Creating a Machine Learning-based Dance Move Correction System and Improving the Accuracy of the System

Nidhi Lawange

Received September 03, 2023
Accepted December 10, 2023
Electronic access December 31, 2023

Dance is a sport that typically requires the presence of a dance teacher to correct a dancer's technique. The COVID-19 pandemic challenged the effectiveness of remote dance classes because teachers could not see the precise leg movements due to the limited quality of video through an online platform. Similarly, it took a lot of work for the students to follow the exact leg movements the teacher was trying to teach, especially in a group setup. Consequently, all group dancers could not synchronize the specific sequence of dance moves within a given time interval. Furthermore, in many remote locations of the world, young dance enthusiasts do not have access to dance teachers or dance studios. This granted a wonderful opportunity to create an automatic self-correction system of dance moves for dancers using machine learning (artificial intelligence neural network models) that does not require the physical presence of a dance teacher. In this paper, I experimented on a dance move correction system that I built. This system consists of a Convolution Neural Network and an AI Human Pose Detection Model called MediaPipe. CNN is used for classifying the dance move while MediaPipe is used to get coordinates of landmarks on a dancer's body. Based on specific angle characterizations between landmark segments, the system outputs a correction message to the dancer if the angle between the dancer's feet does not correspond to the proper angle between the legs for that predicted dance move. The hyperparameters of the CNN model were tuned by filter size, number of filters and no. of convolution layers, and number of pooling layers that helped customize the model. The training data set we created for the 4 different dance moves1, 2 was modified to include a variety of possible dance move orientations. Before these experiments on the model, the validation accuracy of the CNN was 72% and the placement of the landmarks by MediaPipe was accurate 97.2% of the time. Thus, the overall accuracy of the dance move correction system was 69% (0.72 * 0.972 = 0.69). With the aforementioned experiments performed on the dance correction system, there was an improvement in the overall system accuracy to 80% validation accuracy. Still, the system maintains its ability to be trained for any general dance move. This research is unique in its way by isolating dance classification from dance correction. By first doing dance classification, the correction methods can be applied uniquely for the classified dance move. So, the methodology can be generalized to any dance move as long as the classification accuracy is at least 80%, and the subsequent dance correction algorithm can be tuned for that specific dance move.

Introduction

The dance move correction system is divided into two parts: a) Identifying a given test image belonging to one specific dance pose among many trained dance poses. Convolution Neural Network is used for identification/classification of images. b) Based on the identified dance move, apply the corresponding test matrix, which includes the angles between the legs and their segments. The joints or landmarks of the legs need to be identified to find the segments of the legs. The three-dimensional pose detection model, such as BlazePose MediaPipe, is used to identify the landmarks on the body. Other models, such as Posenet, identify only 17 landmarks. So, the Mediapipe 3D model was chosen because it produces 33 landmarks with less than 20 ns latency, which is also suitable for analyzing videos.

The ability of a system to correct a dancer's technique by using machine learning and Artificial Intelligence with an overall accuracy of at least 80% will allow this dance correction system to truly aid many dancers all over the world within the comfort of their own home and in remote places where there is rarely access to dance teachers.

Literature Review

Several research papers were identified for the literature review that address dance classification and correction. Correction of Chinese Dance Training Movements Based on Digital Feature Recognition Technology [1] focuses on a different approach in image classification. It relies on the dance move in which the body forms a B-Spline polygon. This B-spline polygon that is formed through the connection of body segments in a given image is mapped using the B-Spline equation on a given image, and it tracks the changes at each node. It uses fast-advance algorithms in ray tracing calculations. Although this paper explores feature

extraction that uniquely identifies the spline curve in a Chinese dance move, there are certain constraints on the paper's idea. The methods suggested by the paper to correct Chinese dance moves is limited to only one particular dance move. The system described in the paper also does not give specific correction messages to dancers about how to improve their technique. Lastly, it is very possible that the paper's proposed methods are overfitting the model; overfitting could be occurring when the system compares photos of dancers doing a dance move directly to the training data for that dance move using the B-spline algorithm. Then, through this algorithmic comparison, the correctness or incorrectness of a dance move is identified.

Sports Dance Intelligent Training Correction System based on Multimedia Image Action Real-Time Acquisition Algorithm[2] uses the method that performs overlapping blocks on a marked image, calculates the color features from marked blocks, and uses different weighing coefficients to then complete the matching image color markings in multi-media vision. It uses common digital image processing techniques such as image filtering etc. but provides poor matching stability. The technique proposed in this paper does general human motion detection but does not give any specific correction messages to correct a specific move while my technique does.

