Comparative Study on Model Skill of ERT and LSTM in Classifying Proper or Improper Execution of Free Throw, Jump Shot, and Layup Basketball Maneuvers

John Paul Q. Tomas *, Kevin I. Lucero, Christian Jose P. Ajero, and Renz Justin V. Thomas

School of Information Technology, Mapúa University, Makati, Philippines; Email: kilucero@mymapua.edu.ph (K.I.L.), cjpajero@mymapua.edu.ph (C.J.P.A.), rjovivo@mymapua.edu.ph (R.J.V.T.)
*Correspondence: jpqtomas@mapua.edu.ph (J.P.Q.T.)

***Abstract*—Traditional basketball sports training is highly subjective as coaches manually analyze and train players based on their own curriculum and knowledge. With today's advancements in computer vision and machine learning, we can now use these technologies to correctly recognize and classify human actions. Researchers have utilized motion tracking systems to analyze human motion in various fields such as motion capture, sign language translation, gesture controls, virtual reality, and even for medical treatments. These systems commonly use RGB-D cameras to capture data due to the features the cameras offer especially their ability to capture depth images. Coupled with machine learning, such a system that can recognize and classify human actions has been more feasible than ever. This study will use a Microsoft Kinect V2 to capture the footage of players performing three maneuvers: the jump shot, free throw, and lay-up. The data would be collected and pre-processed using C#, Kinect SDK, and Kinect PV2 libraries. The model will be able to classify if each maneuver was performed properly or not by tracking the whole body and its parts along with its joints. The proponents will then use Scikit-Learn as the platform to train an Electrical Resistivity Tomography (ERT) model and a Long Short-Term Memory (LSTM) model and find out which model will be more robust for this kind of application.**

***Keywords*—computer vision, motion tracking, posture recognition**

I. Introduction

Many research efforts have focused on the problem of pose and gesture recognition. Computer vision has been used by researchers to observe and recognize human posture and movements through Human Motion Tracking and Posture Recognition. Pose and gesture recognition aims to recognize meaningful expressions performed by humans in their everyday lives [1]. These expressions are seen through the motion of the hands, arms, head, facial expressions, or the whole body itself. These recognition systems aim to facilitate interfacing between computers and humans [2]. The general idea of image acquisition gesture recognition in sports is to first use a camera to capture footage from athletes, then extract the key features, filtering out noise and isolating the body from the background to recognize the individual body parts and their posture. Most of these approaches model human skeletal joints and their motions. Hu *et al.* [3] were able to recognize pass, dribble, lay-up, and shooting motions. Rahma *et al.* [4] also created a model capable of identifying the moments' key poses in a free throw that was performed and extracting Hu moments from the frames of the key poses. All this information helps players, trainers, and coaches work together in improving and refining the technique of the player.

II. Related Research

This section shall detail the relevant and related previous works that this study will draw upon and innovate from. Specifically, this section will discuss basketball maneuvers, RGB-D tracking and recognition, and the use of algorithmic classifiers used by previous works and their findings.

A. Basketball Maneuver

Modern sports training is about the collection and analysis of basketball player's posture data to improve the science of a coach's training plan and to improve the effect on it on the athlete. Traditional training methods were very subjective as it is based on the training theory of the coach, experience, and his team's skill level to develop a training plan. The core of modern sports training is precision, efficiency, and objective. The effectiveness of the training is greatly improved if the coach has a means of accurately monitoring the athlete's movement posture. According to the study of Ji [5] to effectively recognize the posture from the physical state of the basketball player, they have divided the postures into two states: static and sports. The sports state is the state of the athlete when he is performing basketball actions, in which the athlete's limbs are moving

Manuscript received August 14, 2022; revised September 21, 2022; accepted October 21, 2022; published June 25, 2023.

while the static state refers to the strict neutral state of the athlete's limbs and not in motion.

