KU LEUVEN

FACULTY OF SCIENCES

Penalty Kick Analysis: Visual Recognition, Pose Estimation, and LSTM

Gabriella Vinco

Supervisor: *Prof. Jesse Davis*

Co-supervisor: *Lotte Bransen*

Master thesis submitted in fulfillment of the requirements for the degree in Master of Science in Statistics and Data Science

2022-2023

Preface

This is for Boomer.

Summary

The penalty kick is one of the most dramatic events that can occur in a soccer game. It can be the source of heartbreak or the source of glory for millions of fans. With an event that can be game defining, only around 17.5% of penalties are saved. Coaches have long made the hypothesis that the direction the kicker will shoot the penalty kick can be predicted based on their body movements prior to the shot. Here we test this hypothesis. Are we able to determine the direction of a penalty shot based on the kicker's body movements before they make contact with the ball? This question is examined using three main concepts 1) object detection by means of YOLO v7, 2) pose estimation by means of YOLO Pose, and 3) categorical prediction through LSTM. By implementing this process on video data gathered from YouTube, we can extract pose data and track the kicker's movements over a sequence of frames. This sequence of skeletal keypoint coordinates can then be input in the LSTM to train a model that predicts the final direction of the kick whether it is "Center", "Left", or "Right". The predictions made by the model were then compared with a random generator intended to act as a goalkeeper randomly choosing a direction to dive. On average the random choosing goalkeeper had a predictive accuracy of 33.8% whereas the trained model had a predictive accuracy of 41.6%, displaying a 7.8% improvement in predictive accuracy.

Contents

Chapter 1: Introduction

The penalty kick, for decades, has been a game-defining event in soccer. It has been the source of heartbreak and glory for millions of people worldwide. For something so impactful, so much of the event is left up to chance. Having a background as a goalkeeper, coaches instruct you that there's a method to read the kicker's body movement as they are shooting the ball (Hutchins, 2022). Some coaches hypothesize that by watching the kicker's hips right before kicking the ball or by watching their planted foot you are able to determine the direction they intend to go. While these ideas and instructions have merit, in practical applications it's simply not feasible. The reaction time required to not only read the shooter, but also process and act accordingly is so small that it is impossible to react. This is a huge aspect of what makes penalty kicks so challenging and why goalkeepers only save around 17.5% of penalties (Vizcaino, 2021). After the 2022 World Cup is added to the list of 34 other World Cup matches that have been decided by penalties (Nag, 2023), it sparks the question: is it actually possible to predict the direction that a penalty is kicked based on the kicker's body movements? In addition, am I able to predict the direction consistently more accurately than it would be by randomly selecting a direction?

The ultimate task I want to achieve here is to determine if we are able to predict the direction of the penalty kick using the pose estimation data gathered from the videos. In this analysis, we extract the movement data from videos found on YouTube whether it's professional footage, semi-professional footage, or amateurs creating their content regarding penalty kicks. In the first portion of the analysis where we are focused on object detection, we will use YOLO v7 (Wang et al., 2022) to identify the kicker of the penalty kick and grab their location in the video frame by frame. The next step would be to then identify the skeletal key points of the kicker using a pose estimation library. In this analysis we used YOLO pose (Maji et al., 2022) to extract the key points. The third and final step would be to create a Long Short-Term Model (LSTM) (Olah, 2015) which is a form of Recurrent Neural Network (RNN) that is optimal for the type of data that we're working with. This specific type of neural network is optimal for processing, classification, and prediction of sequential data like

we have in the videos of these player’s movements. It has the benefits of a RNN without the vanishing gradient issue that affects long term memory. This means that we are able to take into account not only predictive features from the last few frames, but rather update the prediction from previous frames with each frame that's taken of the kick.

Chapter 2: Background

2.1 What is the Penalty Kick?

The penalty kick is much more complex than it first appears to be. However before starting the analysis it is important that we understand the fundamental rules about penalty kicks. There are two main situations where the penalty kick occurs:

1. A player commits one of the ten offenses or fouls listed inside their own penalty area
2. If the score is still level at the end of extra time, kicks from the penalty mark shall be taken to determine the winner (Depending on the competition's rules)

Once the penalty kick has been decided by the referee, the ball is placed on the penalty mark. This penalty mark is 11 meters (12 yards) away from the goal line. Below in Figure 1, the dimensions and measurements of the soccer field markings in a standard penalty area or 18 yard box are displayed.

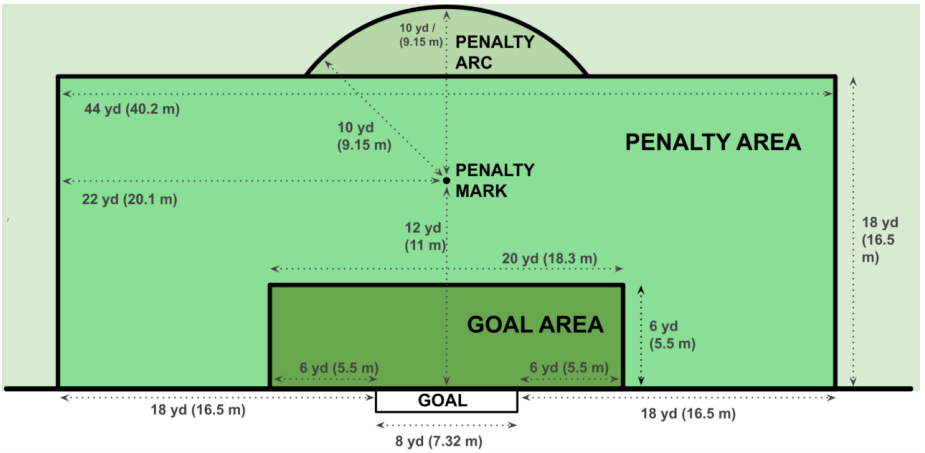


Figure 2.1: Distances of the marked points in a penalty box

The structure of this event is that there is a kicker lined up behind the ball ready to take the shot. Meanwhile the goalkeeper is standing on the goal line waiting in a ready stance for the kicker to shoot. The stipulation is that the goalkeeper must have at least one foot on the goal line until the ball has been kicked. All the other players

are to stay outside of both the penalty area as well as the penalty arc. The combination of not being able to close in the angle of the shot and having a direct ball kicked at them from so close is a huge disadvantage for the goalkeeper in this situation. The kicker has some limitations as well in that they must be moving forward towards the ball as well as shooting in a forwards direction. The kicker must wait until the referee indicates the shot can be taken, this is done by blowing the whistle, and after that the kicker is free to shoot (Federation Internationale de Football Association, 2015).

With penalty kicks being taken from such a close distance, the reaction time is very limited. The strongest shooters can kick at speeds of up to 80 mph. This means that the ball reaches the goal line in 500 milliseconds (John, 2022). A goalkeeper takes 600 milliseconds to move from the center of the 24-foot-wide goal to one of the posts (John, 2022). In addition, the average visual reaction time for an athletic person is roughly around 235 milliseconds (Jain, 2015). If the goalkeeper were to wait until the ball was kicked to react, the ball would already be halfway to the goal, and by the time they dove to the anticipated side the ball would be resting in the back of the net.

The implementation of penalty kick shootouts to resolve a tie at the end of the game was introduced during the 1978 World Cup. Since then penalty kick occurrences in World Cup Matches have nearly tripled, and with the addition of VAR technology, they have only increased more across all leagues and major competitions.

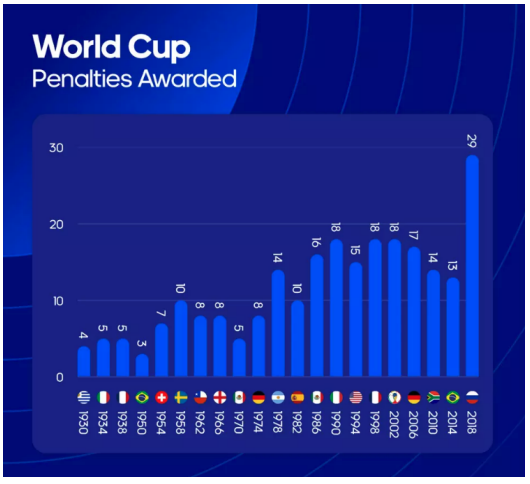


Figure 2.2: Graph Displaying the Amount of Penalties Taken in World Cups up until 2018

Since 2009 about 75.5% penalty kicks have resulted in goals, 17.5% were saved, the remaining percentages were not shot on frame (Vizcaino, 2021). An additional fun bit of information is that women goalkeepers save penalties at a higher rate, 17.75%, than men do at 17.55% (Vizcaino, 2021).