The unique part of this research is that it focuses on separating the identification of dance moves and the actual correction of dance moves (2 different steps) unlike the aforementioned dance correction systems and the other dance correction systems that exist. Other existing dance move correction systems incorporate the correction of the moves within the classification of the moves. The unique aspect of this research is accomplished by implementing a convolution neural network to identify specific dance moves while using a pose detection model to determine the landmarks on the human body and then using particular angle criteria matched to already identified dance move to check that move's correctness. That way, you can train a model for a variety of dance moves or human poses, and still, each can be corrected by applying specific rules using angles between legs, etc., to correct the dance moves. These specific rules for the angles depend on which dance move has been predicted in the first step. It allows for the expansion of this dance correction system to practically any type of human body movement and its associated correction if the rules of correction are clearly defined.

Results

After following the steps which will be expanded on later in the Methods section, the CNN model's training accuracy as well as validation accuracy reached 83%. Both training loss and validation loss were recorded to be mostly continually decreasing path as can be seen in Figure 1 below. The accuracy of the human pose detection model combined with the accuracy of the correction criteria message was 97.2%. This 97.2% accuracy consists of both the processing of the previously classified image through the human pose detection model MediaPipe and the extracted angle and correction message based on the angle test criteria. So, the overall accuracy of the correction system was measured at 80% (0.83 x 0.972 = 0.80) which fulfills the original research goal of achieving the overall accuracy of the correction system to be at least 80%.

Table: 1 below shows the result from the CNN model training with layer and parameters specified that led to the training and validation accuracy and training and validation loss displayed in Figure 1. Model: "sequential_10"

Discussion/Analysis of Results

In this paper, a system for dance move correction was built that consisted of a machine learning model, specifically a Convolutional Neural Network, and a Human Pose Detection Model, specifically MediaPipe. It was found that the system's accuracy increased by ensuring no overlapping images of dancers in the training data[3,4] set and several unique images of dancers doing the same dance move. The data was collected from images of professional dance poses from Shutter stock and Adobe stock websites, containing about 1,000 images labelled in 4 categories of dance moves: Arabesque, Passe, Leap, Sous sous.

When data augmentation was implemented on the data set, the validation accuracy decreased Some of the dance moves, such as Arabesque, were identified as leaps by the CNN model since the Arabesque image was oriented in several different directions with data augmentation. So, instead, we used images with many different orientations that would keep that dance move the same valid dance move in the image for training purposes; for ex: some images with an Arabesque with the left foot on the ground and right foot lifted and some other images with the right foot on the ground and the left foot lifted. With this approach, validation accuracy improved. As a future extension to the project, we will get an expanded set of varying image data sets for training to correct the issues with data augmentation.

There was a general trend of the decreasing activation value. Increasing the number of filters could have given narrower and deeper image classification by the CNN. This could have helped improve accuracy.

The fact that the performance of the system improved with the reduction in activation size showed that lowering the activation size while increasing number of filters when moving from one Conv2D layer to the next layer helped capture a global, high-level representative information of the image, thus increasing validation accuracy.

There could have been room for error in how the human pose detection model placed landmarks. After the angle correction criteria was modified to accommodate for these errors, the accuracy of the correction message output increased. For

