Predicting the Outcomes of Penalty Kicks

Harrison Eckert, Maxwell Hardcastle, Kashish Pethani, Kyle St. Thomas

April 21, 2024

Abstract

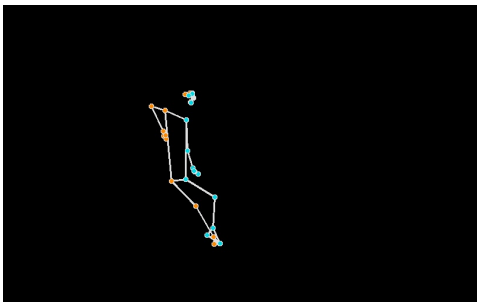
In this project, our group will attempt to train a machine learning model to accurately predict the result of a penalty kick in the sport of soccer. In order to accomplish this, we intend to use pose estimation as a framework to extract features that will then be used as training data. We set out to find if the data points correlated with the shot taker's designated lower-half body parts would prove to be significant in determining the outcome of the penalty. A neural network was trained using isolated frames from video clips that were deemed to be significant. Significant data in this case refers to data pulled from a sequence of frames that track the motion of the kick from pre-impact to follow through. Trained with said data, the model was able to achieve an mean absolute error of approximately 0.2568.

Introduction

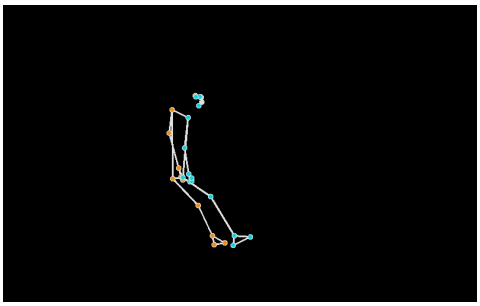
Soccer is the most popular sport with more than 4 billion fans around the globe. With an emergence of machine learning, we were curious in identifying which aspects of the sport could utilize a machine learning technique. A well trained model would outperform humans on certain tasks. Therefore, we decided to let the machine predict where the ball would go with just the images of the kicker during the penalty shootouts in soccer. To put it simply, the balls final position in the box can be predicted before hand while the ball is being kicked. From a practical stand point, if a goal keeper knows where the ball would go ahead of time, this will certainly boost the goal keeper's probability of blocking the strike successfully.

Methodology

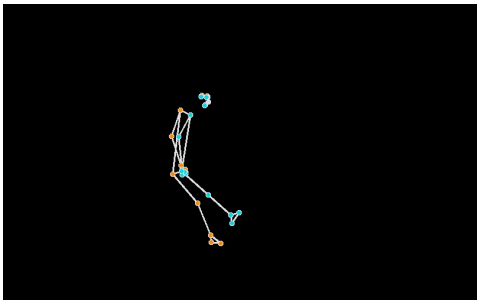
The pre-processing stage of the project began with the utilization of MediaPipe's pose estimation framework. Using the 'Heavy' version of the pre-trained model, which prioritizes performance at the expense of frames per second, body features were gleaned from each frame, and their respective data points were appended to a list. To ensure that pose data accurately corresponded to the body that appears in the frame, the model's outputs were mapped onto each frame and played back for analysis. Below are some example images outputted by the MediaPipe model. The first image represents the pre-impact frame from which each set of five data points begins. The rationale behind including this frame was that certain kicks may originate from similar foot positions. The image point is the output of the model for the frame in which the ball is struck. The last image is one of three follow through outputs. Three data points during this phase were deemed necessary as the natural motion of the foot following impact is crucial in determining where the ball will end up.



(a) Pre-Shot State



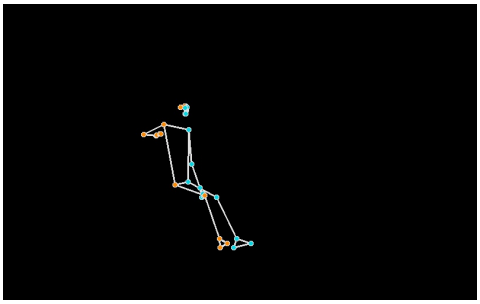
(b) Impact State



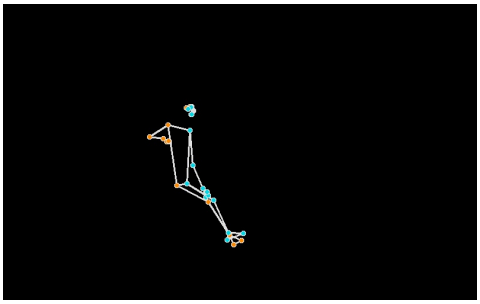
(a) Follow-Through State

Figure 1: Example Skeleton Data

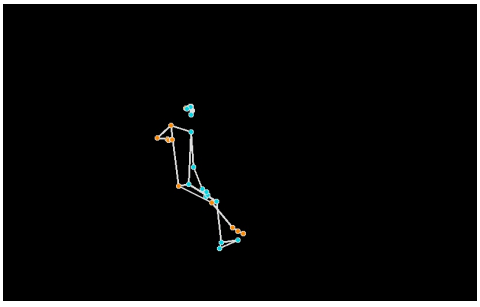
When the evolution of these outputs were deemed acceptable, lower-body data belonging to the aforementioned five frames were written to a csv file. In quite a few cases, the MediaPipe model would confuse lower-body data when the follow through resulted in the crossing of legs. An example of this occurrence can be seen below. The first two frames are acceptable; however, in the last frame, lower half data is swapped.