Another source shown subset in the table below reveals the conversion rate of penalty kick to goal alongside the average time in minutes for a penalty kick to occur. Some of the top European leagues along with a couple other leagues were selected to show the variance in the conversion rates. We find that the conversion rates fluctuate, but remain anywhere between 71 and 82%. A worldwide average score would be somewhere around 77% (CIES Football Observatory, n.d.).

Table 1: Conversion Rate for International Soccer Leagues

Minutes/Penalty	League	Conversion Rate
245	Ligue 1	82.2%
235	Serie A	81.3%
321	Premier League	80.3%
252	Pro League	78.5%
212	Liga MX	76.8%
252	Primera Division	74.9%

2.2 Object Detection Methods

This section describes the theory behind object detection as well as evaluates why this is appropriate for the use in this analysis. Object detection is a task within computer vision that seeks to identify and locate objects within an image or video

(Kukil et al. 2023). The type of object detection algorithm that is used in this analysis is a single shot detection that goes by the acronym YOLO (You Only Look Once) (Redmon et al., 2016) which performs object detection utilizing convolutional neural networks.While there are other methodologies like Single Shot MultiBox Detector (SSD) (Liu et al., 2016) or Faster Region-based Convolutional Neural Networks (R-CNN), these were not used for different reasons. SSD is faster than YOLO and Faster R-CNNs, but has a lower performance in terms of accuracy and also requires more data when training (Srivastava et al., 2021). A big issue I encountered was the limited time and data that I was able to collect. The model I create should be as close to real-time as possible while also remaining accurate. Then we examine Faster R-CNNs where it outputs a higher accuracy than the other methods, but its complexity is apparent in its speed (Srivastava et al., 2021). It requires multiple passes over the image unlike we see in YOLO and SSD being one stage detectors (Srivastava et al., 2021). Lastly there is YOLO, which while having its deficits is the option that I decided to use. While this section focuses on the general theories of the object detection techniques, later we will go further into the version of YOLO that was used and its added benefits from the original versions of this concept. Another huge reason for choosing YOLO over the other methods is that it's widely used and has a lot of documentation and support available to learn from online or when encountering issues.

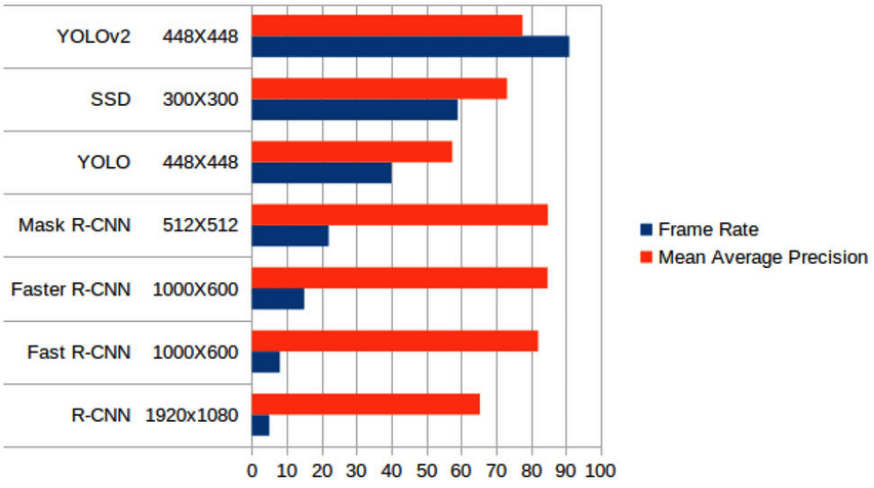


Figure 2.3: Comparison between object detection methods

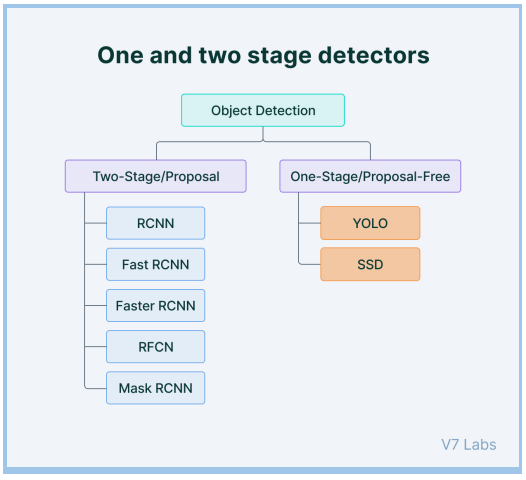


Figure 2.4: Examples of detectors separated by number of stage detectors

Being a one-stage detector, YOLO works by taking an image and segmenting it into a grid where each square calculates a bounding box with a confidence score between 0 and 1 and a class probability all within a single pass. This allows for less demanding computations as compared to two-stage detectors like Faster R-CNN (Srivastava et al., 2021). YOLO is able to maintain an acceptable balance between speed and accuracy. It works by predicting bounding boxes in terms of class probabilities directly from images in one pass or in terms of video data it would perform the same way just regarding each frame as an image. As seen in the horizontal bar plot above, it lacks Mean Average Precision, but in comparison with the other methods that are performing at the same level of precision, its speed is levels ahead. The biggest advantage YOLO provides is its speed which makes real-time processing achievable. Some other advantages in using YOLO for object detection is the simplicity in training because it doesn't require as much training data compared to other detectors such as Faster R-CNN (Srivastava et al., 2021).

When it comes to evaluating the accuracy of these models, there are two main evaluation methods. The first of those is called Intersection over Union (IoU) which searches to determine the localization accuracy and calculate the errors among the model (Srivastava et al., 2021). This works by comparing the bounding box defined in the annotated image alongside the model-predicted bounding box. The comparison happens by measuring the overlap between the two bounding boxes over the entire area of the two boxes combined to then get a ratio which is then the IOU value (Srivastava et al., 2021).

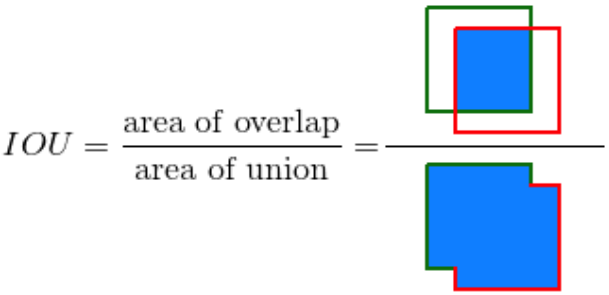


Figure 2.5: IoU Equation and Graphic

The other method of evaluation is average precision, which uses the area below the distribution of a precision and recall curve to provide an average precision by class for this specific model. Then that average can be taken from this average precision metric which then provides us with the mean average precision (mAP) for all the classes in the model. A distribution of where the areas taken to compute the average precision consists of two factors first is a ratio of how many predictions the model makes under a class over how many predictions are actually intended to be in that class. whereas precision is evaluating how many predictions are correct and positive and identified as positive over the total amount of predictions.

2.3 Pose Estimation using Yolo Pose

Out of all the procedures in this analysis, this is by far one of the more visually interesting ones to those who are still learning about computer vision. Pose estimation seeks to identify joints on the human body and plot it out on top of the image to reveal a skeleton like structure. However, there are several ways to go about estimating the skeleton and poses. When learning about this there are two main concepts, the top down approach and the bottom up approach.

The top down approach works first as a human detector, then once it identifies what it believes to be humans, it then predicts the joints within the bounding boxes. Some downsides of this approach are that the accuracy of the keypoints are heavily reliant on the bounding boxes being accurate (Maji et al., 2022). The other issue is if you are processing a video with multiple people in it, the processing time will be longer since all people need to be identified and then predicted individually. An analogy for this would be when looking in a microscope you first see the organism that you are searching for. Then once you have identified the whole organism, you are able to focus in and observe the body parts or detailed aspects.