The filtering, collection, and analysis of a basketball player's posture data accurately and recognizing the sports posture will significantly improve the coach's training plan and improve its effectiveness [6, 7]. Basketball gesture and motion recognition is a hot topic amongst researchers in the field of computer vision. Currently, there are two types of motion recognition used in this field. The first is using motion sensors which are worn by the participants to measure the inertia of the body and are sent to the processing machine. The problem with this method is its prohibitive cost and the large amount of equipment involved [8]. With image capture, however, it is much cheaper and more accessible. Using a camera to collect the athlete's image or video is one of the main principles of image acquisition. Afterward, the motion features in the image and video are extracted and a classifier is used to recognize the athlete's athletic gesture [9].

The step for shooting a basketball is nearly universal for all players there is still a difference in the manner of movement among different individuals. From the study of Okazaki *et al.* [10], they grouped the jump shot into 5 phases: first is preparation, second is ball elevation, third is stability, fourth is release and inertia for last. The preparation phase is when the player collects the ball close to them and engages the muscles in their body in preparation to shoot. During the ball elevation phase, they describe the step where the player has moved the ball over their arms and begin to lift the ball, the stability phase is where the player begins to use their wrist to control the movement of the ball during the shot attempt. The release phase is where the player uses a combination of elbow extension and wrist flexion to impart momentum on the ball as it launches, and the inertia phase is the phase after the ball has begun traveling in the air and the player recovers from the action of shooting. There is, however, no universal optimal shooting form discovered yet, only kinematic models [11]. The recognition of various sports postures is the key to basketball posture recognition.

B. Human Skeleton and Action Recognition

The accurate estimation of human body orientation will greatly enhance human pose estimation, body tracking, pose estimation, and action recognition and feature extraction [12–14]. RGB-D cameras can capture RGB images along with their per-pixel depth information in real-time. Compared to 2-D information, the geometry information brought by depth makes it possible to have accurate tracking regardless of the illumination change, and partial occlusion in pose and orientation estimation. The geometrical information is also acquired in addition to 3D information. While these RGB-D cameras are cheaper, more convenient, and have lower computational complexity, they are of course with challenges and issues. They are susceptible to noise, ambient occlusion, and inaccurate tracking of skeletal joints.

To address the issues regarding tracking accuracy of single RGB-D Cameras (i.e., Kinect v1, v2), an experiment by Motta *et al.* [15] have used trained classifiers and to retrieve the depth data that would separate the actor's silhouette from the background using existing methods and then developed their method of creating a 3D skeleton instead of 2D to produce more accurate results. The major contributing factor is the combining of the image texture data along with the depth data to detect and classify body parts per pixel, estimated joint positions, and 3D virtual skeleton.

C. Recognition Using Machine Learning

There are various available techniques to represent human activities in RGB-D footage. Akam *et al.* [16] followed the Bag of Features approach in feature extraction. Their method was to extract the local motion and appearance features which are then recognized by detecting the important interest points from the special domain by extracting visually distinctive points using the Speed-Up Robust Features (SURF) Algorithm. These SURF points are then filtered by Motion History Image and Optical Flows to only extract significant motion points from the sequences.

To represent the shape, motion information, and appearance the Bag of Features is generated by the combined feature 20 vector values from the RGB-D video frames filtered by HOG. Hu moments which are the similarities between two patterns were determined by applying FFT on the binary representation of the idle position of the input data and other positions are stored as a 2D matrix. A correlation filter is then applied to each frame to generate a 2D correlation plan. Position detections are determined by the vector with max peak.

By combining the hu-moment features, it generated feature vectors with the HOG features to represent the action information from each RGBD video. The feature vectors are combined and encoded into a single code by using the bag of features algorithm by extracting the local image features and Hu-moments. K-Nearest Neighbors (KNN) algorithms are used for the classification of the different actions from videos. Their proposed method scored a higher accuracy score than other existing methods.

In Braidot's approach [17], motion capture was proven to be effective using Microsoft Kinect, the commercially available RGB-D camera. With the use of its native Application Programming Interface (API), it allows to recognize and automatically track people in real-time skeleton through Randomized decision forest trees that to predict 3D position accurately even without information on its time The space data for each coordinate system of the skeleton are generated after the runtime has determined the individual body parts.

