JOURNAL OF
EMERGING INVESTIGATORS

Article

Levering machine learning to distinguish between optimal and suboptimal basketball shooting forms

Hrishikesh Deosthali[1], Anthony Cuturrufo[2]
[1] Liberal Arts and Science Academy, Austin, Texas
[2] University of California Los Angeles, Los Angeles, California

SUMMARY

Basketball is a highly competitive sport and good shooting form is crucial to a player's success. For high shooting accuracy, several body parts must be aligned, including proper leg positioning, elbow placement, hand posture, and shooting wrist curvature. With the progress in machine learning, Artificial Intelligence (AI) tools can be developed to provide feedback and personalized guidance on basketball shooting form. In this research, we compared Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) to identify the most suitable model for integration into AI tools meant for basketball shooting form analysis. We hypothesized that CNN models will perform significantly better for basketball shooting form analysis than RNN or MLP models because CNN models are known to be better suited for Human Action Recognition (HAR) than other models. We evaluated five models to test our hypothesis - an MLP and an RNN model using Cartesian coordinates of body joints, an MLP model using angles of body joints, and two video-based CNN models using raw and cropped video data. Contrary to initial expectations, the accuracy of the MLP model with Cartesian coordinates of body joints outperformed the CNN model with cropped video (88.3% versus 83.3%). Since MLP models typically require less computational resources, they can be used to build efficient AI tools for basketball shooting form analysis in resource-constrained environments such as mobile phones.

INTRODUCTION

Basketball is an extremely competitive sport, and shooting accuracy is a key determinant of a game's outcome (1). Shooting is one of the most challenging skills to be developed in basketball (2). It is known that coaching, conditioning camps, practice games, and self-training help improve the shooting performance of basketball players (3). However, basketball players often lack guidance during their formative years, which can result in the development of incorrect postures and forms. If left unaddressed, these errors can become deeply ingrained habits, leading to incorrect techniques, poor player performance, and a higher risk of injuries.

Correction of these habits in the later stages of the basketball journey is a challenging task for both the coaches and players. Self-training to perfect the body components necessary for optimal form is often difficult due to the lack of immediate feedback. Artificial Intelligence (AI) tools using computer vision can be developed to address this challenge. Computer vision in sports is gaining popularity for applications such as player tracking, trajectory analysis, personal and team performance analysis, and detecting and classifying objects (4). Previous research has focused on identifying actions in still images and adding a temporal aspect for video analysis (5). Contemporary basketball shooting analysis includes data pre-processing, feature extraction, time series analysis, and machine learning optimization (6). Convolutional Neural Networks (CNN) have been shown to be an effective algorithm for developing machine learning (ML) models to analyze basketball players' shooting actions (7). Similarly, Histogram of Oriented Gradients (HOG), Support Vector Machines (SVM), Naive Bayes (NB), Recurrent Neural Networks (RNN), and Multilayer Perceptron (MLP) are popular algorithms used for developing action recognition models in basketball and other sports (8, 9). Machine learning for human action recognition is still an open research topic, and the application of AI in basketball is still in its infancy (10, 11). Limited research has been done on the analysis of basketball shooting form and the development of AI tools to improve player skills.

Our research focused on finding the model with the highest accuracy in predicting whether a basketball player's shooting form while making a shot was optimal or suboptimal. We hypothesized that CNN models would perform significantly better than RNN and MLP models to analyze a basketball player's body posture while making a shot. The primary reason is that CNN models are well-suited for capturing spatial dependencies and hierarchical features, making them ideal for tracking a basketball player's shooting form (12 - 14).

We analyzed and compared multiple machine learning models 1) MLP with Cartesian coordinates of body joints, 2) MLP with angles of body joints, 3) RNN using Long Short-term Memory (LSTM) with Cartesian coordinates of body joints, 4) 3D CNN with raw video, and 5) 3D CNN with cropped video. To train and test these models, a total of 225 videos were collected from two sources - UCF101, an Action Recognition dataset published by the University of Central Florida, and self-recorded videos. The five models were evaluated using a binary classification approach to differentiate between an optimal and a suboptimal shooting form. Contrary to expectations, the MLP model with Cartesian coordinates of body joints performed better than the CNN models.