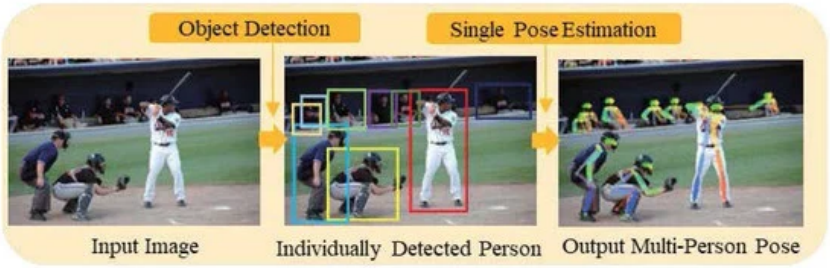


Figure 2.6: Step by step example of top-down approach

With the bottom up approach, the first aspect detected is the key points and then the key points are grouped and from there we are able to determine the skeleton of the entire human (Maji et al., 2022). This on the other hand would be similar to star gazing where you might identify parts of a constellation, but might not necessarily see the entire picture of the constellation. The drawback of this approach is that while it may be able to detect more complex poses, it suffers in misdetections and inaccuracies in grouping the skeleton together.

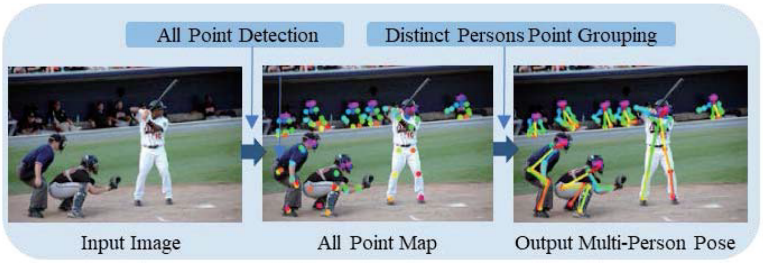


Figure 2.7: Step by step example of bottom-up approach

An advantage to YOLO Pose being a multi-person pose detector, if there were to be a goalkeeper picked up in the background, it wouldn't distort the skeleton and key points of the kicker. As we continue to develop a dataset of keypoints to train the LSTM on, maintaining the accurate skeleton of the kicker is one of the most important things to retain. The advantage that this method has in comparison to the bottom-up approach is that it doesn't require the pre-processing step to group detected key points into a skeleton shape since the bounding boxes provide an

inherent grouping of the key points automatically (Maji et al., 2022). And in comparison to top-down approaches, this method only requires a singular inference, rather than several forward passes since the people's bodies and their poses are all determined within a singular pass (Maji et al., 2022).

2.4 LSTM Prediction

LSTMs are a form of recurrent neural network (RNN) that as stated in the name accommodate both long and short-term memory within the model. This differs from typical recurrent neural networks because a downside of RNNs is that they struggle with long-term memory. This is due to the vanishing gradient problem which basically can be described as, the further back in back propagations the gradient converges to zero (Olah, 2015). LSTMs are specifically designed to avoid this problem allowing for the retention of both long and short-term memory.

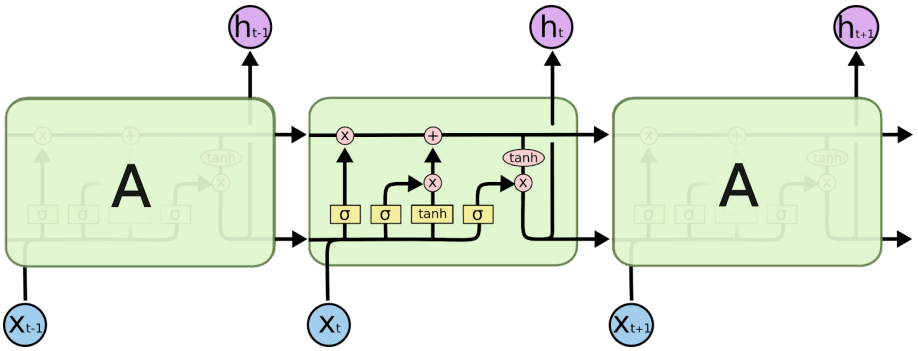


Figure 2.8: LSTM Diagram

LSTMs are made up of these four main components: the forget Gate, the input Gate, the cell state, and the output Gate. Going in order, the first part of the algorithm we will look at is the forget gate. The forget gate decides which information should be passed forward or which information should be forgotten. This happens by passing the information from the previous step through a sigmoid function where the output is a number between 0 and 1, which translates to forget or retain respectively (Olah, 2015). The forget gate can be seen as the chef who, depending on the order that comes in, chooses which ingredients are necessary to be able to make the dish. Should an order for fish come in, the chef wouldn't need to gather the ingredients to

make a pizza. They could take the ingredients like flour and mozzarella off of their mind and only retain the ingredients needed for this specific fish recipe.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

The input gate performs a similar task of passing information through a sigmoid function, but this time it's evaluating the information that the cell state is to be updated with (Olah, 2015). This can be thought of as when an order comes in, the head chef will assign the other chefs to take on certain aspects of the dish. For example, there may be one chef whose specialty is sauces or another whose specialty is pastry. The goal of the main chef is to choose one of his assistants that will work most efficiently and effectively to put out the best dish. This is the same idea as the input gate, where they evaluate the importance of the information that gets passed through to update the cell state and adjust accordingly.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

The cell state is the feature that puts the long term in LSTM. This is the feature that gets updated upon every time step on the information from the forget gate and the input gates (Olah, 2015). This information continues to get updated throughout the running of the algorithm and provides a sort of weight that is representative of the information that is relevant from previous time steps. This could be compared to the knowledge of recipes that you have gathered over your career as a chef. Depending on the ingredients you have available to you and the chefs that you have in your kitchen you have to adjust your recipe to accommodate what's available for you. Meaning that you still have the knowledge of techniques and overall cooking but you have to acclimate and adapt to the environment that you have around you at the time.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Lastly we look at the output gate, which is the final portion of the time step. This is where the long term information (cell state) and the short term information combine to create a hidden state to determine what information is to be passed on (Olah, 2015).

This is like the plating of the dish, where the chef combines the sauces with the pastry and the fish in the correct proportions to then be either served (prediction) or to be incorporated as an ingredient of a larger dish entirely (passed to the next time step).

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Common applications of LSTMs are used in language modeling, speech recognition, handwriting recognition, and also time series forecasting. However, given the data we are working with, the application in this analysis is ideal. Here we were able to take into account not only the movements that a kicker makes at the beginning of their run-up to the ball but allow it for reevaluation at every sequence or frame up until the kick.

Chapter 3: Methodology

3.1 Video Data Collection

In total 1308 clips were collected, but after processing and preparing the clips before the LSTM, only 792 clips remained. One of the requirements for the input of the LSTM was that it needed to be uniform in shape. During the object detection processing, all of the videos were set to the same frame rate of 30 frames per second. By limiting videos to 20 frames we could maintain the balance between having enough data and having long enough sequences to provide consistent results. Videos that were longer than 20 frames were trimmed and only the last 20 frames from the end were retained. This was possible because all of the videos prior to any form of analysis in this study were cropped at the moment of contact with the kicker's foot and the ball. So by collecting the last n frames the body movements just prior to the kick would be observable still regardless of how many frames were chosen to be retained. This process of limiting the amount of frames retained occurred during the data preparation stages prior to the LSTM.

All of the clips that were collected came from YouTube and ranged from footage from professional games and broadcast video to footage that amateur players recorded of themselves. The data was collected at several instances over a period of time to allow for new content to be uploaded to the platform. There was a limited selection of videos that fit the qualifications to start with so by checking periodically for new uploads and using keywords in different languages I was able to accumulate the dataset.

Benefits of collecting from YouTube are that it is an international platform with uploads from users all over the world. This allowed me to collect videos from not only the top European leagues but also leagues from India, Japan, Jamaica, and South Africa as well as content from both men's and women's soccer matches. It is hoped that this diversity in the data set will improve future predictions of this model and create more accurate results regardless of what gender or race the player in the

video is. In addition, having the variety of levels (pro, semi-pro, amateur) allows for the models to be trained on different video qualities to try and make it more robust. Fortunately, at this point in the merging between technology and soccer a lot of teams, even semi-professionals are recording matches and a lot of this data is readily available online. As computer vision and the implementation of technology in soccer continues to progress, the models will only be improved with the potential for many applications across various levels of play.

