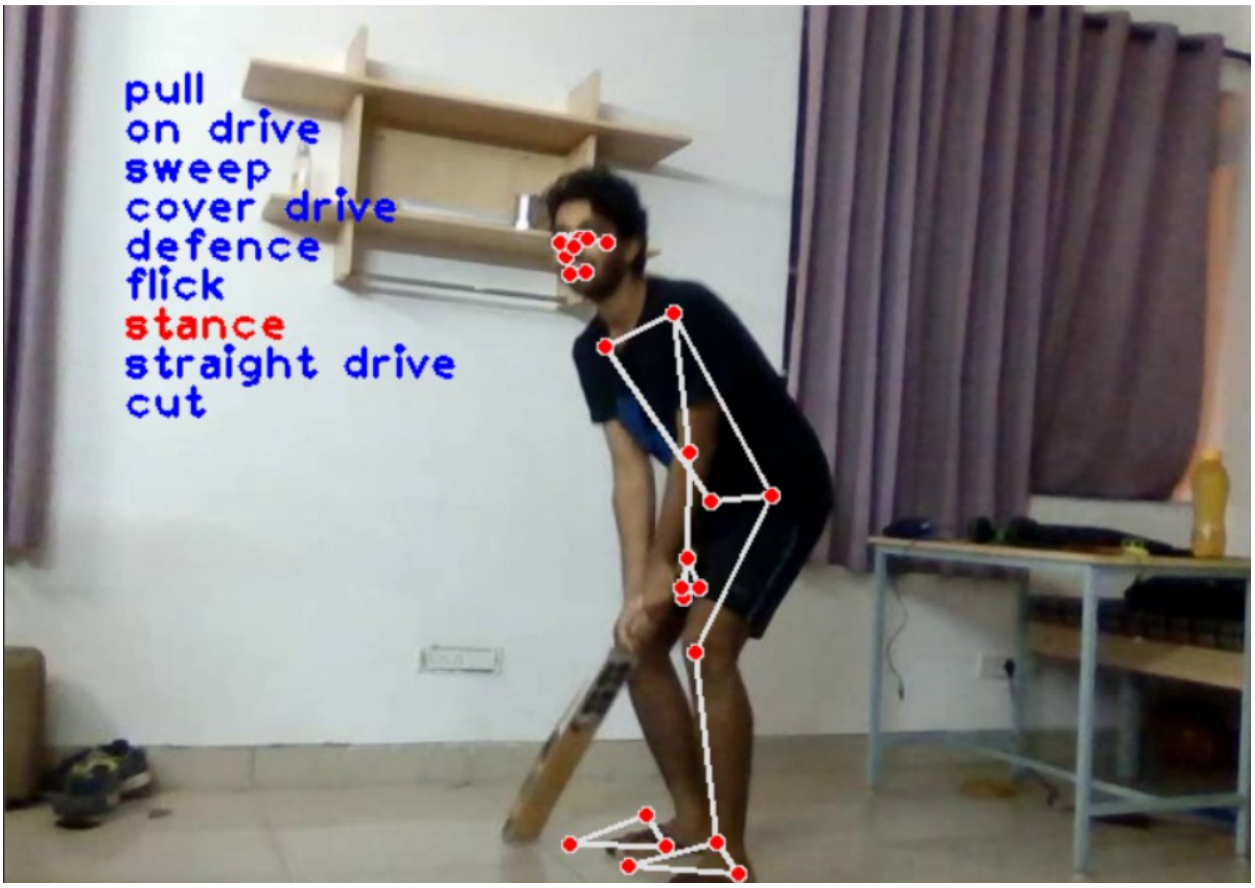
   test.py - This consists of a trained CNN model which will be used for the testing. For testing input, we are using a video consisting of cricket shot frames and umpiring actions frames. The module will highlight the action which will be taken by the person in the video according to the predicted class label.