This research proposes a suitable resource-efficient model with an accuracy above 80% to analyze a basketball player's shooting form. Additional research will be needed to

https://doi.org/10.59720/24-046

enhance the accuracy of the models under various lighting and background conditions, video angles, video qualities, player physiologies, and camera angles. Multimodal models combining MLP and CNN could also potentially enhance overall accuracy, but more work will be required to confirm applicability to basketball form analysis (15).

RESULTS

A total of 225 videos of basketball shots with optimal form and suboptimal form were collected from the UCF101 Action Recognition dataset and self-recorded videos (16). From this input dataset, 158 videos were used to train these AI models, while the remaining 67 videos were used to test the accuracy of the models (**Table 1**).

For Model 1, the Cartesian coordinates of each body joint in all video frames were concatenated in temporal order before passing to the neural network model. The accuracy of all models was further enhanced by tuning hyperparameters, such as hidden layers, the number of hidden units in each layer, the dropout, batch size, and epochs (**Table 2**). With the tuned hyperparameters, the MLP model with Cartesian coordinates of body joints achieved the highest validation accuracy of 88.3% (**Model 1, Table 1**).

For the shots with suboptimal form, the majority of the y coordinates of the left elbow were below the value of 0.425, indicating the ball was shot at a higher position, while the y coordinates of the left knee were primarily below the value of 0.675 indicating that there was less bend in the knees (**Figure 1**). Similarly, for shots with suboptimal form, the majority of z coordinates from the left elbow and left knee were located above the values of -0.3 and -0.2, respectively, making it evident that the ball was held too far out (**Figure 1**). For the x coordinates, there was no distinction between shots with optimal and suboptimal forms.

Model 2 replaced Cartesian coordinates with angles of each body joint. The angles were derived using the Mediapipe Pose landmark model and a 3-point formula (**Table 3**). Model 2 yielded only 68% validation accuracy, suggesting that angles were insufficient to model the body form of a basketball player precisely.

Model	Validation Accuracy
Model 1 (MLP model with Cartesian coordinates of body joints)	88.3 %
Model 2 (MLP model with angles of body joints)	68.3 %
Model 3 (LSTM model with Cartesian coordinates of body joints)	69.05%
Model 4 (3D CNN with Raw Video)	73.3 %
Model 5 (3D CNN with Cropped Video)	83.3 %

Table 1: Validation accuracies of machine learning models used to analyze basketball shooting form. The validation accuracy was the highest at 88.3% with Model 1 (MLP with Cartesian coordinates of body joints), while Model 5 (3D CNN with cropped video) showed a validation accuracy of 83.3%. Other models performed worse than Model 1 and Model 5.

Model 3 used a RNN model trained on sequential time series data to analyze videos. We used the LSTM model and passed Cartesian coordinates of body joints to this model (the same ones used in Model 1) for all video frames. The LSTM model processed this input data array to build a machine-learning model. The LSTM model was trained through 40 epochs, with the highest validation accuracy being 69.05%. Two instances of 3D CNN were evaluated – one with raw video (**Model 4, Table 1**) and another with cropped video (**Model 5, Table 1**). Raw videos used data from the UCF 101 data set without any modification. The cropped videos removed most of the background information in the video frame and left primarily the player in view. Model 4 achieved an accuracy of 73.3%, while Model 5 achieved a validation accuracy of 83.3%.

The validation accuracies for Models 1 and 5 were significantly different (p-value<0.05, t-test), indicating that the difference in accuracies between Model 1 and Model 5 was unlikely to have occurred by chance. Contrary to our hypothesis, Model 1 showed higher validation accuracy than Model 5.