Another factor that was taken into account was the camera angles that were used to record the penalty kicks. For televised events there are usually set camera positions, but it tends to vary more with amateur shot footage. Regardless, all of the data collected was from the camera perspective of the kicker, meaning that the camera angles that were shot from behind the goal (seeing the goalkeeper's back, kicker's face) were not used. Only the angles where we were able to see the kicker's back and goalkeepers face or a perpendicular view. Examples of acceptable camera angles are shown below in figure 3.1.

Figure 3.1: Video Selection Angles

After the videos had been collected from YouTube, there was still a lot of cleaning that needed to be done. The videos were segmented clip by clip on Adobe Premiere Pro using a feature called Scene Edit Detection which looks for cuts in the video and segments the video into individual clips. The clips that contained the penalty kick and the movements leading up to it were retained and all others were removed. The clips

were then exported from Premiere Pro and ready for the next pre-processing step. This next step was crucial in the overall analysis, because here the clips were separated into labeled folders “Left”, “Center”, and “Right”. These labels are seen demonstrated in the figure below.

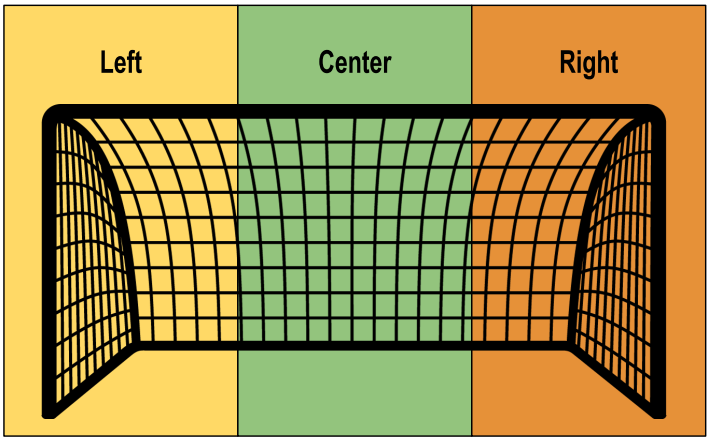


Figure 3.2: An Illustration of the Decision Areas for the Labels Used

This is the labeling that would be used as the Y data sets later on in the LSTM analysis. For simplicity I only used 3 labels, but it has the potential to grow and become more complex in the future should we want a more specified location. The direction labeling was performed subjectively and decided on based on several factors including but not limited to the visible area of where the ball crosses the line and the push-off visible in the goalkeeper’s legs. The videos were analyzed multiple times prior to confirming the labels and any videos that were questionable were slowed down and viewed frame by frame.

Once in the folders, the videos were trimmed down to the content that we wanted to observe, which were the movements prior to the kick. In the images below we can see a sample of the frames that make up one of the clips. Here we only wanted to retain the movements prior to the kick because the focus of the analysis is to use the prior movements to predict the direction of the ball after being shot. Any movements the kicker makes after the ball had been kicked would be useless for our prediction, so therefore only the frames prior to contact with the ball were retained. In figure 3.3

below, the frames that are in the green tinted cells are an example of the frames that were retained, whereas the frames that are in the red tinted cells would be the ones that were cut out of the clip.

Figure 3.3: Example of frames from a clipped video

2.1 Image Data Collection

In addition to the videos that were collected for this project, there were also 483 images that were scraped from Google Images to train the object detection model. This data was collected separately from the video data collection because I wanted the entire analysis process to be able to run on whatever data was presented to it

rather than just optimized for the data collected specifically for this study. Should the object detection model have been trained on the same data it was eventually predicting we would see high average precision, but this wouldn't be the true measure of the model. Only by training a model with data independent from its evaluation data could we reliably evaluate it.

The data was uploaded onto RoboFlow to annotate the images with each of the classes that I wanted to train the model on: kicker, goalkeeper, referee, and player. I made the decision to annotate all of the people on the field to try and increase accuracy as much as possible and to try and eliminate any misclassifications (Abramovich 2019). This data was then split for cross validation dedicating 61% (n=294) to the training set, 21% (n=101) to the validation set, and 18% (n=88) to the test set.

The total amount that was accumulated was 869 images, which consisted of the 483 originals as well as 486 augmented versions of those original images. The addition of augmented images came from the acknowledgment that there are situations where some games were filmed at night under good vivid lighting, some were filmed in the day, some were lower quality videos, and varied in saturation. Especially when dealing with video data from all different levels of production, this step is included to try and replicate some of the flaws that might be apparent in the video data. The augmentations I selected to use on the data were horizontal flipping, 25% increase and decrease of exposure, added noise in up to 6% of pixels, and also up to 25% grayscale. Another benefit is that it provided more data to train the object detection model on, doubling the amount I had collected.

3.2 YOLO v7

Beginning the analysis, we focus on the first portion which is object detection using YOLO v7 framework (Wang et al., 2022). Using the images that were collected we are now able to use them in training the object detection model. By using images that were gathered independently from the video data, we strive to create a more universal model that is adaptable to a wider input of video data. Training the model

that is essentially the foundation of this analysis would only provide false accuracies and create many problems with overfitting throughout the whole study. Here by using YOLO v7 we look to isolate the kicker from the other players surrounding on the field. By isolating the kicker's body we can then further analyze the key points extracted from the pose estimation model, and eventually predict the direction of the penalty based on the sequences of movement they make in their run up to shooting the ball.

We have already examined YOLO, but much has changed since the version we had observed earlier was released. We can now go deeper to look at the advantages of YOLO v7 and its application here (Wang et al., 2022). While this version still follows the fundamental concepts of previous versions, there is much that has been improved. Specifically one of YOLO's struggles in the past has been identifying smaller objects, but several features have been added to combat these issues (Wang et al., 2022). The first major change unique to YOLO v7 is changes to the loss function from a standard cross-entropy loss function to now a loss function called focal loss. Focal loss makes the distribution statistics of a bounding box highly correlated to its real localization quality (Li et al., 2020). Essentially what it does is that it places more importance on the classes that are more challenging to identify by adjusting a weight associated with it that is then used to calculate the loss. The weights are calculated based on the confidence score that the model determines where if the object is given a higher confidence score, the loss is less and vice versa (Li et al., 2020). In a sense it doesn't worry so much about the objects that are easy to detect and detectable with high confidence, but rather on the objects that pose more of a challenge.

In addition YOLO v7 has a higher resolution that allows for further improvements in the detection of smaller objects. It's also quicker in terms of processing frames per second, requires less computation, and has a higher average precision than the other previous versions. Within YOLO v7 there are several models: YOLOv7-X, YOLOv7-w6, YOLOv7-e6, YOLOv7-d6, and YOLOv7-tiny. Each one of these models has different attributes like size and number of layers that make them more suitable for different applications (Boesch, 2023). For example YOLOv7-tiny is a version that is "lighter" that allows for the ability to run on mobile computing devices or distributed edge servers (Boesch, 2023). In comparison YOLO v7 (Wang et al., 2022) is the

most basic model that is intended for ordinary GPU computing, and this is the foundation of the model that I created here in this step.

In actuality all we needed to extract was the bounding box of the kicker. I trained a model using the weights of the YOLOv7 base model and trained it on the annotated image data (n=869) to create the kicker detector. This data was split for cross validation dedicating 61% (n=294) to the training set, 21% (n=101) to the validation set, and 18% (n=88) to the test set. After altering the python script for detections I created a mask and exported the contents of the image that remained inside of the kicker's bounding box.

This object detection task is really a key component of this analysis. Due to the nature of the output, it is reliant that the model is effective in identifying the kicker because the frames and clips that get passed on are those where 1 kicker class was identified. While running the script, I had noticed that there were some instances where either 0 detections were made and the frame was passed through without a mask or there was more than one detection made and passed through more than one bounding box. Since we know for a fact there should only be one kicker, and either of these outcomes would alter the results in the pose estimation step I decided that it was better if these instances were omitted rather than being passed through. This potentially has the problem of removing too many frames from clips that were difficult to detect, but the cost for having more accurate data to work with later on.