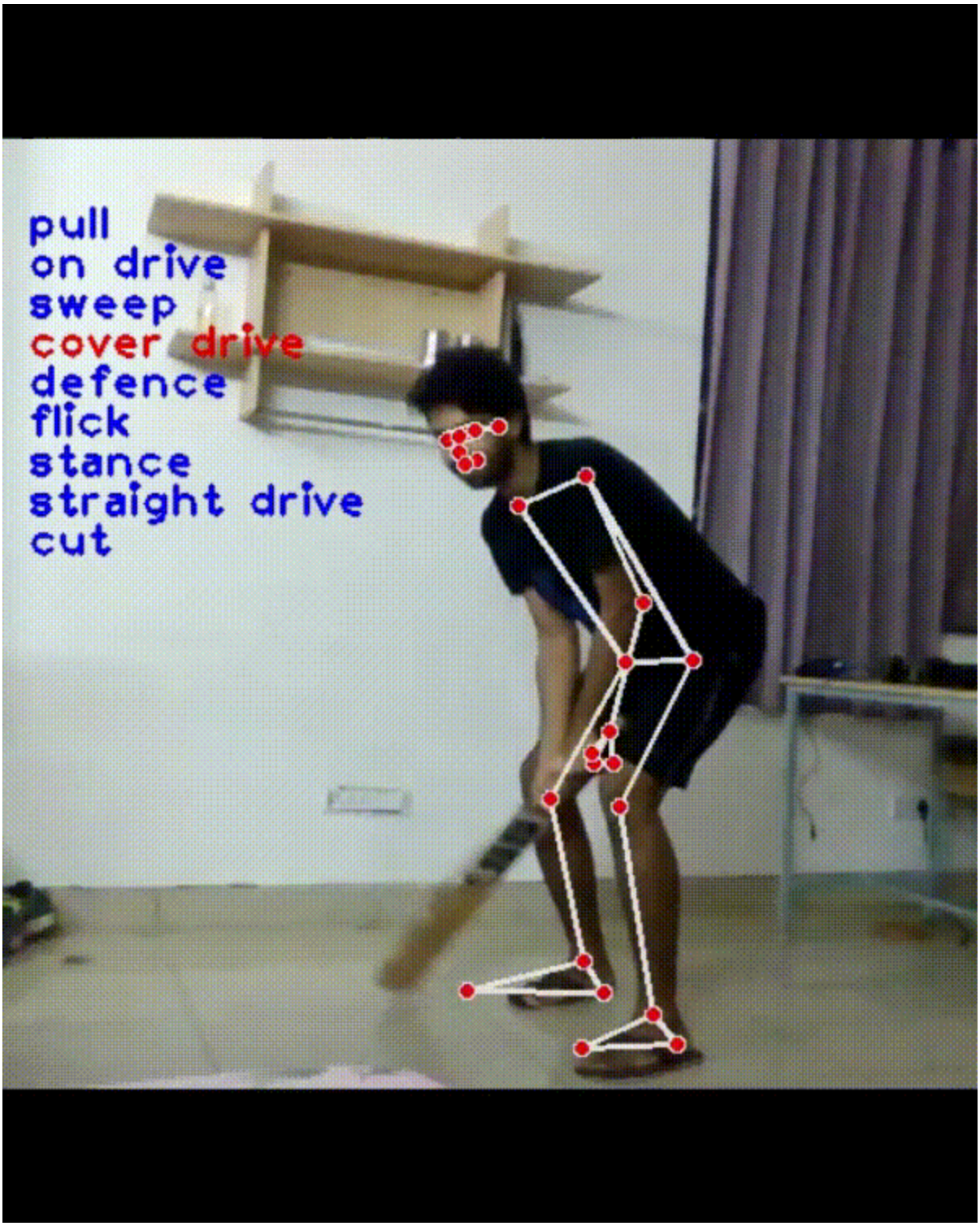
Workflow

6. **Result**


Wide
Bye
Free Hit
Out
None
Leg Bye
Six
Four
No Ball
adidas


Wide
Bye
Free Hit
Out
None
Leg Bye
Six
Four
No Ball


pull
on drive
sweep
cover drive
defence
flick
stance
straight drive
cut



7. **Advantages & Disadvantages**

Advantages :

- Applicable for pose estimation in three fields of cricket :
    - Batting (Cricket shot detection)
    - Bowling (Legal bowling action detection)
    - Umpiring (Umpire signal detection)

- Has an additional feature of commentary.
    - Visually impaired person can listen to the name of the shot/umpire's signal detected by the model.

- Capable of performing detection on video dataset and real time detection.

Disadvantages:

- Requires high end configuration systems.
- New or Creative shots can not be detected initially as they need to be added in the database for training.