DISCUSSION

Validation accuracy is an objective measure of whether a machine learning model has been trained effectively to predict outcomes with a high degree of certainty. It is expressed in percentages, with a higher percentage indicating the model's better prediction capability. Model 1 showed the highest validation accuracy at 88.3%. The high accuracy of the MLP model with Cartesian coordinates of body joints (Model 1) indicates that the coordinates of body joints strongly correlate to a basketball player's shooting form.

Model 2 (MLP model with angles of body joint) achieved a validation accuracy of 68.3%. Intuitively, Model 2 should show similar accuracy as Model 1, but the validation accuracy showed that the angles of body joints are insufficient for effectively comparing body postures. One potential reason could be that the Cartesian coordinates (x,y,z) define a unique point in 3D space and provide comprehensive spatial information about the body posture. On the contrary, angles lack information about the rotation or orientation of those joints in space. Therefore, the same joint angle could be derived through multiple configurations of the body joint. This could make it difficult for the model to compare body postures in different videos.

Model 3, which used the Stochastic Gradient Descent optimizer, originally had issues with exploding gradients, evident with validation loss showing invalid values. To mitigate this problem, hyperparameters were tuned, applying

Model	Learning Rate	Hidden Layers	# Hidden Units	Epochs	Dropout	Batch Size
1	0.0001	3	(128, 64, 32)	40	0.1	8
2	0.001	3	(128, 64, 32)	50	0.3	32
3	0.0001	3	(128, 64, 32)	40	NA	20
4 & 5	0.0001	4 Conv2Plus1D layer Residual Block layers GlobalAveragePooling3D layer Flatten Layer		12	NA	32

Table 2: Hyperparameter values used for all models. The table shows hyperparameter values tuned to maximize the validation accuracy of each model. These parameters were manually tuned for each model.

https://doi.org/10.59720/24-046

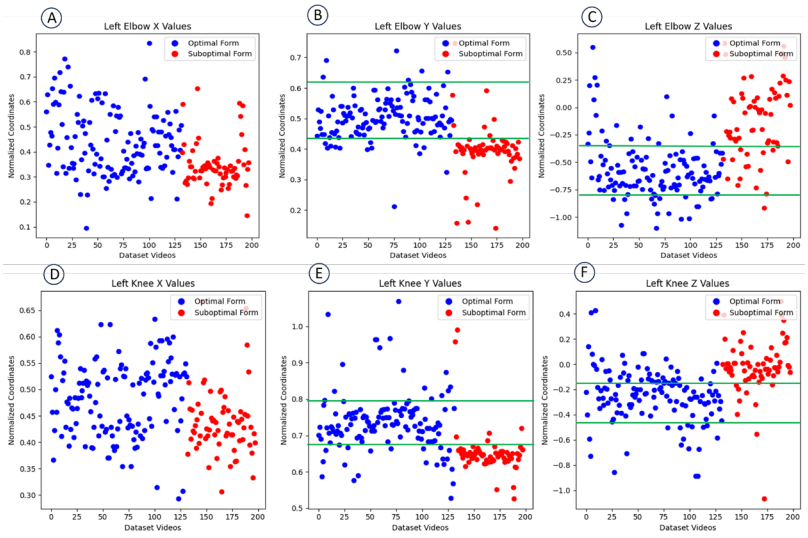

Figure 1: Spread of X, Y, and Z coordinates of left elbow and knee. The Cartesian coordinates of each joint of a player's body were extracted using Mediapipe Pose library. Coordinates of the **A-C)** left elbow and **D-F) l**eft knee. A clear distinction exists between the Cartesian coordinates of an optimal and suboptimal basketball shot. The y and z coordinates of shots made with optimal form are located within a range, while those with suboptimal form are outside the range. This indicates incorrect hand, elbow, and knee positioning for suboptimal shots.

a lower learning rate and gradient clipping. This reduced the exploding gradient issue, yet it did not eliminate it. With these persistent issues, the LSTM model could not surpass an accuracy of 69.05% because of failure in learning patterns.