3.3 Yolo Pose

Now with our kickers detected and each frame crop down to only isolate the kicker's bounding box, we move on to our next phase of the detection which is the key point extraction and pose estimation. As described previously, YOLO v7 (Wang et al., 2022) has several sub models for various applications. There are models suitable for tasks like instance segmentation, but the one that interests us is the model for pose estimation. We can refer to this model as YOLO Pose (Maji et al., 2022). This approach provides heatmap-free joint detection as well as 2D multi-person pose estimation. The foundations of this concept are based in the YOLO object detection

framework that we have covered in the previous sections. Specifically it exhibits the one stage approach seen in YOLO which differs from other pose estimation approaches which are typically heatmap based and two-staged. Also this approach retains the benefits of both top-down and bottom-up approaches by retaining the advantages. It doesn't require the post-processing that is required in bottom-up methods, and it only requires a single pass rather than the multiple passes that are seen in top-down methods. When compared against other common pose detection methods such as OpenPose (Maji et al., 2022) and MediaPipe (Kukil et al., 2023), YOLO Pose has a higher average precision which in this application is very important. In this stage, the priority is accumulating data that is accurate to the kicker's movements and body mechanics. Unfortunately this comes at the price of speed, where other options produce an inference much quicker.

In each image or in our case, frame of a video, we get an output of 51 items. These items correspond with the coordinate and the confidence (x, y, confidence) of each of the 17 key points detected in the frame. This third item, confidence, is determined based on the visibility flag of that key point (Maji et al., 2022). We have to be aware that not every frame will have a clear visualization of each of the 17 zones where keypoints are to be detected. This is where this feature is useful. The confidence score ranges from 1 being the absolute key point with full confidence, and 0 being completely invisible with no confidence. While this is an important feature of this method, this confidence score was not included in the data that was passed forward to train the LSTM. This was done in order to train the model solely on the body key points and try to keep it as simple and functional as possible. Excluding the confidence scores, I narrowed it down to 34 items that were output per frame.

```
[23]  vid_list2[0]

    '\'vidname\': \'MRightThe_Biggest_German_Youtuber_Penalty_Shootout__2021_____FKFRIDAY_Ep._04_013.mp4\', \'coords\': [[array([
    806,      141.88,      0.6958,       805.5,      132.12,      0.7959,         805,      134.25,     0.03186,       820.5,
    126.88,     0.97754,         846,       126.5,     0.19482,       839.5,         168,     0.94971,         884,       160.12,
    0.88525,         869,      233.12,     0.92725,             914.5,      224.25,     0.67725,       864.5,         297,     0.8
    6816,         869,      267.75,     0.58838,       907.5,      291.25,     0.96631,       939.5,         285,     0.94971,
    874.5,      393.75,     0.90088,       924.5,       390.5,     0.85498,         947,         467,     0.65625,              95
    3.5,         474,     0.60303])], [array([        789,       151.5,     0.60449,         789,      141.75,     0.69482,
    787.5,      143.12,    0.031738,         806,      136.88,      0.9707,         827,      135.62,     0.13...'
```

Figure 3.4: One Video Entry in Cumulative Dictionary Before Reduction (coordinates = 51 items)

video	label	F1	F2	F3	F4	F5	F6	F7	...	F11	F12	F13
'Right___RIGORI_CHALLENGE_Di_Serie_A__chi_vinc...	R	[315.07, 194.28, 313.06, 188.14, 315.24, 187.9...	[321.14, 191.03, 318.83, 185.35, 321.57, 185.1...	[328.05, 189.04, 326.09, 183.7, 328.99, 183.36...	[336.19, 189.18, 334.37, 183.21, 336.82, 183.3...	[338.89, 189.99, 337.61, 184.13, 339.41, 184.3...	[347.38, 193.9, 345.48, 188.98, 347.55, 188.7,...	[355.12, 197.38, 352.44, 191.48, 355.95, 191.8...	...	[381.74, 186.92, 378.85, 181.69, 381.94, 181.7...	[369.21, 183.32, 366.78, 178.17, 370.6, 178.32...	[370.56, 182.05, 367.99, 177.08, 370.89, 176.9...
'Right___RIGORI_CHALLENGE_Di_Serie_A__chi_vinc...	R	[204.29, 253.78, 204.82, 250.7, 202.12, 250.3,...	[209.26, 253.2, 210.57, 250.15, 206.6, 249.62,...	[214.76, 251.84, 215.42, 248.99, 212.1, 248.48...	[222.34, 253.15, 222.84, 250.26, 220.61, 249.7...	[229.09, 255.08, 229.63, 252.15, 227.62, 251.4...	[235.75, 256.21, 236.13, 253.04, 233.8, 252.72...	[242.21, 257.22, 243.0, 254.08, 240.64, 253.83...	...	[271.53, 252.0, 271.97, 249.21, 268.58, 249.0,...	[277.86, 249.94, 278.01, 247.02, 275.3, 246.75...	[284.65, 248.56, 284.47, 245.78, 282.04, 245.3...
'Right_50_Callout_Penalty_Challenge_002.mp4'	R	[330.29, 200.26, 330.15, 194.82, 328.96, 194.4...	[335.63, 201.06, 335.39, 195.83, 334.35, 195.5...	[341.34, 201.63, 340.94, 196.56, 339.96, 196.1...	[345.97, 202.39, 345.79, 197.14, 344.83, 196.9...	[350.33, 202.93, 349.53, 197.79, 349.11, 197.6...	[354.23, 206.39, 354.01, 201.23, 353.07, 201.0...	[359.23, 210.37, 358.9, 205.01, 358.49, 204.99...	...	[376.34, 209.92, 374.85, 205.4, 374.96, 205.1,...	[379.31, 207.08, 377.56, 202.53, 377.98, 202.3...	[382.54, 206.3, 380.98, 201.73, 380.93, 201.67...

Figure 3.5: Dataframe Consisting of the 34 Keypoints per Frame Along With Label and Video Title

Throughout the analysis, these frames and their respective coordinates remained in the original sequential order that was seen in the videos. This characteristic especially is important when pre-processing for the LSTM since we are trying to observe the movements in their progression towards the ball. If this were to become shuffled out of order, it would no longer preserve the sequential dependence needed to predict based on the previous movements. Also noted that within the frame, the coordinates remained in the same order as intended by YOLO Pose so as to maintain the ability to identify joints and limbs of the skeleton. This list was then incorporated into a dictionary that allowed for the inclusion of the video name, as well as the label of the final direction of the ball so then the coordinates could be tracked at a later point. Each one of the dictionaries corresponded with a portion of a video that was collected in the initial data collection phase. The model was deployed using the weights from the pre-trained model that was trained on the COCO dataset which has data on 17 landmark topologies of the human body. This is where we reach the conclusion of our data collection process. At this point we are prepared to proceed to the LSTM stage where we can train a model to process the numerical key point data from the videos and predict the direction that the kicker will shoot the ball.

3.4 LSTM

The final portion of the analysis is the LSTM portion. Here we use the data that we've collected before from the prior two models to train this final model to predict the direction that the ball will go after being kicked in a penalty kick. Essentially what we're looking for here is patterns within the sequences of the kicks. We're searching to identify information from prior body movements that would help indicate the direction of the balls ending place in the goal. We need to retain the information that is pertinent and disregard any nuances in movement that don't help us determine the final location. After researching more on analyzing and processing pose data, I decided some form of normalization needed to be implemented on the key points extracted from the pose estimation step (Ren et al., 2016).

```
[105]  main_nest_arr[0]

       [-0.19966111,  0.35263251, -0.2051147 ,  0.3475265 , -0.20505214,
         0.35298587, -0.20504171,  0.30114841, -0.20415537,  0.34591873,
        -0.17251825,  0.26683746, -0.1722367 ,  0.36418728, -0.13188217,
         0.21897527, -0.13127737,  0.38659011, -0.10821168,  0.22053004,
        -0.09055787,  0.41022968, -0.08143379,  0.27712014, -0.08156934,
         0.34326855, -0.01219499,  0.24551237, -0.02677268,  0.36553004,
         0.06048488,  0.21489399,  0.04445777,  0.35616608],
```

Figure 3.6: Coordinate Values for the First Frame in the First Video of the Training Set After Normalization

After other layers like dropout and normalization layers were added in with the attempt to find the best model. The network construction was very much a process of trial and error. I began with the most simple version of the model with one LSTM layer and one dense layer using a softmax activation function.

Chapter 4: Results

4.1 Object Detection Results

Some difficulties encountered in this step included issues with recognition and detection while there were non-kicker players around, for example, a penalty kick that occurs within regulation time rather than a penalty kick after a game. Another issue faced in detection was not identifying the referee as opposed to the kicker or also the goalkeeper as opposed to the kicker. Changes were made to the underlying detection script within the model to only retain frames essentially that had one detection, meaning that if a kicker and another player were both detected as kickers that way rather skip that frame and only retain the frames that have one kicker detected so we know that there's only one kicker in a penalty kick. Unfortunately, this technique can cut down our data by a decent amount but I found that this resulted in greater accuracy and the more reliable prediction ultimately. An example of this is shown in the sets of images below. The number of frames is greatly reduced from what it actually is in the project just for simplicity in the visual representation.