In the experiment of Paraskevopoulos *et al.* [2] which aimed to recognize 8 simple gestures, after extracting the 3D skeletal joint coordinates from the Kinect for vector and displacement calculations, they trained a handful of machine learning algorithms from Scikit-Learn such as KNN, Random Tree (RT), Lagrangian Support Vector Machine (LSVM), Endurance Time (ET), etc. They used k-cross validation to evaluate their results and ET garnered the best average accuracy of 92.3%.

III. FRAMEWORK

Braidot *et al.* [17] showed the viability of using RGB-D capture technology for tracking the human body to obtain important kinematic information about a player's technique. They showed that by tracking the lower body, they were able to reliably obtain information about how a player's knees were positioned and thus were able to successfully find individuals with potential/probably varus or valgus knee injuries based on a simple criterion based on the angles they were able to compute. Others were able to apply the use of RGB-D sensors for tracking the pose of weightlifters to determine if their technique in performing their lift [1, 18]. Torralba *et al.* [1] were able to show the accuracy of a model that tracked the angle of the shoulder, elbow, and hand similarly to Braidot *et al.* [3] and using it in tandem with a machine learning algorithm. In their study, they applied an Extremely Randomized Trees classifier algorithm to classify weightlifting poses as correctly or incorrectly performed with an accuracy of 80%–91% depending on the lift. Hu *et al.* [3] in their study were able to apply the use a camera motion capture system to track the movements of a basketball player during different maneuvers and showed that they were able to recognize different activities by using a dynamic time warping on the time series of different body part motions and computing the overall difference between an observed activity to their developed templates for each activity, with the model determining which maneuver was attempted based on the least overall difference and Rahma *et al.* [4] gathered Hu moments from clips by applying background detection and removal to track a human subject. These studies show the clear usefulness and viability of RGB-D and other visual sensors in the issue of gathering information regarding the pose, 27 posture, and kinematic motions of a person as well as the viability of using this data in tandem with machine learning classifiers to produce useful results. Based on Fig. 1, it represents the approach the researchers will use while developing the model which will be used in determining whether a maneuver was performed with proper or improper technique.

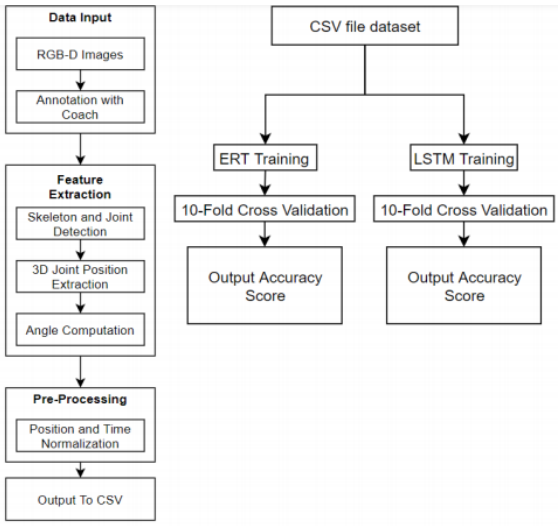

Figure 1. Conceptual framework.

The proponents will use an RGB-D sensor to capture RGB-D images of players performing various basketball maneuvers and will annotate these recordings as executed properly or improperly based on the judgment of a qualified coach, these labels will be considered the ground truth. The proponents will then extract from each frame the positions and coordinates of the head, spine, shoulders, elbows, hands, wrists, hips, knees, ankles, and feet similar to previous works [1, 3, 10, 11] and then compute the angle created by the hands, elbows, shoulders and hips, knees, feet to provide the machine learning algorithms more relevant data to train on [1]. Computing the angle between the three points (A, B, C) is done by taking the vector formed between points A and B (BA) and points B and C (BC) and then applying the following formula to find the angle θ between them based on Eq. (1).

$$\theta = \arccos\left(\frac{BA \cdot BC}{\|BA\|\|BC\|}\right) \quad (1)$$

Once these features are extracted the proponents will normalize the data set by first using a "peak detection and alignment" technique based on the work of Hu *et al.* [3] using the motion of the shoulders and hands to align the attempts with each other to reduce the noise in the data introduced by different players taking different lengths of time 28 to perform the same maneuver. The coordinates of the tracked points will also be normalized by taking the value of the coordinates of the middle torso and subtracting it from all the tracked values, this will put the middle torso at the point of origin and will make all other points relative to the position of the origin. This will be done to attempt to reduce the noise in the movement of the tracked points due to a combination of the movement of the entire body and the movement induced by muscles.