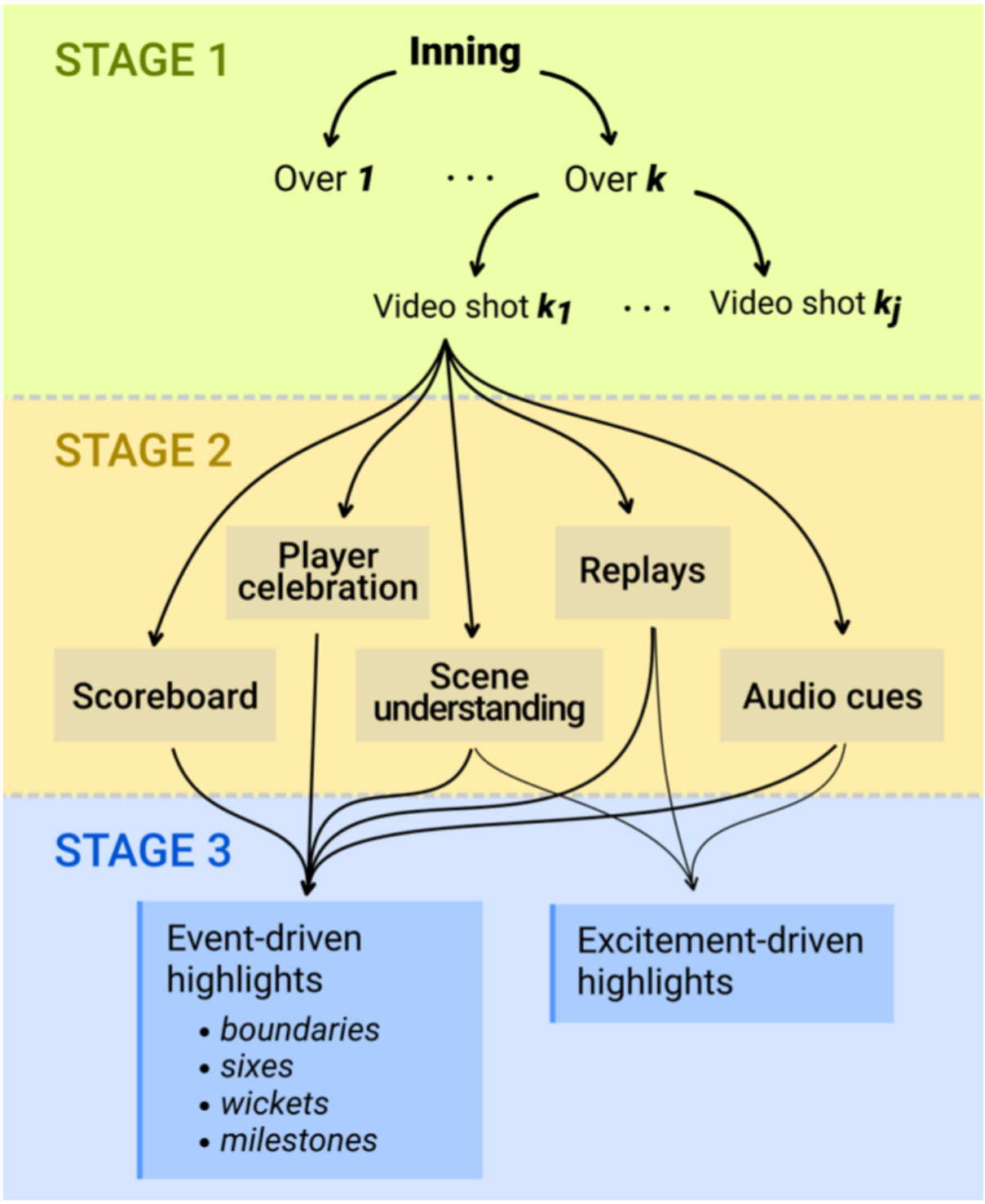
8. Applications

1. Automatic Cricket Highlight Generation

Producing sports highlights is a labor-intensive work that requires some degree of specialization. We propose a model capable of automatically generating sports high- lights with a focus on cricket. Cricket is a sport with a complex set of rules and is played for a longer time than most other sports. One of the primary reasons behind this is the outburst of sports media available on the internet. With many sports matches throughout the year, it is tough for a sports fan to keep up with all the news. Therefore, highlights serve as an essential source of information to keep fans updated with the latest happenings without consuming too much of their time. However, manual highlight generation requires professional editing skills and is a time-consuming process, which limits the amount of media that can be summarized on short notice. This fuels the need for systems

capable of automatically generating highlights of sports events, something that can leverage recent developments in machine learning, computer vision, and multimedia. The major contributions of our work are:

- We propose a first-of-a-kind system capable of automatically summarizing highlights of cricket matches, with results comparable to the highlights generated by professional manual approaches.
- Our method focuses on events as well as exciting features for highlight generation. This allows us to have better quality cricket highlights compared to previous approaches.
- We show that our system is very robust in terms of detecting essential events and that users support the quality of our highlights.