Figure 4.1: Frames of clip prior to being processed by YOLO v7

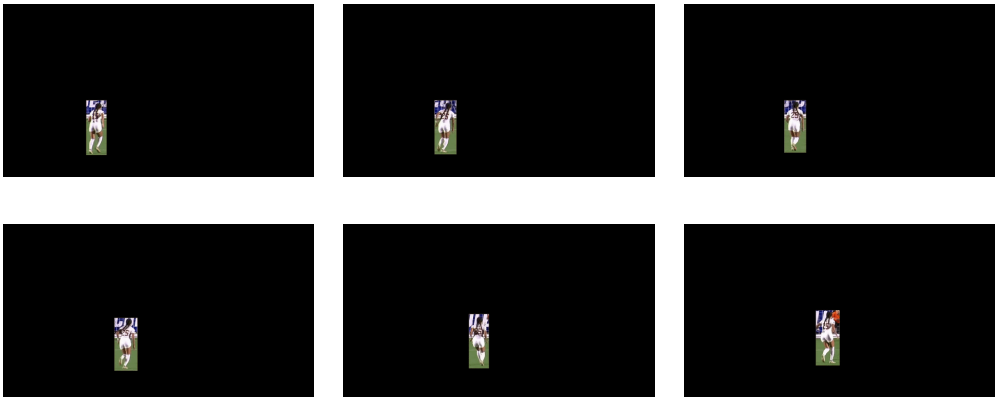


Figure 4.2: Frames of same clip after processing by YOLO v7

4.2 Pose Estimation Results

When determining the approach to use in this study, multiple pose estimation libraries were tested in an attempt to find one that best fits this application. Some of the other pose estimation tools I tested were OpenPose and MediaPipe. Ultimately Yolo Pose was decided upon because of its simplicity as well as accuracy. In addition this method is structured for multi-person pose estimations. I had encountered issues with single-person pose estimation libraries where the goalkeeper or another player in the background of the bounding box would produce a distorted skeleton combining both bodies when it was intended to identify only the kicker. This led to completely inaccurate results at a stage where accuracy was very important.

With this model being pre-trained on the COCO dataset, there wasn't really an evaluation to be had since I wasn't creating it and testing it myself. With that said I monitored all the videos that were produced alongside with the coordinates and looked for any visible inconsistencies or inaccuracies. If any were found, I dealt with this issue by going and improving the annotations with the intention of the bounding boxes in the object detection training set. The hope was with a better defined bounding box, that other players wouldn't be registered by the pose estimation model and skeleton key points of any other people wouldn't be collected along with the kicker’s. After going back and adjusting many of the annotations to be more carefully selected as well as improving the dataset that the object detection model was trained

on to account for a wider variety of camera angles I found that this helped cut down the amount of unintended detections of players other than the kicker.

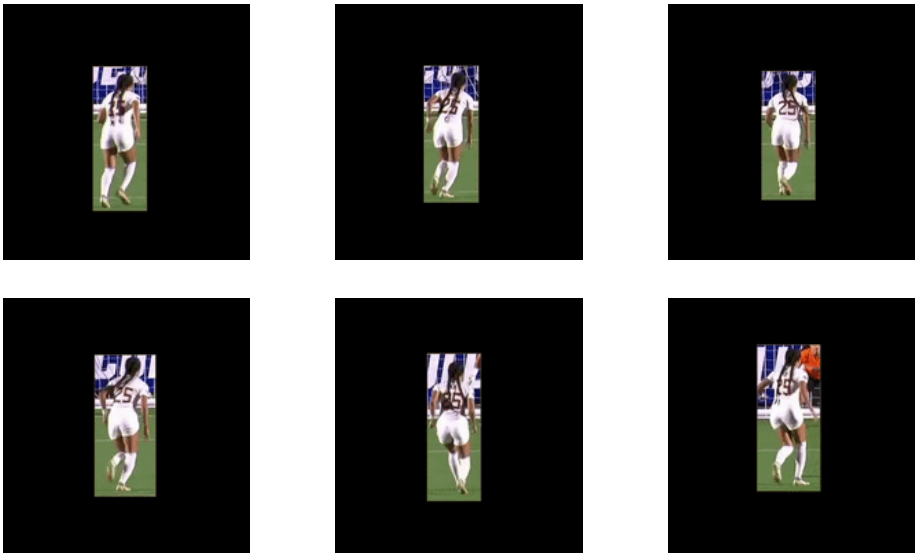


Figure 4.3: Frames of clip after processing by YOLO v7

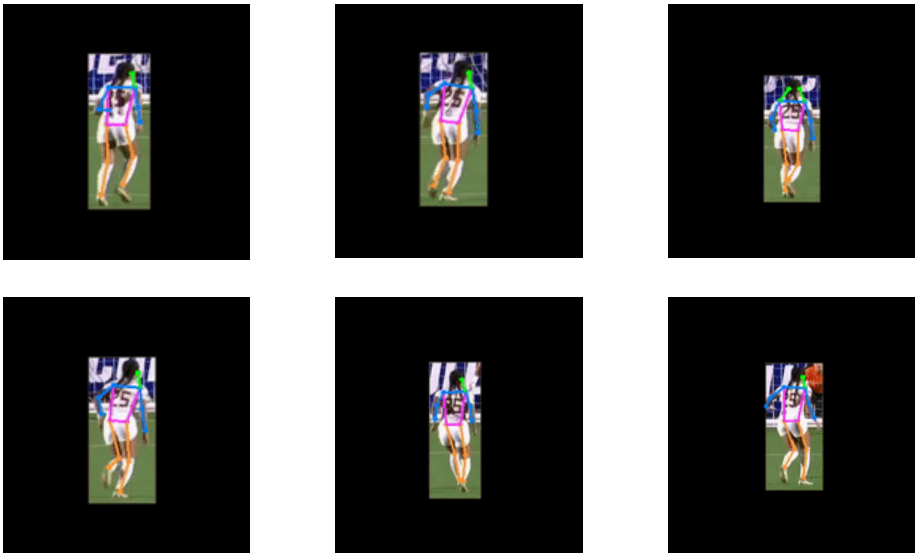


Figure 4.4: Frames of same clip after processing by YOLO Pose

This was performed by taking the size of the videos, which had already been altered in the object detection modeling at a uniform size of 567 x 960 pixels. Then by using

the X and Y coordinates for each of the 17 points output by the pose estimation model I normalized the points to have values between -1 and 1.

This technique was seen in other literature that I encountered performing similar concepts.

4.3 LSTM Results

Working with the keypoint data unadjusted proved to be a difficult task and it was nearly impossible to achieve an accuracy above a random guessing level. This random guessing was created by a random number generator between 1 and 3 to replicate if a goalkeeper were to randomly decide. To evaluate the efficiency of the models produced in this section metrics like accuracy, precision, and recall were used. These metrics allowed for comparison between models as well as understanding their performance compared to a random selection of direction. In addition when referencing hyperparameters, this is used as a general statement to indicate aspects like batch size, dropout rate, activation functions, and epochs. When generally referring to this term, it is to indicate that a combination of these features were attempted in an effort to optimize the model.