To keep track of the momentum of the player the original coordinates of the origin will still be available for the machine learning algorithms. Once all this data is gathered and processed it will be fed through two machine learning algorithms to learn how to classify them based on the annotations the coach provided. The Electrical Resistivity Tomography (ERT) algorithm and the Long Short-Term Memory (LSTM) algorithm will be used to classify and distinguish whether the maneuver was performed properly. Once the data has been collected the proponents will use a 10-fold cross-validation scheme to determine which of the two machine learning algorithms performed better.

IV. METHODOLOGY

This chapter discusses the way the study will be conducted as well as the tools. This chapter shall cover details regarding the tools and libraries that the researchers will be using as well as describing their approach to data gathering, pre-processing, and testing of the research.

A. Hardware and Tools Used

This section covers the hardware and software used by the proponents in conducting the research.

- Microsoft Kinect v2

The proponents shall use the Microsoft Xbox One Kinect v2 as their primary data-gathering tool and a laptop to run the software needed for the prototype. The Kinect v2 is a more recent model that introduces enhanced features when compared to the Kinect v1 [3].

The Kinect v2 has improved features compared to its predecessor. The Kinect v2's Motion tracking is better than the Kinect v1 due to the improvements in the resolution of the cameras and the number of skeletal joints it can track which includes the person's thumb. Kinect v2 uses "time of flight" technology to determine the "depth" or distance of points from the camera. It is also capable of individual tracking of fingers and the stretching and shrinking movements with hands and arms.

- Microsoft Kinect SDK

The Kinect for Windows Software Development Kit (SDK) 2.0 enables the development of applications that performs gesture, motion, and voice recognition using the Kinect V2. It contains the essential drivers necessary to run the Kinect as an input device for a machine running a Windows-based operating system.

- Kinect PV2

Kinect PV2 is an open-source library developed for Processing 3.0 capable of performing joint and skeleton tracking using the RGB-D Kinect input. Processing 3.0 is another open-source project readily available online which was started by Ben Fry and Casey Reas in the Spring of 2001 as a data visualization software platform to help teach programming fundamentals and has now turned into a development tool for professionals. Kinect PV2 provides the functionality to detect and track joints as well as the functionality to save and store this information in the form of a Comma-Separated Values (CSV).

- Scikit-Learn

Scikit-Learn is a free Python machine-learning library originally developed by David Cournapeau. It is a very popular machine-learning library that has implementations of many of the most popular machine-learning methods. The proponents will use the Scikit-learn implementation of extremely random trees for the creation of the first model.

- Tensorflow

TensorFlow is an open-source symbolic math software library commonly used for machine learning. The library was developed by the Google Brain team on November 9, 2015. The proponents will use Tensorflow's Keras API implementation of LSTM RNNs for the creation of the recurrent model.

B. Data Gathering

A total of ten participants will be asked to perform a free throw, a jump shot, and a layup at least 20 times each while being recorded using the Kinect v2 under the supervision of one specific coach who will assist in manually labeling the technique of the attempts as proper or improper, these labels will be used as the ground truth. The setup will require a well-lit environment either indoors or outdoors as seen in Fig. 2. The Kinect will be placed 1.4 meters away from the player as it is the recommended distance for capturing single actors and it will capture the front of the player. Only one coach will be supervising the labeling so that the labels are consistent with the training. The participants will be given five seconds to perform the task during which they will be recorded by a KinectV2 at 60 frames per second. This will generate 300 frames per attempt and 6000 frames per participant. Each attempt will be saved individually with its frame value (which can be viewed as its temporal marker) and kept separate from each other. Should the data set contain an unreasonably uneven split between proper and improper forms (such as 80%–90% proper or improper) for any of the tasks, the proponents shall attempt to record the offending task again with the same coach until a more appropriate ratio is achieved. This is to avoid problems with unbalanced classes during training which tends to create unoptimized classifiers for new input the model has not seen.