Model 4 and Model 5 achieved 73% and 83% validation accuracies, respectively. For Model 4, the raw videos contained a lot of background information, in addition to the basketball player, which may have caused the CNN Model to train on the wrong set of assumptions. This can introduce bias, an error introduced in the machine learning model when it trains on the wrong information in the input data. It is possible that Model 4 had low validation accuracy due to bias. Model 5, on the other hand, used cropped videos using a bounding box around the basketball player. This significantly reduced the background information and may have helped reduce the bias introduced by the raw footage, allowing the model to get a more accurate representation of the body form during shooting.

Joint Angles	Landmarks used for 3-point angle formula
right elbow	right shoulder, right elbow, right wrist
left elbow	left shoulder, left elbow, left wrist
right knee	right hip, right knee, right ankle
left knee	left hip, left knee, left ankle
right ankle	right knee, right ankle, right heel
left ankle	left knee, left ankle, left heel
right wrist	right elbow, right wrist, right pinky
left wrist	left elbow, left wrist, left pinky

Table 3: Calculation of body joint angles. The Mediapipe Pose landmark model uses 33 landmarks to represent a body (21). We used these landmarks to determine angles for joints. The table shows which landmark points were used to calculate angles for body joints.

The high accuracy of the MLP model with Cartesian coordinates of body joints implies that the spatial and temporal details captured by CNN may not have significant advantages over the features captured in the MLP model. The higher validation accuracy of Model 1 compared to Model 5 does not support the hypothesis that the CNN models would perform significantly better than other models for basketball shooting form analysis.

A key benefit of MLP models is that they can drive efficient implementations in resource-constrained environments, such as mobile phones. The number of independent features representing information in an image or a video frame is called dimensionality (17). The MLP models use a single set of features (Cartesian coordinates or angles of body joints) to build information about basketball shots compared to other models that use complete 3D video. This low dimensionality drives low computational requirements such as storage memory (RAM), memory bandwidth (data transfer speed from memory to processor), and Tensor Operations Per Second (TOPS).

The accuracy of MLP models can potentially be improved further with the use of body sensors. Further research will be required to look at the trade-offs between the impact of sensors on body posture, an improvement in accuracy, and an increase in cost.

Similar to the MLP models, 3D CNN models can also drive efficient implementations using grayscale videos. Grayscale videos can help reduce computational complexity since they include only intensity values per pixel. This can lead to faster processing times, lower memory requirements, and lower computational costs. However, the accuracy of machine learning models with RGB (Red-Green-Blue) color videos tends to be higher for human action recognition since the videos contain more information to distinguish between objects and actions (18). Additional research will be required to analyze the impact of grayscale vs RGB videos for basketball form analysis.

The video dataset for all five models reflected a broad

https://doi.org/10.59720/24-046

spectrum of player postures categorized as 'optimal' and 'suboptimal.' However, the execution of a basketball shot will differ across players. The speed of the shot, how high the player jumps, and the angle of the shot will differ across player physiologies. Therefore, machine learning models must be trained and tested across physiologic variations to make them more accurate and reliable. Similarly, ambient lighting, camera quality, and angles affect the dynamic range and contrast in a video, which can adversely impact the model's ability to accurately identify an action. Camera quality and lighting can also create shadows and reflections, which can make the model interpret incorrectly. Even though we trained these models across multiple players and lighting conditions, additional data may help enhance the accuracy of these models further.

This research showcased accurate machine learning models for shooting form analysis of a basketball player. Both the MLP model with Cartesian coordinates of body joints and the CNN model with cropped video can be used to develop an efficient mobile phone app, which a basketball player can use for self-training and shooting form improvement. With additional research on the topics listed above, the machine learning models can be deployed for use in live broadcasting events as well as professional player training.