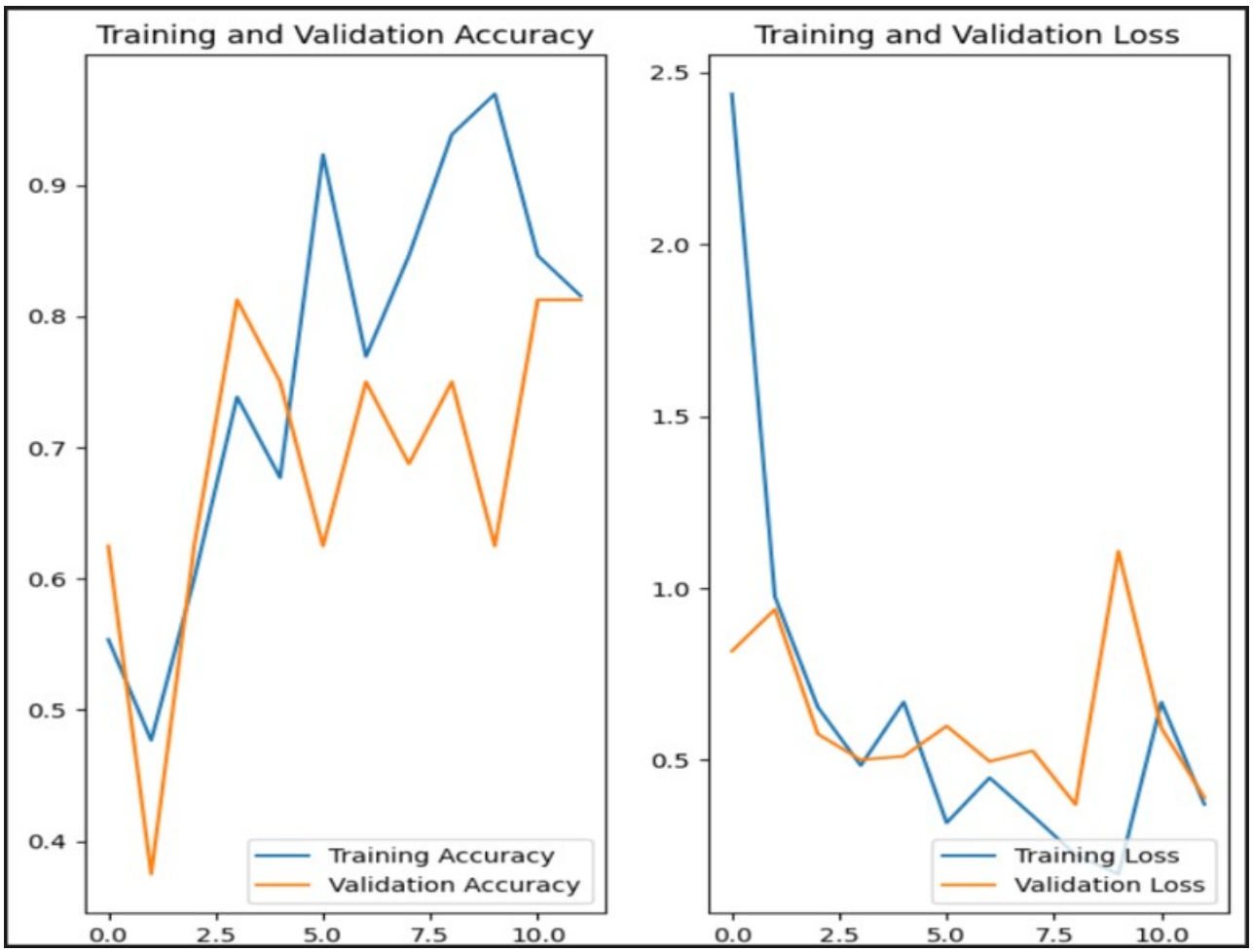

Fig. 1 Training and Validation Accuracy & Training and Validation Loss Graphs

example: Increasing the range of the correct angle to identify an Arabesque from only 90 degrees to any value between 90 and 125 degrees helped improve the accuracy with which dancers' images were correctly identified as an Arabesque. This showed that modifying the angle correction criteria was also helpful for improving the validation accuracy of the system. The Figure 2 overall flowchart improving the validation accuracy of the overall system is shown below:

By isolating dance classification from the actual dance technique correction algorithm, this system can be used for any human pose including any dance as long as the corresponding dance move is well tuned on a convolution neural network with at least overall 80% accuracy, and a human pose is detected using a corresponding angle criterion between different landmarks provided by a human pose detection model like MediaPipe.

This dance correction system with overall 80% validation accuracy, tested in Jensen Performing Arts dance studio, Milpitas, California, was found to have the same ability to correct dance moves as that of the live dance teacher.

Methods

The Figure 3 flowchart of the overall dance correction system is shown below.

The CNN model provides pixel by pixel comparison and provides the percentage confidence that an image belongs to a specific dance move. That helps to apply specific rules of angles between legs, for instance, for that identified dance move. The CNN does not provide coordinates of body joints. But once the dance move is recognized with 83% confidence from the CNN, the image is run through the human pose detection model, Mediapipe[5]. Mediapipe provides coordinates of joints on the body. Before choosing the Mediapipe model, I evaluated several human pose models such as Single Pose vs Multi Pose and selected the single pose model for my research purpose and