Figure 2. Experimental setup where the maneuver is recorded in front of the participant.

C. Feature Extraction

The proponents of this research used the Kinect PV2 for feature extraction on the Kinect input data. The Kinect PV2 library using the Kinect SDK will be used to identify the subject which will detect and track their joints and other landmark features. The Kinect PV2 will then extract the 3D position of these landmarks. The data will then be compiled into a Comma-Separated Values (CSV) file with correct and incorrect techniques differentiated from each other using the labels the coach provided. Process can be seen on Fig. 3.

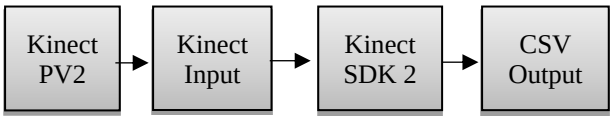

Figure 3. Feature extraction.

D. Pre-processing

To increase the efficiency and speed of learning for the classifier algorithms, the data will first be pre-processed to apply some uniformity to the data set.

To account for the difference in acceleration and position relative to the sensor present for each trial, each axis data (X, Y, Z) for the middle torso will be subtracted from all the other tracked features. This will make the middle torso the point of origin for the coordinates, and it allows the data to be viewed as movements relative to the origin which negates the noise in the data generated by the motion of the entire body compounded with the motion of individual limbs. This will allow the algorithms to be able to view the motion of the body parts separately from the motion of the entire body.

Another issue with the data is within the 5-second time window allotted for each attempt, individual attempts may be started earlier or later than other attempts (such as a player lingering for a second longer in the ready position than they did previously). To align the frame sequences a "peak detection and alignment" was implemented based on the work of Hu *et al.* [3] using the position of the shoulders and hands. This will be used to adjust the start of each attempt to standardize the starting time of each attempt.

Finally, due to the importance of the angle that the knees and elbows create in performing these basketball maneuvers this angle will be computed and added to the data set using Eq. (1).

E. Dataset

The proponents shall use the Kinect v2 camera for capturing video of at least 10 different collegiate basketball players performing 20 free throws, 20 jump shots, and 20 layups each. Using the data gathering methodology and Feature Extraction above, this will create a data set of RGB-D and tracking data of 200 free throws, 200 jump shots, and 200 layups. Each of these entries will be tagged with the assistance of a coach to determine if they were performed properly.

F. Model Development and Testing

The CSV files will then be used to train two classifiers, one with a recurrent neural network using a long-short term memory architecture, and one with extremely random trees which is a decision tree type classifier. These classifiers will be trained to classify inputs as proper or improper shooting forms. These classifiers will be evaluated based on how well they can be trained to recognize a proper shooting posture/technique using a 10-fold cross-validation scheme. In the 10-fold validation scheme, the data set will be split into 10 equal-sized partitions at random and then the model will be tested 10 times using 9 partitions to train the model and 1 partition to validate each time, through the 10 times the model is trained each partition will be used to validate it once and only once. The 10-fold cross-validation was chosen to split the data set of 200 attempts evenly into 10 groups of 20 randomly chosen entries to ensure enough trials are run. 10 folds will be used to reduce the bias of the models, this is done to lower the variance of the result. Once this 10-fold cross-validation is completed the proponents will then be able to assess the model skill of each model used in making the model.

V. Results and Discussion

To obtain the results, each of the three RGB-D Kinect datasets, namely the jump shots, free throws, and lay-ups were fed into two separate models with each having its own machine-learning algorithm. The first is LSTM while the other is ERT, and both were validated with 10-fold cross-validation.

A. ERT Results

The ERT model was able to produce an average accuracy of 93.5% for the Free Throw, based on Fig. 4, then 78% for the Jump Shot, based on Fig. 5, and finally, at about 76.5% for the Lay-ups based on Fig. 6.