MATERIALS AND METHODS

Validation accuracy is calculated on data that has not been seen previously by a trained machine learning model. It is defined as:

$$Validation\ accuracy = \frac{\alpha}{\alpha+\beta}, where$$

$$\alpha = number\ of\ correct\ predictions$$

$$\beta = number\ of\ incorrect\ predictions$$

$$\alpha+\beta = total\ number\ of\ predictions$$

The machine learning model was trained on a set of input data and then used for inference on unseen data. For training and inference, we used videos from a human action recognition database hosted by the University of Central Florida and self-shot videos (16).

Data Collection

The Center for Research in Computer Vision at the University of Central Florida has several action recognition datasets, including basketball shooting videos. We used the UCF101 dataset because it provided several quality videos for the models (each video had a frame rate of 30 fps) (16). From the UCF101 dataset, we obtained 135 videos of optimal shooting form and 10 of suboptimal form. To adjust this imbalance in data, we used a cell phone camera and recorded 80 self-videos of basketball shots with suboptimal form. Each basketball shot demonstrated factors of bad forms, such as the body twisting or the wide angle of the elbow. In the end, the dataset consisted of 225 videos, of which 70% were used for training the machine learning models, and 30% were used for testing the models.

The input videos were manually tagged as optimal or suboptimal based on the fundamental principles of form shooting, including elements such as having shoulders aligned and elbows tucked in. Note that an optimal form cannot always correlate to the success of a basketball shot since a player may be able to put the ball through the basket with a bad form.

Each video was edited using Canva video editor tool to start from the point the player got the ball in their hands to the point the player's legs landed on the court (jump shot) or the point when the ball left the player's hand (non-jump shot).

Data Pre-Processing

Preliminary testing on raw videos with the five neural network models yielded a validation accuracy of <60%. After investigation, we found that the player's placement in the video impacted the ability of the models to correlate the actions between the two videos. This impacted the accuracy of the models. Therefore, we decided that a bounding box around the player would be necessary to improve accuracy. The bounding box centered the field of view on the basketball player while reducing the amount of background information. This was done by locating three key points of the player's body - left leg, right leg, and nose – using the MediaPipe libraries (19). With the position of these three body points, each video was cropped by adding a padding of 175 pixels to the coordinates of all 3 points (**Figure 2**). The cropped videos were saved to a new file using the CV2 video writer. This served as a new dataset for the models.

Methodology

All models except Model 4 used cropped video using a bounding box around the basketball player. Model 4 used the raw video as input. The data then went through a pre-processing stage to get the correct parameters for the models (**Figure 3**). Each model used libraries from Mediapipe, OpenCV, or Tensorflow. Mediapipe is an open-source framework from Google that includes libraries for multimedia processing, OpenCV is an open-source software library for computer vision and machine learning, and Tensorflow is an open-source framework that helps developers build machine learning models. The source code for this research is available on Github (github.com/rishydeosthali/ML-Shooting-Forms).

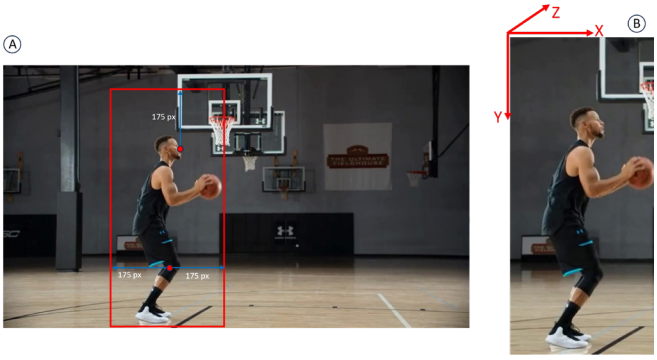

Figure 2: The process of adding a bounding box centered around the region of interest (basketball player). A) Raw input video contains unnecessary background information compared to the region of interest (basketball player). This generates a bias in the CNN machine learning model, leading to low validation accuracy. **B)** Model accuracy increases significantly after the video is cropped around the central axis generated from the left knee, right knee, and nose. A padding of 175 pixels was added to the three points before cropping. The coordinate axis system used to generate Cartesian coordinates of body joints is given in the upper left corner.