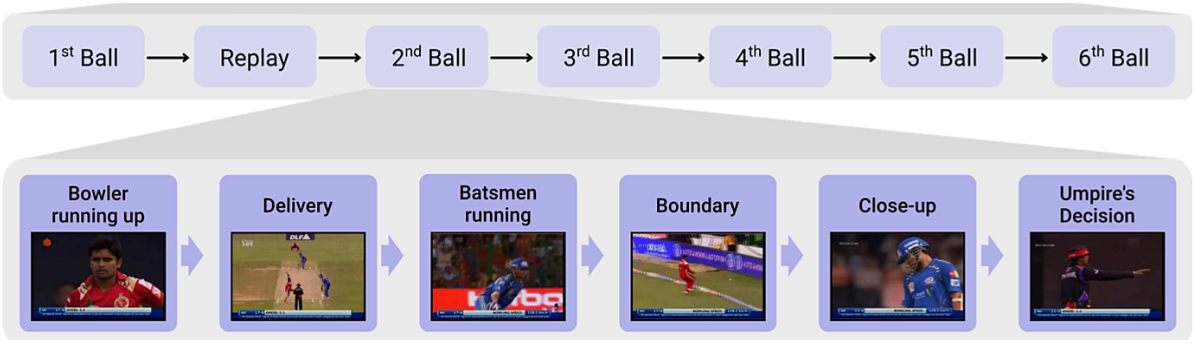


Block diagram representation of our model, which can be broken down into three stages. In the first one, the whole video clip is divided into a series of video shots. In the second stage, important cues are extracted

from these video shots. Finally, these cues are used for generating import highlights in the last stage.

Methodology-Video shot detection is an important aspect of sports video summarization. A complete cricket match can be broken down into a series of video shots, i.e., a set of frames that are part of the same camera. Therefore, events in a cricket match can be considered as sequences of video shots. An example of such a sequence is illustrated in the Figure below. There were two main reasons for choosing video shots as building blocks to generate highlights:

- Since highlights are generally a combination of essential events, video shots can be used for highlight generation.
- Processing video shots makes the model more efficient in terms of time. It must be taken into consideration that a cricket match might have a long duration. Therefore, applying sophisticated algorithms to individual frames will lead to an increase in the overall processing time of the framework. In this work, the first frame of each video shot is considered the keyframe that is used for representing the entire video shot, and all the processing is carried out on that keyframe.



An example of how video shots can be thought of as building blocks for a given over. An over in a cricket match refers to six legal deliveries that can be bowled consecutively by one bowler. These deliveries are made up of a

series of video shots. Thus, important events in a cricket match will be sequences of video shots.

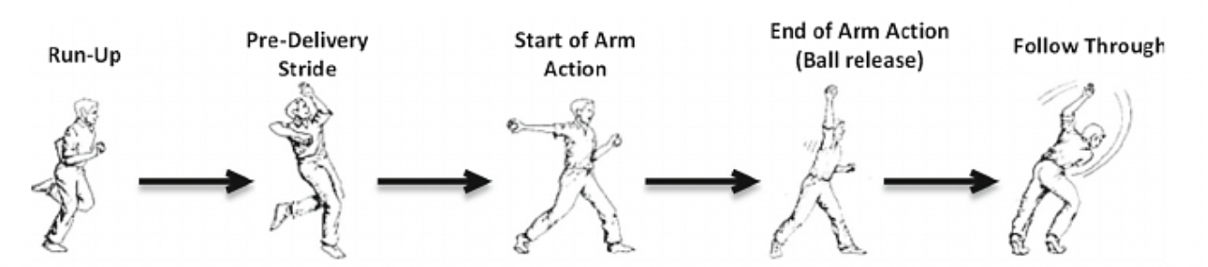
Concluding we could say that, We propose a new technique to automatically generate cricket highlights, focusing on event-driven. We showed that our framework can achieve comparable results to manual highlights and that it yields acceptable results for cricket fans. Overall, we demonstrated that by dividing a cricket match into video shots, we can create high-quality highlights without human supervision. Although there is still room for improvement, we also intend to extend this work to generate automatic captions for sports videos as well as for highlight clips of individual players.