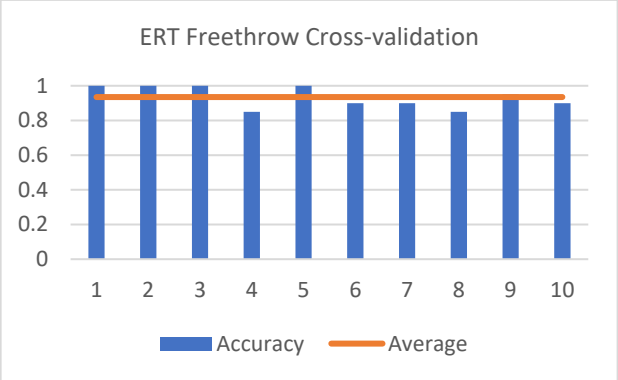

Figure 4. ERT freethrow cross-validation.

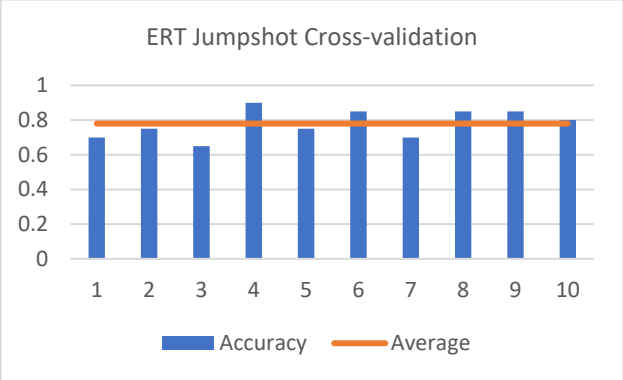

Figure 5. ERT jump shot cross-validation.

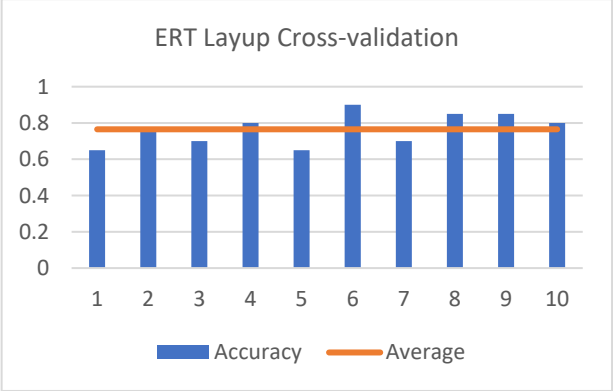

Figure 6. ERT layup cross-validation.

B. LSTM Results

As for the LSTM model, it produced an average accuracy of 73.5% for The Free Throws as seen in Fig. 7, then 68% for the Jump Shots based on Fig. 8, and finally at about 70% for the Lay-ups as seen in Fig. 9.

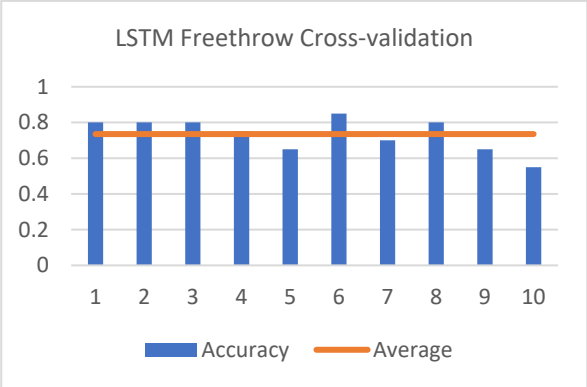

Figure 7. LSTM freethrow cross-validation.

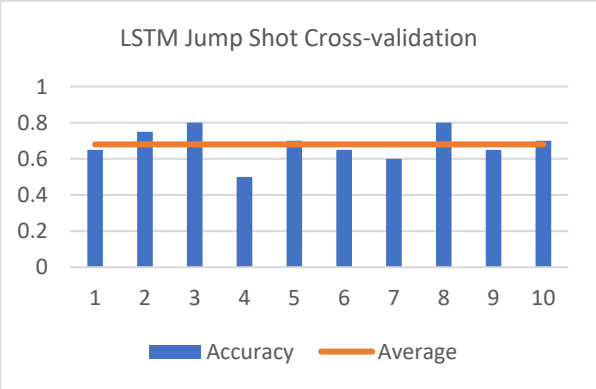

Figure 8. LSTM jump shot cross-validation.