https://doi.org/10.59720/24-046

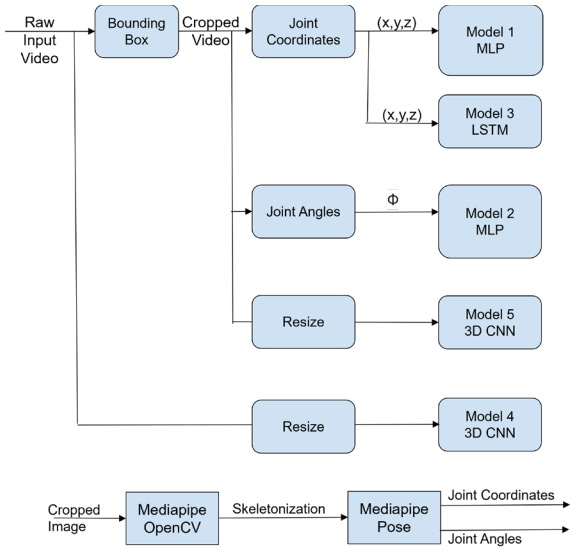

Figure 3: Architecture for evaluating the accuracy of machine learning models for basketball shooting form. Methodology used by each machine learning model. Model 4 used the raw video as input, while all other models used a video cropped using a bounding box around the basketball player. Models 1-3 used Mediapipe libraries to skeletonize the image and extract Cartesian coordinates and angles.

Model 4 used raw video from the dataset as input, while all other models used cropped videos. All models used more than one intermediate layer. The weights and biases between the connections of neurons were adjusted using backpropagation. Backpropagation is a feedback mechanism used in machine learning to reduce error and improve accuracy during training. First, the difference between the model's prediction and target value is calculated using a mathematical process, also called loss function. We used the Binary Cross Entropy loss function in our training. The backpropagation process then adjusts and optimizes the weights and biases of neurons to minimize the error between predictions and target values.

Each model underwent multiple iterations (epochs) over the entire dataset to fully train and tune the parameters. After testing various parameter configurations, the optimal values of the learning rate, number of hidden layers, number of hidden units in each layer, the dropout, batch size, and epochs were finalized to maximize the accuracy of each model **(Table 2)**. All models used the Adam optimizer except LSTM, which used the Stochastic Gradient optimizer.

Model 1 (MLP with Cartesian coordinates of body joints)

Each video image frame was passed through Mediapipe and OpenCV libraries to create a skeleton image of the player **(Figure 4)**. This process, called skeletonization, generates a unit-width skeleton of an object by removing the outermost pixel layers. This simplifies the object shape, which is helpful for shape analysis. Using the Mediapipe Pose library, we extracted the x, y, and z coordinates of each joint of a player's body for each video frame. We used the OpenCV coordinate system, which uses the top left corner of a video frame as the origin, with the z value indicating the depth. All coordinates were normalized using z-score normalization, which standardizes the data with a mean value of 0 and a standard deviation of 1.

