PROJECT KITTY

Yuvraj Singh Nirban
aka NovaPrime2077
hypernova2189@gmail.com

Abstract

Despite having a population of approximately 1.4 billion individuals, each having their own talents, India severely under-performs in sports tournaments like the Olympics, FIFA World Cup, etc. The model presented here uses the power of Machine Learning & Computer Vision to enhance the quality of sports technique of each aspiring Indian and likewise any person on the Internet. The model uses football as an example here, but with adequate data, this model can be used for any sports. The approach used here involves procuring the data from official matches that involve star players such as Lionel Messi, Cristiano Ronaldo, Kylian Mbappe, and many more. The model studies their shooting techniques in various situations during the penalties with varying camera angles, lighting,etc. Finally, the user uploads his own data to model, and a simple Euclidean norm between the user and the model's value predicts how much offset the user is from being a professional. Obviously, a threshold is applied to classify good penalty kicks.

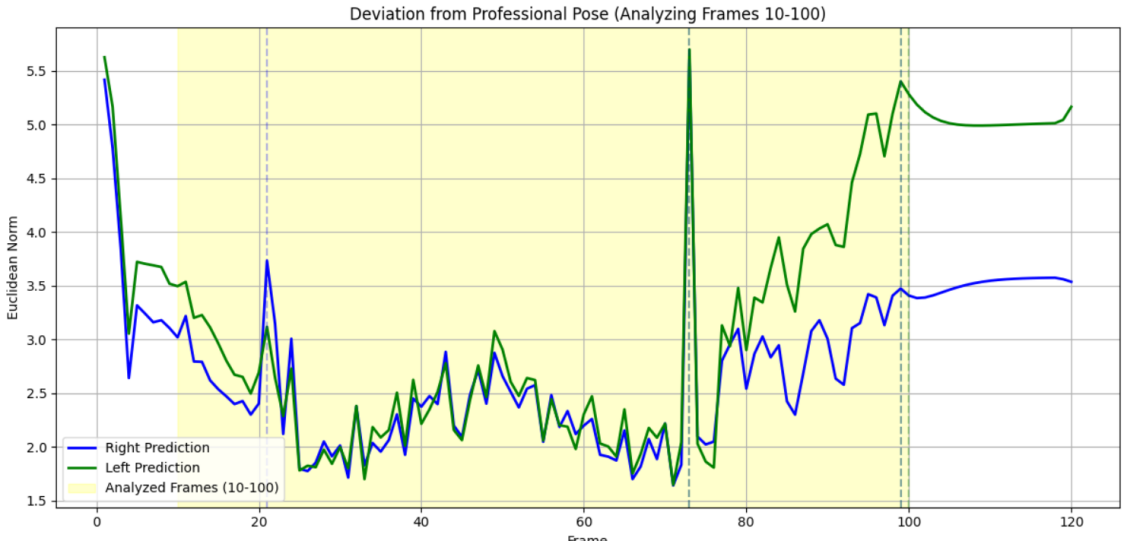

Figure 1: Model prediction of Harry Kane's dominant foot

Table of Contents

1. Introduction

Machine Learning has ventured into every field we know, the use of prediction algorithms is rapidly gaining popularity and success in almost every field it is employed. Here we present our **OffSide** model that brings the very idea into the football field. The flowchart presented below is the guide to the pipeline of **OffSide**.