2. Classification and legality analysis of bowling action in the game of cricket

One of the hot topics in the modern era of cricket is to decide whether the bowling action of a bowler is legal or not. Because of the complex biomechanical movement of the bowling arm, it is not possible for the on-?eld umpire to declare bowling action as legal or illegal. Inertial sensors are currently being used for activity recognition in cricket for the coaching of bowlers and detecting the legality of their moves since a well-trained and legal bowling action is highly signi?cant for the career of a cricket player. After extensive analysis and research, we present a system to detect the legality of bowling action. We propose a method to examine the movement of the bowling arm in the correct rotation order with a precise angle.

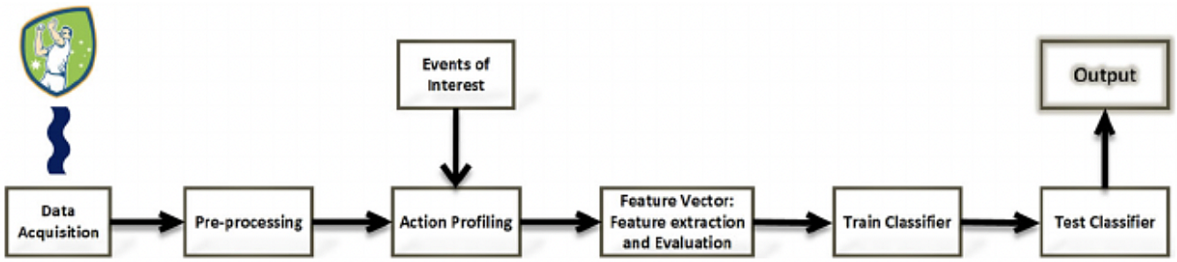
The system evaluates the bowling action using various actionprofiles. The action profiles are used so as to simplify the complex bio-mechanical movement of the bowling arm along with minimizing the size of the data provided by the classifier. The events of interest are identified and tagged

The idea of action profiling has been inspired by the concept of data profiling. Just like the analysis in data profiling is performed to assess the data quality for a data warehouse to simplify the content, relationships, and structure of the data in order to discover and validate metadata; action profiling is performed to simplify the complex bio-mechanical movement and clarify the relationship between different moving joints of the bowling arm to investigate whether the bowling arm conforms with the standards dictated by the regulating authority for a legal bowling action. We aim to develop different profiles of the events that can be used to assess the legality of the bowling action, as well as analyze the arm extension. The key events in a bowling action are run-up, pre-delivery stride, the start of the arm action, the end of the arm action (ball release), and follow through, as shown in Fig below.



Key events of bowling action.

Our process starts with collecting the bowler's data for the various bowling actions followed by pre-processing. The pre-processing steps involve calibration, removal of missing values, and outlier detection. This is then followed by action profiling, feature extraction, and evaluation. A classifier is then trained based on the training set to build a computational model for classification. A conceptual scheme of the classification process for the assessment of a given bowling action is shown in Fig below.