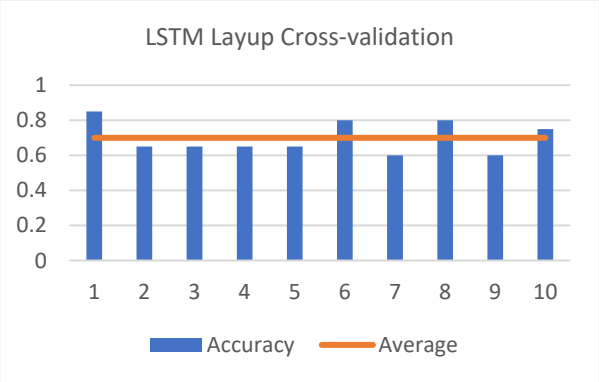

Figure 9. LSTM layup cross-validation.

C. Findings

Based on the above results, the ERT model was able to have better accuracy than the LSTM model. The ERT model provided higher accuracy for all the maneuvers under examination. While the ERT model had a higher accuracy the LSTM model was also able to consistently achieve over 90% accuracy during training on its epochs throughout the folds and still provided promising results at the end. The ERT model, having an average accuracy of 82.66% across the three maneuvers, outperformed the LSTM which was only able to garner an average accuracy score of 70.5%.

Both classifiers were able to provide the highest accuracy for free throws, indicating it is the easiest maneuver to classify while the accuracy of the other maneuvers showed a significant dip in accuracy. This can likely be attributed to the simplicity of performing the free throw as the subject is stationary as opposed to the other two dynamic maneuvers where the jump shot requires the subject to jump and the layup requires the subject to run and jump. The data set shows that the Kinect tracking sometimes failed for a few frames at a time for the two dynamic maneuvers causing irregularities in the data set as well as the inherent increase in complexity of a dynamic maneuver is likely the reason for the difference in accuracy. Both models were proven to be capable of determining if a maneuver was performed properly or improperly.

VI. Conclusion and Recommendation

The proponents experimented to determine the effectiveness of applying ERT or LSTM models to classify whether a specifically observed basketball maneuver was performed correctly or incorrectly. The proponents gathered data from 10 individuals performing 20 Free Throws, Jump Shots, and Layups using a Kinect V2 to track the 3D position of joints on the subject. These coordinates, along with a computed angle between the limbs, were saved on a CSV file which was used as the data set for a 10-fold cross-validation scheme to determine the accuracy of each model. The performance of both the LSTM and ERT models was able to produce substantial results. The average accuracy across the 10-fold cross-validation of the ERT model was better than the accuracy of the LSTM model. The results clearly show the viability of using the ERT model in determining whether a given Free Throw, jump shot, or Layup was performed properly or improperly.

The proponents of the study recommend continuing further investigations into the performance of these two classifier algorithms for this application using a larger data set due to both providing promising results from the current limited data set. An attempt at creating an ensemble classifier using these two models could also be attempted. Future research could also gather data from an actual game and apply a similar methodology; however, the Kinect V2 sensor was sufficient in this application due to the lack of visual obstructions but in a game, the occlusion caused by players constantly moving around would cause problems for tracking the players. A different non-visual-based tracking device such as a wearable accelerometer would be more advisable for such an application.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Kevin Lucero and Renz Justin Thomas conducted the data gathering. Christian Jose Ajero assisted in writing the paper. John Paul Tomas provided supervision over the study. All authors had approved the final version.

ACKNOWLEDGMENT

The researchers would like to give thanks to our thesis adviser, Prof. John Paul Q. Tomas; panelists Prof. Mary Jane Samonte, Prof. Joel C. De Goma, Dean Ariel Kelly Balan, Ma'am Sally Zara; and thesis coordinator, Dr. William Torres. We also thank Dr. Bernardino C. Ofalia, Ed.D. Educational Administration, MA Psychology, and coach Eric Canzon at De La Salle University.