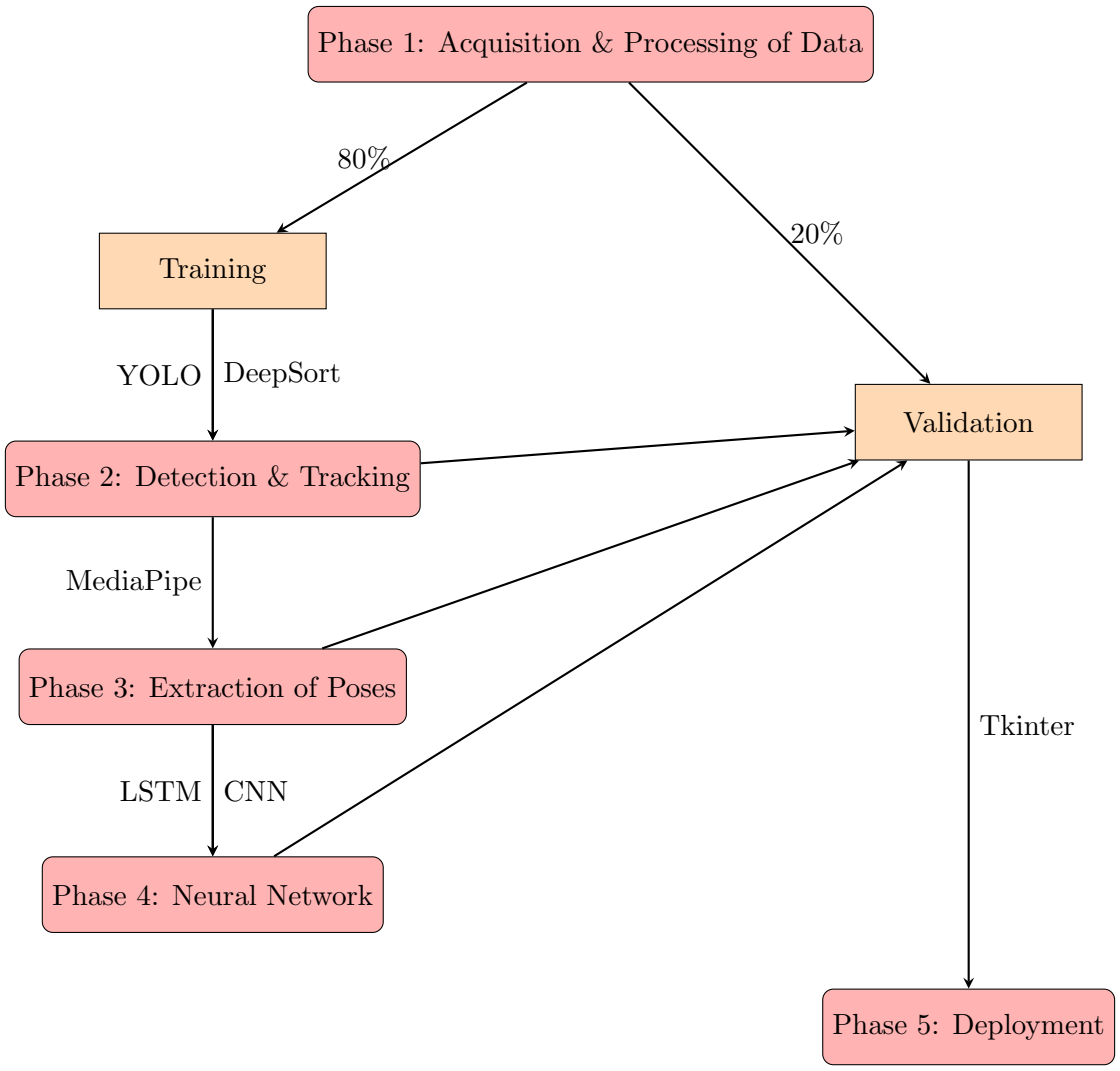

Figure 2: Pipeline of **OffSide**

2. Procuring of Data

2.1. Overview

The dataset of the model is created by acquiring different clips from the FIFA World Cups 2022,2018,2014, etc. The clips are chosen manually due to the limited amount of data available. The dataset is split into 2 different sets, i.e. the training and validation set.

2.2. Downloading

The important data are stored in the format of a CSV file. The CSV file contains all the necessary information to download the data. The raw videos are pulled from official handles like FIFA,etc. and are downloaded in a separate directory.

2.3. Region of Interest

The CSV file for each video contains the duration which is considered as region of interest, the region of interest is basically the duration in which the professional player's shooting action is taking place. A single raw video can have multiple regions of interest, The clips are then saved to another directory, and the previous raw video directory is emptied to ensure the memory does not get stacked due to unnecessary raw videos which have no further use.

2.4. Processing & augmentation of data

The model is fed data that consists of variable camera angles to ensure that it is robust to angle changes, while the data for the front angle are scarce, the side angles and the back angles dominate the dataset. Data Augmentation is performed to increase the data and robustness of the model.The dataset initially contained 74 unique examples from the three editions of the FIFA World Cup, but they are augmented to have a total of 148 examples with 74 examples for the left foot and right foot each; this essentially means that each clip was used twice once in the correct orientation and later in its "mirrored" version.