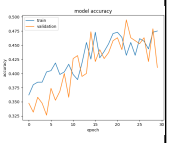
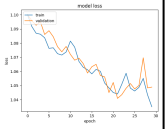
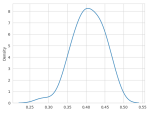
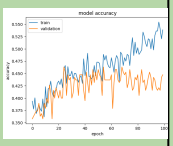
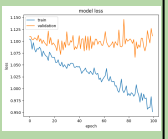
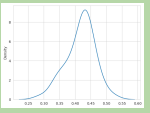
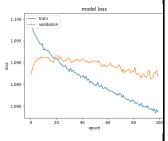
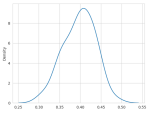
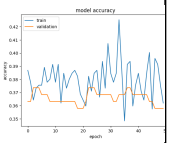
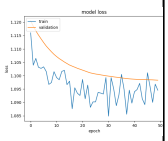
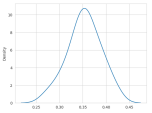
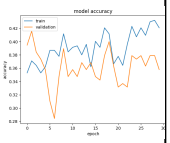
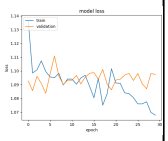
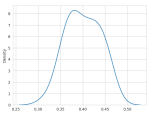
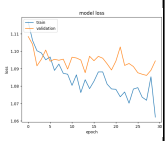
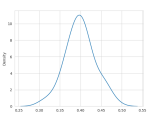
Initially in the preprocessing prior to the LSTM I retained the last 10 frames, due to the fact that any videos less than 10 frames long at this point in the study were no longer considered. So by setting the threshold so low I had the intention of retaining more data since there were more videos with at least 10 frames. By retaining this amount of frames we would have had an overall size of n=1085. However by retaining more frames we would be able to use earlier movements in the run up to the shot as part of the prediction. When observing the effects of retaining 20 frames we see that the overall size is n=791 which is less than the overall size if 10 frames are retained, but not a drastic difference and we have a longer sequence to train the LSTM with. With the decision to retain 20 frames, we were left with n=261 videos that were labeled “Left”, n=270 videos labeled “Center”, and n=260 videos labeled “Right”. Now with our uniformly shaped data we can use Keras Tuner to speed up the testing of variations of the hyperparameters such as batch size, dropout values, and learning

rates. The hyperparameters that I specified to be tested in this portion were batch size: {16,32,48,64}, dropout rate: {0,0.1,0.2,0.3,0.4,0.5}, dense layer activation function: {ReLU, sigmoid, softmax}, learning rate: {0.01,0.001,0.0001}. This process was then run on different optimizers {Adam, Root Mean Square Propagation (RMS Prop), Stochastic Gradient Descent (SGD)} for 100 epochs. After selecting the optimal hyperparameters I used the model to form predictions based on the test data to then evaluate. This prediction/test set was then compared with a randomly generated dataset of the same size (n=159) to simulate a random direction being guessed by a goalkeeper. To compare the models with one another and with the randomly generated output I used the accuracy, which was calculated as the equation shown below. This metric is used to demonstrate the proportion of instances that were correctly predicted.

$$Accuracy\,(Acc) = \frac{TP}{TP + TN + FP + FN}$$

The models were to be evaluated by using the outputs from the data, the Y test set, along with the predicted outputs from the trained models. From this point the hyperparameter tuning began to find the best combination to predict our data.

Table 4.1: Filtered Optimal Results

Model Accuracy	Model Loss	Batch Size	Dropout Value	Epochs	Optimizer	Accuracy (1 run)	Average Accuracy (50 runs)	Confusion Matrix (1 run)
		48	NA	30	Adam lr = 0.001	0.4025	 0.406	.425 .375 .291 .35 .4 .316 .225 .225 .392
		48	.4	30	Adam lr = 0.001	0.4088	 0.416	.463 .318 .288 .317 .363 .288 .219 .318 .423
		48	NA	100	SGD lr = 0.01	0.390	 0.3977	.446 .5 .173 .311 .4 .493 .243 .1 .333
		48	.2	50	SGD lr = 0.001	0.396	 0.353	.5 .290 0 .25 .355 0 .25 .355 1
		16	.4	30	RMS lr = 0.01	0.421	 0.401	.436 .25 .234 .372 .25 .35 .192 .5 .416
		32	.4	30	RMS lr = 0.001	0.428	 0.399	.432 .333 .244 .365 .666 .341 .203 0 .415

This is a framework that allows for combinations of hyperparameters to be quickly tested and provides a list of the best performing combinations as an output. To narrow down all the combinations for hyperparameters, I ran the tuner so that it would produce an accuracy based off of 1 run for each combination tested. I gathered the top two performing combinations for each optimizer and monitored the training curves to watch out for models that were being overfit and compiled a list of the best hyperparameter combinations. After I had filtered the selection of hyperparameters, I ran the models 50 times for each hyperparameter combination, each time re-establishing the training, validation, and test set as well as recreating the model with the intention of keeping each run independent from each other. The average accuracy displayed in table X above shows the mean accuracy of the 50 runs as well as the distribution of accuracy scores. While on the initial runs where the best hyperparameters were picked the RMSprop optimizer showed a higher accuracy, but in the end it was the models that used the Adam optimizer that had the highest average accuracies overall. More specifically, the best model that was tested is the model with a batch size of 48, a dropout value of .4, ran for 30 epochs, and using the Adam optimizer at the default learning rate of 0.001. This model produced an accuracy of 41.6% when compared to the average accuracy of the randomly generated outputs (33.8%) which shows a 7.8% improvement in prediction accuracy.

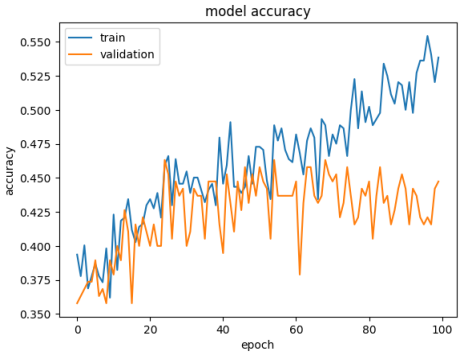


Figure 4.5: Accuracy Curve for Selected Model

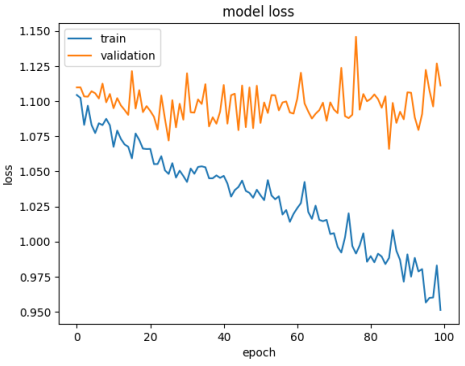


Figure 4.6: Loss Curve for Selected Model

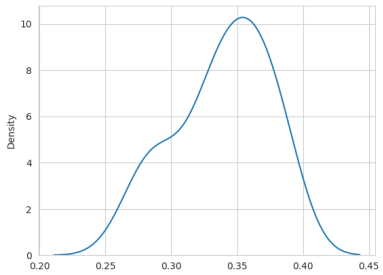


Figure 4.7: Random Outputs Accuracy Distribution

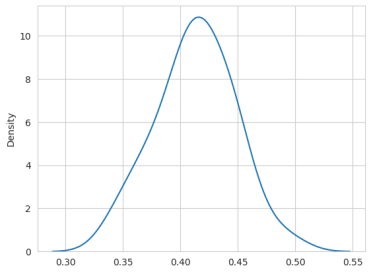


Figure 4.8: Selected Model Accuracy Distribution

.338	.32	.22
.338	.38	.38
.322	.3	.4

Figure 4.9: Random Outputs Confusior Matrix

.447	.307	.213
.297	.384	.362
.255	.308	.426

Figure 4.10: Selected Model Confusio Matrix

Chapter 5: Applications

In theory, this sort of analysis has the potential to benefit several areas of soccer. As we have seen in recent years with the VAR technology, the game is ever-changing and evolving. By creating a model that analyzes the kicker's body movements, we could train goalkeepers to identify signs earlier on in the run-up to the ball to allow for more of a reaction time. This could be applied to more than penalty kicks, it could be incorporated in scouting reports or used for injury detection based on repetitive or incorrect movements. Another potential application could be for sports betting. With soccer being the most popular sport in the world, it is no surprise that it has the largest betting market as well. Analyses suggest that the market is growing and is expected to reach 297,638.20 million in USD by 2030. With a given broadcast delay and a computer that processes quickly enough, using a model to predict the outcome of penalties for betting could be possible if real time bets are allowed. In addition, another application could be graphics or interactive polls where fans try to predict the side or specific spot where the kicker shoots the ball. Perhaps there could be some sort of competition where fans are in competition with the model in making a prediction. An application like this could be a fun game and a great way to engage fans in the sport.

Chapter 6: Conclusions

Final Statement

In this analysis we see more than ever the challenge in predicting penalties at a high accuracy. The goal initially was to outperform a random selection, which we were able to achieve. Although 7.8% improvement may not sound groundbreaking, if we were to apply this to practical use and then observe the conversion rates of penalties it would become much more apparent how impactful 7.8% improvement would be. This analysis and process of model creation is not just limited to penalty kicks. This has the potential for so many other applications within soccer and throughout sports. Incorporating computer vision in both data collection as well as player skill optimization and recruiting is the future of sports. Combining the worldwide popularity with the huge financial market that surrounds soccer this is certainly a huge field for potential.