REFERENCES

[1] A. J. Torres, C. Silubrico, D. Torralba, and J. P. Tomas, "Detection of proper form on upper limb strength training using extremely randomized trees for joint positions," in *Proc. the 2nd International Conference on Computing and Big Data*, 2019.

[2] G. Paraskevopoulos, E. Spyrou, and D. Sgouropoulos, "A real-time approach for gesture recognition using the Kinect sensor," in *Proc. the 9th Hellenic Conference on Artificial Intelligence*, 2016.

[3] X. Hu, *et al*., "Basketball activity classification based on upper body kinematics and dynamic time warping," *International Journal of Sports Medicine, U.S. National Library of Medicine*, Apr. 2020.

[4] A. M. Rahma, M. A. Rahma, and M. A. Rahma, "Automated analysis for basketball free throw," in *Proc. 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2015.

[5] R. Ji, "Research on basketball shooting action based on image feature extraction and machine learning," *IEEE Access*, vol. 8, pp. 138743–138751, 2020.

[6] J. Abian-Vicen, C. Puente, J. J. Salinero, C. González-Millán, F. Areces, G. Muñoz, J. Muñoz-Guerra, and J. Del Coso, "A caffeinated energy drink improves jump performance in adolescent basketball players," *Amino Acids*, vol. 46, no. 5, pp. 1333–1341, May 2014.

[7] G. Li and C. Zhang, "Automatic detection technology of sports athletes based on image recognition technology," *EURASIP J. Image Video Process*, vol. 2019, no. 1, pp. 138–150, Dec. 2019.

[8] K. E. Mah, E. J. Kezirian, and W. C. Dement, "Changes in basketball shooting techniques in the US from the 1920s to the 1940s: Through to the diffusion of the one-hand shot from middle distances," *J. Cell Sci.*, vol. 119, no. 16, pp. 3316–3324, Apr. 2006.

[9] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu, "Action recognition in still images with minimum annotation efforts," *IEEE Trans. Image Process*, vol. 25, no. 11, pp. 5479–5490, Nov. 2016.

[10] V. H. Okazaki, A. L. Rodacki, and M. N. Satern, "A review on the basketball jump shot," *Sports Biomechanics*, vol. 14, no. 2, pp. 190–205, 2015.

[11] H. Okubo and M. Hubbard, "Comparison of shooting arm motions in basketball," *Procedia Engineering*, vol. 147, pp. 133–138. 2016.

[12] C. Chen and J.-M. Odobez, "We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video," in *Proc. IEEE Conf. Compute. Vision Pattern Recognition*, Jun. 2012, pp. 1544–1551.

[13] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, *et al*., "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[14] W. Liu, T. Xia, J. Wan, Y. Zhang, and J. Li, "RGB-D based multi-attribute people search in intelligent visual surveillance," in *Proc. 18th Int. Conf. Adv. Multimedia Modeling*, 2012, pp. 750–760.

[15] E. S. d. Motta, A. C. Sementille and I. A. Aguilar, "Development of a method for capturing human motion using an RGB-D camera," in *Proc. 2017 19th Symposium on Virtual and Augmented Reality (SVR)*, 2017, pp. 97-106.

[16] R. Al-Akam and D. Paulus, "RGBD human action recognition using multi-features combination and k-nearest neighbors classification," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017.

[17] A. Braidot, G. Favaretto, M. Frisoli, D. Gemignani, G. Gumpel, R. Massuh, and M. Turin, "The valuable use of Microsoft Kinect™ sensor 3D kinematic in the rehabilitation process in basketball," *Journal of Physics: Conference Series*, vol. 705, 012064, 2016.

[18] S. Sutthiprapa, V. Vanijja, and T. Likitwon, "The deadlift form analysis system using Microsoft Kinect," *Procedia Computer Science*, vol. 111, pp. 174–182, 2017.