(b) Pre-Shot State



(c) Impact State



(a) Follow-Through State

Figure 2: Leg Swap Example

Data points belonging to the right side of the leg are written to the left side and vice-versa. These errors were noted as they arose and were amended down the line. In the end, 40 data points were extracted – each with five frames worth of pose data. Each row (frame) had eight columns representing the eight lower-body pose points – and every pose point had a normalized x, y, and z coordinate, along with two other superfluous coefficients. Regex was employed to clean up row data, resulting in each column having a list of solely x, y, and z float values.

Next, labels were extracted from each chosen penalty. In order to account for the differences in video size and camera zoom, the soccer ball's ending x and y coordinates were calculated using the dimensions of the goal frame. These figures were then written to a separate csv file. The same process of data and label extraction was then conducted to construct testing data. Due to limitation of real-life penalty videos that matched our strict criteria, videos showing penalty shots from the popular video game 'FIFA 24' were used.

Once training and testing data were appropriately packaged, the testing phase of the project began. Csv files were read into pandas DataFrame objects and then converted into torch tensor objects. In order to combine the subsets of penalty datum, a median value of x, y, and z coordinates were calculated. Training data was then inputted into our model, which consisted of six hidden activation layers. The first five activation layers utilized a ReLu function, while the last outputted results of a sigmoid function.

Model Parameters

With binary cross entropy as the loss function, and an Adam optimizer with a learning rate of 0.001, training began and continued over 1,500 epochs. Test data was inputted into the trained model and its outputs were then analyzed.

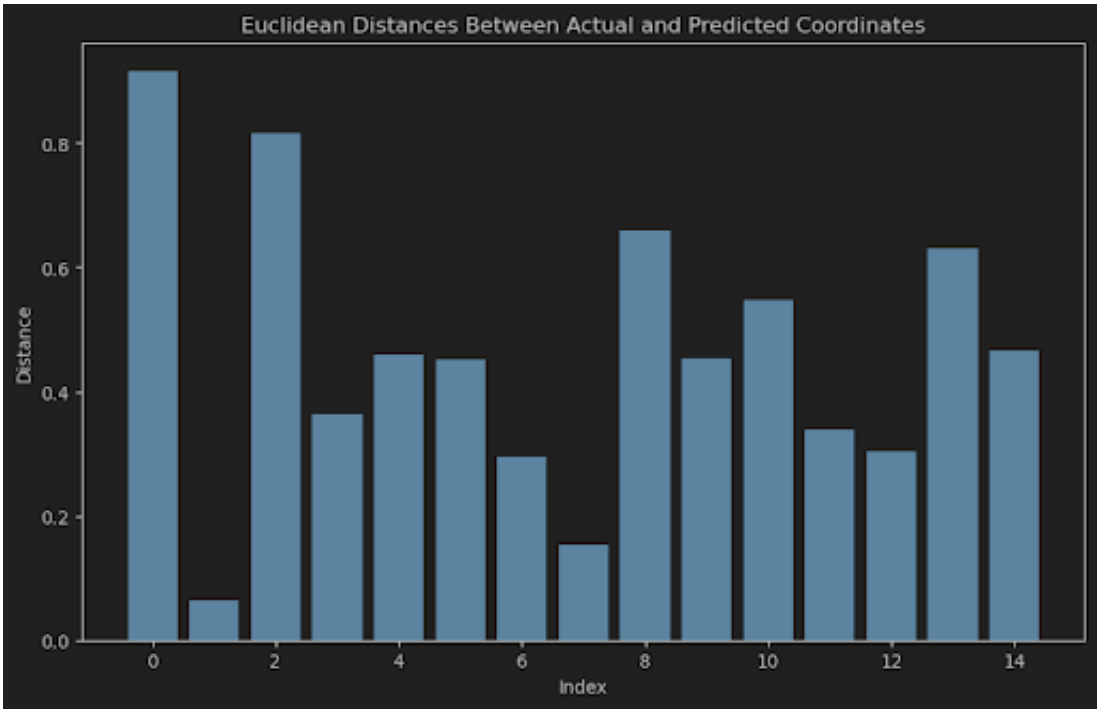
Results

The test dataset consisted of pose data belonging to 15 video clips and their designated labels. Data points were converted into torch tensors and inputted into the pre-trained neural network. Resulting outputs were then compared against their truth labels. Below is a reference table that compares the two labels.