Drawbacks and Limitations

The biggest limitation by far faced in this study was the data collection. While Youtube and the internet have a vast amount of soccer video data available, the resources required to collect, segment, and crop these videos was extremely time consuming. Some areas where this could potentially be expanded in the future is to automate that laborious process eventually to achieve the point where the scraping can be done with minimal interaction.

In addition during the labeling, the decision was made by a human with goalkeeper experience and is considered a subjective opinion. Having a lifetime of experience playing soccer as goalkeeper, I consider myself to be a good judge of reading the ball and understanding the type of movements a goalkeeper is making. However there is

potential to have issues with inconsistent labeling, especially when it comes to determining the boundaries for the center segment of the goal. I based my decisions off of where I was able to see the ball cross the line, the push-off visible in the goalkeeper's legs (which would determine how far they needed to travel to save the ball), as well as where the ball came in contact with the back of the net. This is not something I can easily verify, but each video was viewed several times and even evaluated in slow motion to ensure that I can label it to the best of my ability.

Another factor that is important to keep in mind is that in this analysis we rely on the belief that each three sections of the goal are equally as likely to be shot at. In reality, this is not true and there are certain sections of the goal that are more aimed at than others.

Future Directions

Some future areas to further explore are more detailed now this is in specific body movements. This will tie more into the goal of being able to isolate body parts to then train goalkeepers on specifics to watch out for and anticipate. Using the normalized outputs of the pose estimation coordinates, we can group parts of the body and isolate body parts like the leg or the hips to perform this analysis and better understand which body parts are optimal for prediction. Another interesting area I would like to explore is to incorporate this process with the pre-existing penalty analyses that have been performed. For example there are studies that focus on the stochastic analysis that examine the probability of kickers choosing a certain direction or not. Another interesting analysis I have seen was regarding the order in which the penalty kicks were taken in the penalty shootout, this could be another interesting combination for future analyses.

As mentioned before this analysis can be made to predict more specific directions in the goal. While this study only examined splitting the goal into 3 parts (left, center, right) we could segment the goal into 6 parts (left bottom, left top, center bottom, etc). This would give us a more detailed prediction which ultimately would help the

goalkeeper be more effective in saving the ball. Should we want to move our focus from the direction the ball will go, we could even predict a potential miss. Ideally you could use similar data to predict whether on the body movements in the run up to the ball whether they are more likely to miss the goal (leaning back too much or some other metric to determine this).

Another method for future data collection could be scraping other social media websites using keywords similar to the process I used on YouTube. Websites like Instagram, Tiktok, and Twitter have a vast reservoir of short-form content and even more specifically targeting sports and soccer focused pages could be a great source for this type of data. This is certainly a step that can be continued in the future to improve the accuracy of the models being created in this study. However at this time, YouTube provided the clips in compilations so they could be gathered in large groups rather than individually. In an effort to eliminate some of the time needed to collect all this data, this route was taken.

Bibliography

Abramovich, F., & Pensky, M. (2019, July 17). Classification with many classes: Challenges and pluses . https://arxiv.org/pdf/1506.01567.pdf

Boesch, G. (2023, February 24). *Yolov7: The most powerful object detection algorithm (2023 guide)*. viso.ai. https://viso.ai/deep-learning/yolov7-guide/#:~:text=The%20differences%20between%20the%20basic%20YOLOv7%20versions,-The%20different%20basic&text=YOLOv7%2Dtiny%20is%20a%20basic,distributed%20edge%20servers%20and%20devices.

CIES Football Observatory. (n.d.). *Weekly Post 418 - football observatory*. Football Observatory. https://football-observatory.com/IMG/sites/b5wp/2022/wp418/en/

Divya, R., & Peter, J. (2021). Performance comparison of various object detection models. https://www.researchgate.net/figure/Performance-comparison-of-various-object-detection-models_fig5_349960212

Federation Internationale de Football Association. (2015). FIFA Rules. https://digitalhub.fifa.com/m/3f3e15cc1ab8977b/original/datdz0pms85gbnqy4j3k-pdf.pdf

Global sports betting market – industry trends and forecast to 2030. Sports Betting Market

Scope & Analysis Report to 2030. (n.d.).

https://www.databridgemarketresearch.com/reports/global-sports-betting-market

Hutchins, D. (2022, December 6). *How to read a soccer penalty shot if you're a goalie: 11*

steps. wikiHow.

https://www.wikihow.com/Read-a-Soccer-Penalty-Shot-if-You%27re-a-Goalie

Jain, A., Bansal, R., Kumar, A., & Singh, K. D. (2015). *A comparative study of visual and*

auditory reaction times on the basis of gender and physical activity levels of medical

first year students. International journal of applied & basic medical research.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4456887/

John, A. (2022, June 15). *The physics and Mind games of a world cup penalty kick*. Popular

Mechanics.

https://www.popularmechanics.com/adventure/sports/a5864/world-cup-penalty-kick-p

hysics/

Kukil, Vikas Gupta, & Gupta, V. (2023, May 9). *Yolov7 pose vs MediaPipe in human pose*

estimation. LearnOpenCV.

https://learnopencv.com/yolov7-pose-vs-mediapipe-in-human-pose-estimation/#YOLOv7-vs-MediaPipe-Pose-Features

Kundu, R. (2023, January 17). *Yolo algorithm for object detection explained [+examples]*.

YOLO Algorithm for Object Detection Explained [+Examples].

https://www.v7labs.com/blog/yolo-object-detection

Li, X., Wang, W., Hu, X., Li, J., Tang, J., & Yang, J. (2020, November 25). *Generalized focal*

loss V2: Learning reliable localization quality estimation for dense object detection.

arXiv.org. https://arxiv.org/abs/2011.12885

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016,

December 29). *SSD: Single shot multibox detector*. arXiv.org.

https://arxiv.org/abs/1512.02325

Maji, D., Nagori, S., Mathew, M., & Poddar, D. (2022a, April 14). *Yolo-Pose: Enhancing yolo*

for multi person pose estimation using object keypoint similarity loss. arXiv.org.

https://arxiv.org/abs/2204.06806

Nag, U. (2023, May 27). FIFA World Cup Final Records, stats and faqs - olympics.com.

https://olympics.com/en/news/fifa-world-cup-final-records-stats-faqs

Olah, C. (2015, August 27). *Understanding LSTM networks*. Understanding LSTM Networks

-- colah's blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Park, C., Lee, H. S., Kim, W. J., Bae, H. B., Lee, J., & Lee, S. (2021, November 17). *An*

efficient approach using knowledge distillation methods to stabilize performance in a

lightweight top-down posture Estimation Network. MDPI.

https://www.mdpi.com/1424-8220/21/22/7640

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016, May 9). *You only look once:*

Unified, real-time object detection. arXiv.org. https://arxiv.org/abs/1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016, January 6). *Faster R-CNN: Towards real-time*

object detection with region proposal networks. arXiv.org.

https://arxiv.org/abs/1506.01497

Rizzoli, A. (2021, June 10). *Object detection: Models, Architectures & Tutorial [2023]*.

Object Detection: Models, Architectures & Tutorial.

https://www.v7labs.com/blog/object-detection-guide

Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V.

(2021a, May 10). *Comparative analysis of Deep Learning Image Detection*

Algorithms - Journal of Big Data. SpringerOpen.

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00434-w#Sec

26

Sun, K., Lan, C., Xing, J., Zeng, W., Liu, D., & Wang, J. (2017). CVF Open Access.

https://openaccess.thecvf.com/content_ICCV_2017/papers/Sun_Human_Pose_Estimat

ion_ICCV_2017_paper.pdf

Vizcaino, S. (2021, May 5). *The experience in the performance of the goalkeepers in penalty*

kick. CEFARQ.

http://cefarq.com.ar/the-experience-in-the-performance-of-the-goalkeepers-in-penalty-

kick/

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022, July 6). *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. arXiv.org.

https://arxiv.org/abs/2207.02696

World Cup 2022: Penalties var. BettingOdds.com. (2023, May 8).

https://www.bettingodds.com/news/world-cup-2022-penalties