3. Approach

3.1. Overview

The approach to analyze these data first involves detection of all objects in the video, this is to ensure that all players in each frame of the video are detected and then they are further tracked across all the frames. The initial tracking is manual; i.e. the desired player is manually selected after the DeepSort assign id to each individual in the frame, The manually selected player is tracked across the remaining frames. The rectangular tracking box for the desired players is then analyzed and its data is stored in the ".npy" format for pose extraction from MediaPipe.

3.2. Player detection - YOLO

Figure 3: Player Detection, FIFA World Cup 2018: Argentina vs Nigeria

Player detection involves the usage of OpenCV[1] and YOLO[2] ("You Only Look Once") from Ultralytics. YOLO 11 is used for the current project, this model is based on Computer Vision and Deep Learning and is considered one of the best in the business when it comes to real-time object detection. The processed videos are loaded into the pre-trained YOLO 11 model using OpenCV after which object detection is carried out. In this stage, each person in the frame is enveloped by a rectangular box, which essentially is their detecting outline. Each person detected by the model has a confidence score, which implies how much certain the model is about the prediction. A threshold of generally higher than 0.5 is maintained in order to reduce the detection from the crowd itself, but still a very big problem persists! There are multiple players on the field, and therefore choosing our desired player is extremely important for the model.

[1]Official OpenCV documentation: `https://docs.opencv.org/4.x/index.html`
[2]Official YOLO documentation by Ultralytics: `https://docs.ultralytics.com`

3.3. Player tracking - DeepSort

The DeepSort[3] feature enables us to track player ids throughout the video once YOLO detects human-like objects. There are various ways by which we can actually track our desired player, like tracking their respective jerseys by designing a CNN for itself, but that is a project in itself, but for the current example manual selection was deemed optimal since the amount of data is not extremely large. Initially, all player's data is stored in ".npy" format across each frame, and

Figure 4: Player Tracking, FIFA World Cup 2022: Argentina vs Australia

the user inputs a manual id they want the model to run/test on, then, all coordinates of every detectable human-like object are stored, which is followed by user manually selecting the desired player. This desired player is the only one who has his poses extracted in the next step.

3.4. Pose tracking - Mediapipe

The next task after acquiring the coordinates of rectangular box of our desired player is to extract their pose during the penalty/free kick/shoot/etc. The following task uses MediaPipe[4] by Google. Figure 4 shows the location of the points the MediaPipe extracts, a very important aspect of dealing with the current problem is extracting all the x,y,z,visibility coordinates instead of just the regular x,y,visibility which the MediaPipe extracts by default, therefore the complexity of model is increased to 2 instead of using 1 since in the trade-off of time and accuracy we deem the accuracy much more important in making the model robust to camera angle changes which require the

[3]DeepSort documentation: `https://www.ikomia.ai/blog/deep-sort-object-tracking-guide`
[4]Official MediaPipe documentation: `https://ai.google.dev/edge/mediapipe/solutions/guide`

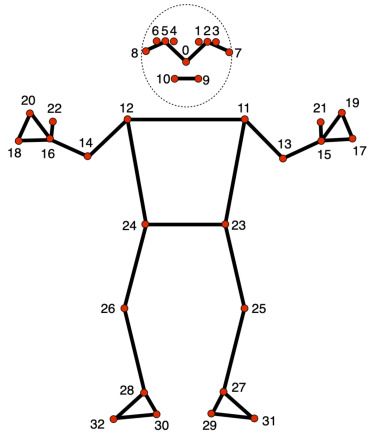

Figure 5: MediaPipe pose landmark detection

Figure 6: 3D Pose extraction, FIFA World Cup 2022: Argentina vs Australia

intervention of z which is the depth coordinate.

One might wonder how tracking all the x,y,z coordinates reduces the impact of varied camera angles? The answer lies in the fact that in 2D the model cannot generalize well on how the joints move across the frames from different camera angles, but this is not the case in 3D.Furthermore, MediaPipe's "z" is relative and normalized with respect to the hip point. This basically means that the "z" values are centered on the body, so the absolute distance from the camera is not known, but the relative depth between the joints is captured; this helps the model to generalize well even across varying camera angles. Finally, since we are dealing with the lower body here, only keypoints 23-32 are valuable to our analysis (see Figure 4).