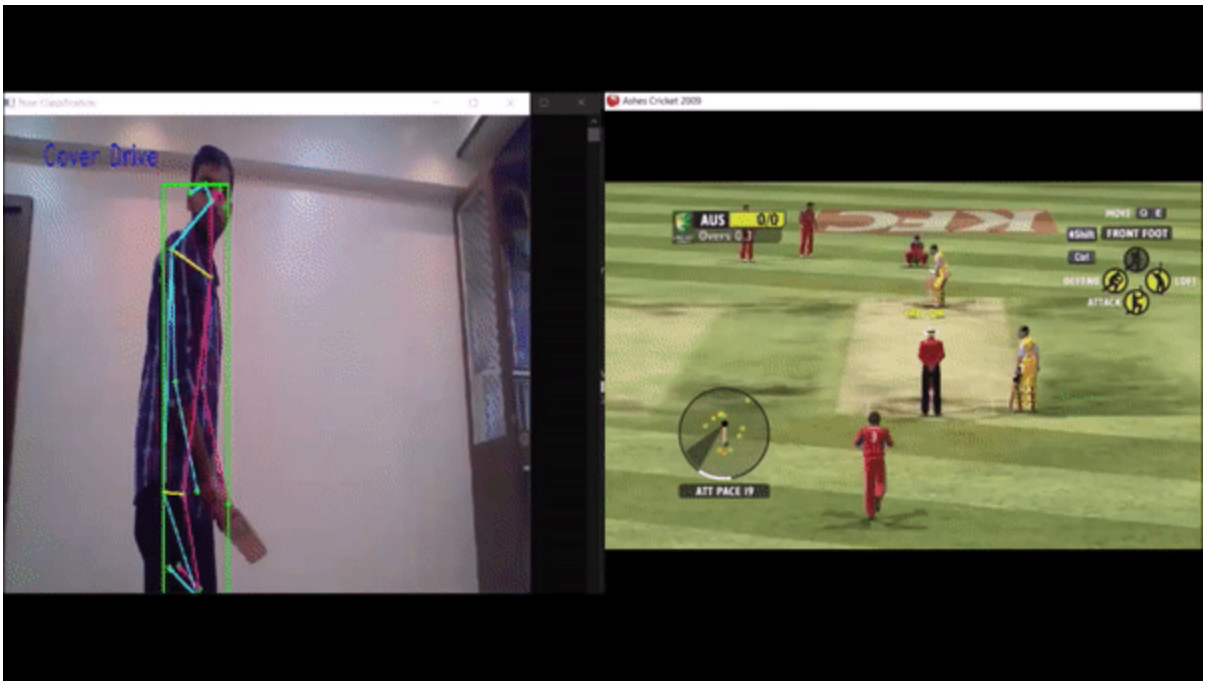
In Conclusion, we could say that this would be a very cost-effective and human error-free method and can be used by any agency around the globe to test the players. If everyone starts using this technology then this can be treated as a standardized testing method for legal balling actions.

3. Pose estimation in 360-degree simulator games

Research in the field of sports performance is constantly developing new technology to help extract meaningful data to aid in understanding a multitude of areas such as improving technical or motor performance. Video playback has previously been extensively used for exploring anticipatory behavior. However, when using such systems, perception is not active. This loses key information that only emerges from the dynamics of the action unfolding over time and the active perception of the observer. Virtual reality (VR) may be used to overcome such issues. We already have arcades all over India offering 360° cage cricket simulations but the problem with them is that they do not check the batsman's stance or pose while throwing a ball which may result in the batsman getting injured severely or throwing a ball that is far away from batsmen to solve these our model can be used to detect batsmen pose and good balls can be bowled thus helping the company with the security of its users. This is just software so it can be implemented on any existing software/hardware without any extra cost.

4. To Play Cricket Video Games

When you hit a shot in real life, the batsmen in the video game will play the same shot after the shot is detected by our model. This could be considered as both a business impact and a social impact. As a business impact, this will help us generate revenue. As a social impact, it brings youth back to the physical game of cricket from a virtual video game. It will help society, specifically the youth, as it will be a kind of exercise for them that is not possible through a simple virtual video game. To conclude we can say that it will have a positive impact on society by engaging the youth in physical activity.



9. **Conclusion**

To solve the problem of cricket pose estimation, we built a deep learning model in PyTorch which is able to predict real time cricket shot, check whether a bowling action is legal or not, and umpire signals and provides the output in the form of commentary. We have designed an architecture that is a hybrid model

consisting of a 3D CNN and 1D CNN for real-time cricket pose detection. The idea is to create a 3D CNN model which will take input as a video frame, then further concatenate the last feed-forward layer of the 3D CNN model with the last feed-forward layer of the 1D CNN model which will take input as the list of human body landmarks of each input video frame. The model will take 32 frames for a single prediction. Finally the output will be used to generate a commentary with the help of the playsound module.

10. Future Scope

We would train the model on a larger dataset so as to increase the accuracy and performance of the model.

11. Bibliography

1. https://arxiv.org/pdf/1709.07220.pdf
2. https://arxiv.org/pdf/1406.2984.pdf
3. https://arxiv.org/pdf/1406.2984.pdf

Solution Link

Video Proposal:
https://drive.google.com/file/d/1_SBRaYaie0T8xh2XEu2Dle80f2gz7JIx/view?usp=sharing

Essential Docs Link:
https://drive.google.com/drive/folders/1XECdTinAxcxuU0mq2YUtmRFGS_CdOA5K?usp=sharing