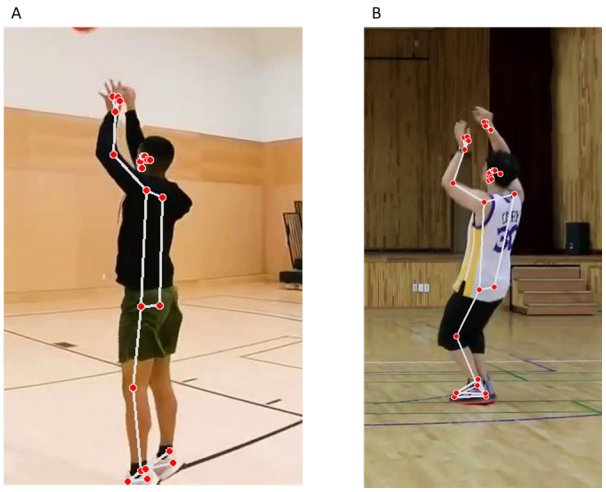

Figure 4: Skeletonization of basketball player in the input video frame. This process is used to extract Cartesian coordinates and angles of body joints. Each video frame is passed through Mediapipe and OpenCV libraries for skeletonization. The skeletonized image is passed through the Mediapipe Pose library to extract Cartesian coordinates and angles of the body joints. The Cartesian coordinates and angles are then sent to the MLP models for analysis. **A)** Optimal and **B)** suboptimal shooting form.

A 2D array containing coordinates for all videos was passed to the neural network model. Each row of the 2D array represented a video. For each row, the coordinates of all body joints of the first frame were followed by the coordinates of the second frame, and so on. Thus, each array row was built by concatenating each video frame in serial order. Note that the coordinates were extracted after applying a bounding box to the input video.

Model 2 (MLP with body joint angles)

Model 2 used the formula below to compute the joint angle from 3 points.

$$Joint\ Angle\ \emptyset = \tan^{-1}\frac{y_3 - y_2}{x_3 - x_2} - \tan^{-1}\frac{y_1 - y_2}{x_1 - x_2}$$

where *(x1, y1), (x2, y2), (x3, y3)* are coordinates of 3 points in a coordinate space. The 3 points were obtained using the Mediapipe Pose landmark model of a human body **(Table 3)**. Like the joint coordinate model, these angles were passed to the neural network model using a 2D array. Model 2 was tuned using hyperparameters and ran through multiple epochs to fully train the model **(Table 2)**.

Model 3 (RNN with body joint Cartesian coordinates)

For RNN, we used the LSTM model from the Tensorflow library (20). LSTM iterates through input sequences in the temporal dimension to analyze specific patterns while combating the challenge of vanishing and exploding gradients posed by RNNs.

Cartesian coordinates of a player's body joints (the same as the ones used by Model 1) were collected per video frame and provided as input to the LSTM model. Each time step in the LSTM model corresponded with a frame of a basketball shot sequence. Thus, the LSTM model was able to capture spatial and temporal information.

https://doi.org/10.59720/24-046

Model 4 (3D CNN with raw video) and Model 5 (3D CNN with cropped video)

Model 4 and Model 5 fall into the category of video-based AI models. Both used 3D CNN to analyze spatial and temporal features of a video, which is crucial for analyzing players' movement and positions over multiple frames. For CNN to process and extract crucial features, we had to convert each frame to a numerical format. To achieve this, we used Tensorflow to convert each pixel from three color channels to a floating-point numerical number.

The 3D CNN models used 16 filters with a kernel size of 7 x 7. Residual blocks were also used to improve each model's training accuracy. Often, a model's accuracy degrades due to an increase in the network's depth, which can be resolved by residual blocks that skip connections between layers, preventing the occurrence of exploding or vanishing gradients. This allows the model to learn features better and stops the degradation of the model's accuracy.

Each input video to the CNN models was resized to 100x100 pixels using the format_frames function in Tensorflow. The format_frames function resizes the video to the requested output size while keeping the aspect ratio constant by adding padding. This contributed to RAM efficiency and the model's speed. Model 4 used the raw video footage from the dataset as input, while Model 5 used cropped videos obtained using a bounding box around the basketball player.