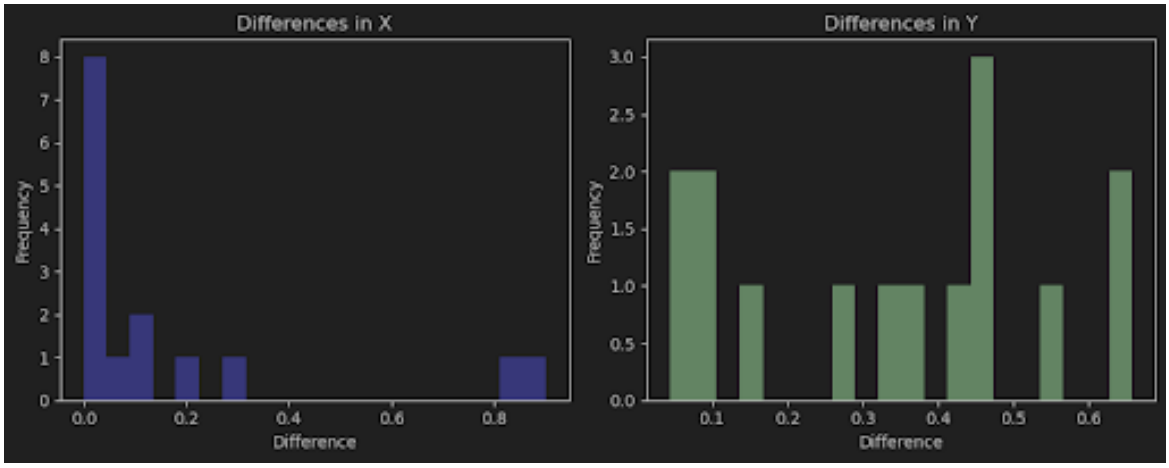
A quick eye test reveals that a majority of the predictions are not too far off of the actual truth values. To gauge accuracy on a more precise level, the mean absolute error was then calculated. An MAE of 0.2568 reveals that on average, predictions were roughly 0.26 units away from their actual values. Due to the fact that the majority of test data consisted of shots that traveled into one of the four corners of the goal, predicted results, on average, fell into the same quadrant of the goal as the actual result did. However, for two of the test data points, the predicted position fell near the left post, whereas the actual ball position was near the right post. Below is a graph that displays the euclidean distance between predicted and actual labels for each of the test data points.

Data Index	Predicted	Actual
0	(0.023864, 0.773612)	(0.925000, 0.609827)
1	(0.072074, 0.613262)	(0.040385, 0.556522)
2	(0.080199, 0.504139)	(0.891787, 0.580645)
3	(0.045580, 0.508559)	(0.064672, 0.144928)
4	(0.935653, 0.206494)	(0.935515, 0.665706)
5	(0.019637, 0.623198)	(0.051158, 0.173913)
6	(0.018990, 0.406400)	(0.078998, 0.694524)
7	(0.796262, 0.195412)	(0.923745, 0.281159)
8	(0.014773, 0.771681)	(0.034816, 0.113703)
9	(0.040066, 0.268157)	(0.135266, 0.711370)
10	(0.117276, 0.082170)	(0.095560, 0.628986)
11	(0.011499, 0.326487)	(0.047025, 0.662791)
12	(0.559170, 0.311000)	(0.860174, 0.266667)
13	(0.075837, 0.109680)	(0.102415, 0.739130)
14	(0.715412, 0.134962)	(0.914176, 0.556851)

Table 1: Comparison of predicted and actual labels



The data suggest that the model tended to do a poor job at accurately predicting the resulting position of the ball using pose estimation inputs. If one were to split the goal into four quadrants, most prediction coordinates would not lie in the same area as the actual coordinates. With that being said, splitting x and y estimates revealed an interesting phenomenon in regard to the model's capabilities. Below are two graphs that separate distances between predictions and truth labels based on their coordinates.



The graphs display that a large majority of predicted x coordinates were very close to their true x coordinate. The same can not be said for y coordinates, as there is a much greater variation between the two labels. Almost half of predicted y coordinates were off by more than 0.4 units. Further analysis reveals that the percentages of total variance explained by the error in x and y estimations are 41.67

Conclusion

Given the limited training data, our group found the results of our model to be promising. We are very confident that if we had the ability to find and process a greater number of penalty videos, our model would be very accurate. Our greatest obstacle to obtaining data was the method we chose to process the pose data, requiring us to manually vet each example. This is a problem that could very realistically be rectified in future work done on this project, by making sure that all of our data comes from a common source (FIFA video games for example) that has a much greater number of data points, and whose videos are all consistent with each other. We could also work on developing our own framework to take positional data points from videos, that is made specifically for the purpose we are using it for.

Individual Contributions

Harrison: Project design, slideshow, final report
Maxwell: Project design, final report
Kashish: Project design, model architecture, model training, final report
Kyle: Project proposal, pre-processing data, final report

Github Repository

Click Here.