3.5. LSTM & CNN

The human motion is actually spatio-temporal in nature, i.e. it depends on both spatial relationships between different joints of the body and temporal relationship in which a joint's position varies with time. The use of CNN in the model is to extract the spatial features across the frames, this is followed by extraction of temporal patterns from the frames using LSTM. The combination

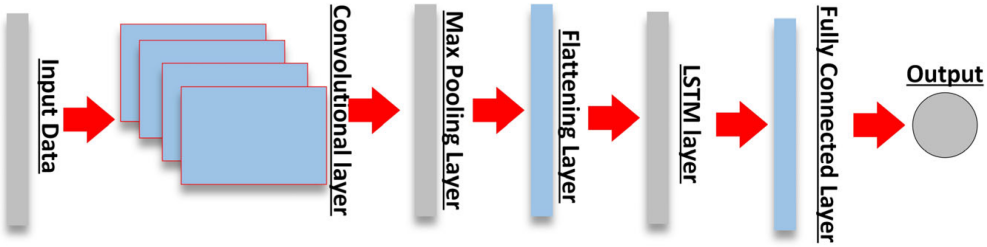

Figure 7: Architecture of LSTM & CNN Neural Network[5]

of CNN and LSTM ensures that the model is robust to noise and occlusions in the data. The input layer has a shape of (120,40) where 120 are frames in the video and 40 are features (basically 40 = features*keypoints, here features are the output(x, y, z, visibility) of the MediaPipe while keypoints are 23-32 points in the lower half of the body). This is followed by 2 convolutional layers to extract the spatial features and then 2 LSTM layers to extract the temporal features. The final layer of neural network is a dense layer that has all the features we need to evaluate how a professional kicks his goals. The last values are used to evaluate Euclidean norm and predict which leg was used to kick the ball and how much it is actually offset from the professionals who have the same dominant foot.

4. Training

The model is trained on the collected data. First, the data is split, 80% data for training the model and the remaining 20% for validating the model. Training is carried out in batch sizes of 8. Out of the 148 total items that were used for training some of them (2 to be precise were found empty due to too much occlusion and lighting problems that no id was properly tracked). The input shape for the neural network is (120,40), where 120 is the number of frames in the video, and since not all videos surely have 120 frames, padding is implemented to ensure that no shape mismatch is encountered.

5. Validation & Evaluation

The model is trained on the given dataset for roughly 20 epochs in which both the training and the validation error are measured, It was observed that when the data are mixed, that is both the

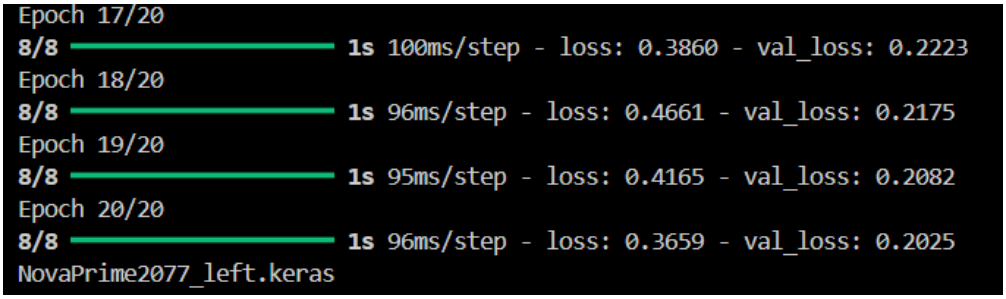

Figure 8: Training and Validation error of model trained on left leg

right and left leg's data were mixed and the model was trained on the entire dataset, the model did not generalize well like it did when the dataset was split between the dominant legs.

The final evaluation of the model takes place when it has to predict the result of a video provided to it. Since the model is designed to predict the professional sequence in each frame when it encounters a fairly new example, it tries to reconstruct the professional pose. Euclidean norm is taken out between the actual pose in the video and the estimated pose considering both cases (one in which the right foot is dominant and in the other left foot is). This is followed by summing