Received: January 31, 2024
Accepted: July 8, 2024
Published: May 20, 2025

REFERENCES

1. Malone, Laurie A., et al. "Shooting mechanics related to player classification and free throw success in wheelchair basketball." *Journal of Rehabilitation Research and Development*, vol. 39, no. 6, Dec. 2002, pp. 701–10.
2. Wissel Hal. "Step 4 Shooting." *Basketball: Steps to Success*, 3rd ed., McGraw-Hill Education, 2004.
3. Delextrat, Anne, and A. Martinez. "Small-sided game training improves aerobic capacity and technical skills in basketball players." *International Journal of Sports Medicine*, vol. 35, no. 5, May 2014, pp. 385-91, https://doi.org/10.1055/s-0033-1349107.
4. Naik, Banoth T., et al. "A Comprehensive Review of Computer Vision in Sports: Open Issues, Future Trends and Research Directions." *Applied Sciences*, vol. 12, no. 9, Apr. 2022, pp. 4429-77, https://doi.org/10.3390/app12094429.
5. Zhang, Yu, et al. "Action Recognition in Still Images with Minimum Annotation Efforts." *IEEE Transactions on Image Processing*, vol. 25, no. 11, Nov. 2016, pp. 5479-90, https://doi.org/10.1109/TIP.2016.2605305.
6. Yan, Wenlin, et al. "A Review of Basketball Shooting Analysis Based on Artificial Intelligence." *IEEE Access*, vol. 11, Aug. 2023, pp. 87344-65, https://doi.org/10.1109/ACCESS.2023.3304631.
7. Liu, Riu, et al. "Recognition of Basketball Player's Shooting Action Based on the Convolutional Neural Network." *Scientific Programming*, vol. 2021, no. 3045418, Jun. 2021, pp. 1-8, https://doi.org/10.1155/2021/3045418.
8. Yang, Qingyao, et al. "Automatic Analysis of Basketball Shooting Based on Machine Learning." *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 1233-38, https://doi.org/10.1109/ICAICA52286.2021.9498159.
9. Cheng, Yao, et al. "Artificial Intelligence Technology in Basketball Training Action Recognition." *Front Neurorobot*, vol. 16, no. 819784, Jun. 2022, https://doi.org/10.3389/fnbot.2022.819784.
10. Ji, Rong, "Research on Basketball Shooting Action Based on Image Feature Extraction and Machine Learning." *IEEE Access*, vol. 8, Aug. 2020, pp. 138743-51, https://doi.org/10.1109/ACCESS.2020.3012456.
11. Li, Bin, and Xinyang Xu. "Application of Artificial Intelligence in Basketball Sport." *Journal of Education, Health and Sport*, vol. 11, no. 7, Jul. 2021, pp. 54-67, https://doi.org/10.12775/JEHS.2021.11.07.005.
12. Simonyan, Karen, and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos." *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 1, Dec. 2014, pp. 568–76, https://doi.org/10.48550/arXiv.1406.2199.
13. Wang, Limin, et al. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition." *European Conference on Computer Vision*, Oct. 2016, https://doi.org/10.1007/978-3-319-46484-8_2.
14. Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, Jan. 2013, pp. 221-31, https://doi.org/10.1109/TPAMI.2012.59.
15. Ihianle, Isibor K., et al. "A Deep Learning Approach for Human Activities Recognition From Multimodal Sensing Devices." *IEEE Access*, vol. 8, 2020, pp. 179028-38, https://doi.org/10.1109/ACCESS.2020.3027979.
16. University of Central Florida, "UCF 101 – Action Recognition Data Set." *UCF, Center for Research in Computer Vision*, www.crcv.ucf.edu/data/UCF101.php. Accessed 27 Oct. 2024.
17. Mandery, Christian, et al. "Dimensionality reduction for whole-body human motion recognition," 19th International Conference on Information Fusion (FUSION), Jul. 2016, pp. 355-362.
18. Guangle, Yao, et al. "A review of Convolutional-Neural-Network-based action recognition." *Pattern Recognition Letters*, vol. 118, Feb. 2019, pp. 14-22, https://doi.org/10.1016/j.patrec.2018.05.018.
19. "Mediapipe libraries.", *Google AI Edge MediaPipe Solutions*, github.com/google-ai-edge/mediapipe.
20. "TensorFlow LSTM model." *TensorFlow.org v2.16.1 API Documentation*, www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM.
21. "Pose Landmark Model." *MediaPipe Pose*, github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose.md.