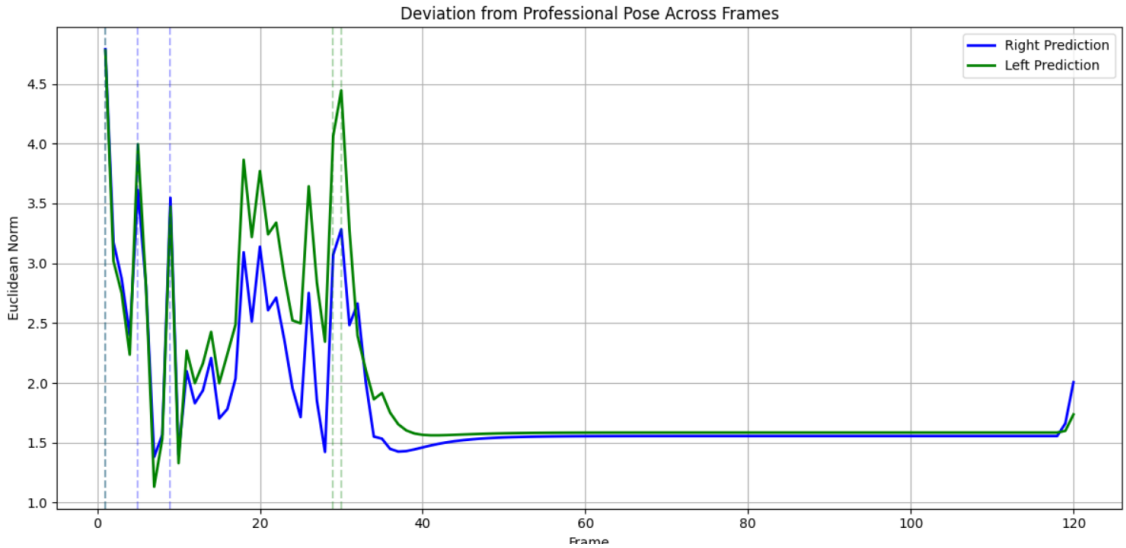

Figure 9: Total deviation (Right): 215.7280 & Total deviation (Left): 231.8698

of Euclidean norm values across the desired frames(Generally leaving out 10-20 frames from start and end since players tend to run and celebrate after the goal is scored and we don't want to add that celebration poses in the model), whichever leg's norm turns out to be lesser is predicted to be the dominant foot, finally if the difference in the actual pose and predicted pose is small, then the model predicts that the penalty is well taken, otherwise it outputs the input parameters which were quite different from each other in both the professional and the video.

6. Possible Improvements

The development of the model has taken roughly 10 days and I think there is enough scope to improve the model, and here I will list a few that I wanted to implement in the model, but was unable to due to both logistic and time constraints.

1. **Automatic Player selection:** The model in this instance has been trained using manual selection of the striker; a Convolution Neural Network (CNN) could have been used here to select the desired player by actually just incorporating their jersey numbers into the .csv files.

2. **Goalkeeper and other players:** Currently the model works only for penalties and I wanted to extend the model to goalkeeping, defense, etc. For instance in goalkeeping one can analyze how quickly goalkeeper moved before the striker shot the ball to analyze how moving before has a significant difference in the outcome.

3. **Deployment:** Currently the model is accessible through either the terminal or via the "tkinter file", I wanted to extend the model to a web app or a full-fledged desktop app to enhance its usability.

7. Conclusion

Project Kitty demonstrates how **OffSide** model based on Machine Learning and Computer Vision is capable of successfully detecting, tracking and extracting poses of players in real time. The power of the model arises for the combination of various start of the art models and algorithms like YOLO 11 by ultralytics, DeepSort and MediaPipe. **OffSide** is able to analyze the pose and movements of professionals of their field and can help individuals wanting to achieve something reach new heights.
The model demonstrates the power of automated player monitoring, which has significant implications for sports analytics, performance evaluation, etc. The modular architecture ensures scalability and flexibility, allowing the system to be extended to multiple players and different sports with small modifications. As with any model **OffSide** has its own challenges that lead to poor performance like bad lighting, scarce dataset, occlusions,etc. Despite all the challenges **OffSide** has proven the potential of A.I. driven sports analytics systems and can be the groundwork for future state of the art models which can take athlete analytics to newer heights.

8. Acknowledgments

During the journey of making **OffSide** I have taken the help of numerous documentation and LLMs and I think it is my duty and responsibility to mention each of the resources that have helped me in

the journey, and I express my gratitude to everyone and everything who/which helped in completion of the project.

1. LLMs like OpenAI's **ChatGPT** and **DeepSeek R1** were heavily used to identify bugs in the earlier stages of the code and especially during the neural network implementation where shape mismatches are quite common; both LLMs have also provided valuable insights about how to proceed in some tricky situations.

2. The official documentation of every module used in the project also contributed to the model.

3. The name **OffSide** was suggested by **Krish Agarwal**[6].

4. The report is written in Overleaf Premium which was provided by **Priyansh Meena**[7].

[6]Contact for Krish Agarwal: kkdon220@gmail.com
[7]Contact for Priyansh Meena: priyansh23006@gmail.com