Table 1 Layer Information & Parameters of CNN Model

Layer (type)	Output Shape	Param #
rescaling_20 (Rescaling)	(None, 180, 180, 3)	0
conv2d_36 (Conv2D)	(None, 180, 180, 16)	448
conv2d_37 (Conv2D)	(None, 180, 180, 16)	2320
max_pooling2d_22 (MaxPoolin g2D)	(None, 90, 90, 16)	0
conv2d_38 (Conv2D)	(None, 88, 88, 32)	4640
conv2d_39 (Conv2D)	(None, 86, 86, 32)	9248
max_pooling2d_23 (MaxPoolin g2D)	(None, 43, 43, 32)	0
dropout_10 (Dropout)	(None, 43, 43, 32)	0
flatten_10 (Flatten)	(None, 59168)	0
dense_28 (Dense)	(None, 512)	30294528

Total params	30,312,210
Trainable params	30,312,210
Non-trainable params	0

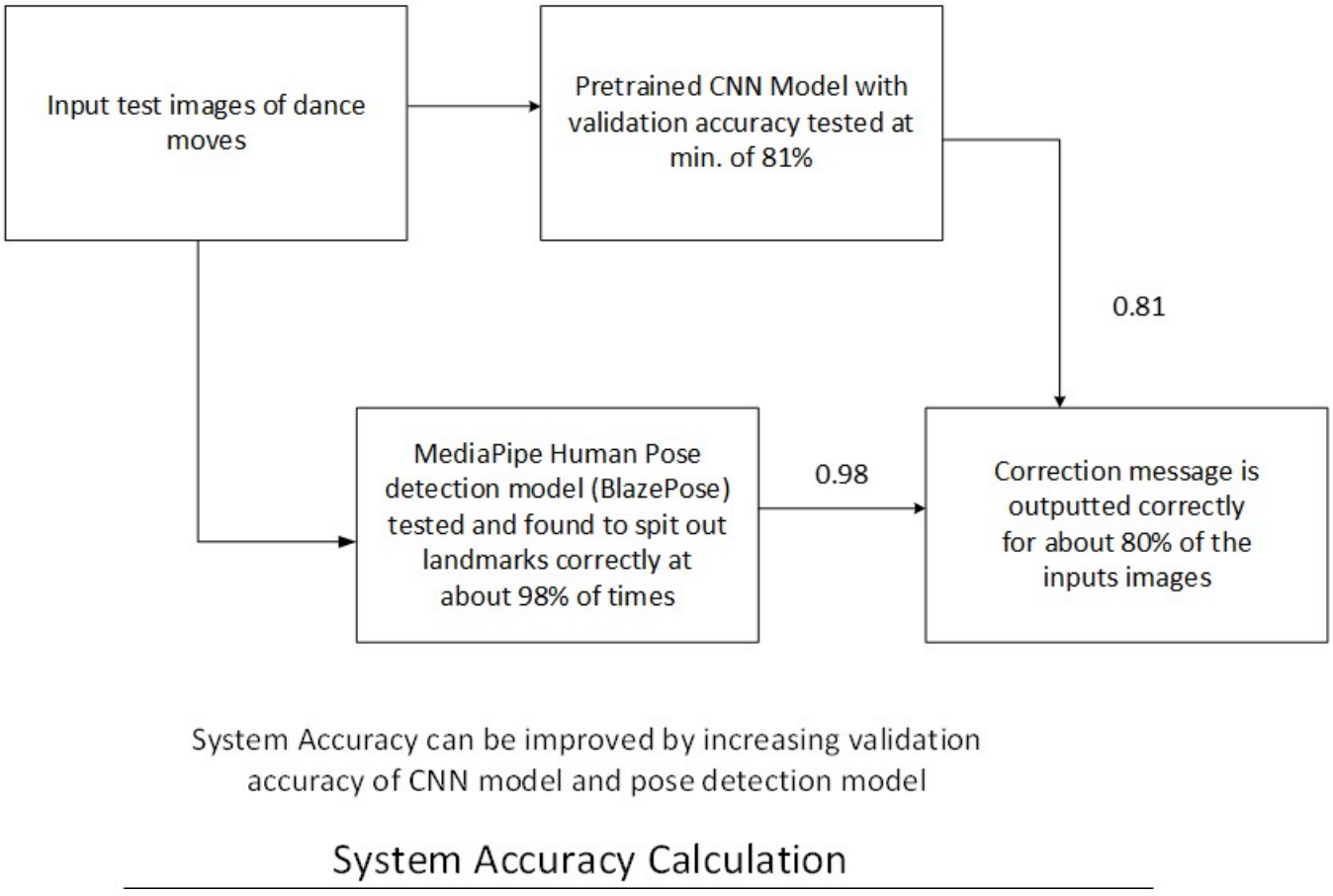

Fig. 2 Dance Move System Accuracy Calculation, Flowchart

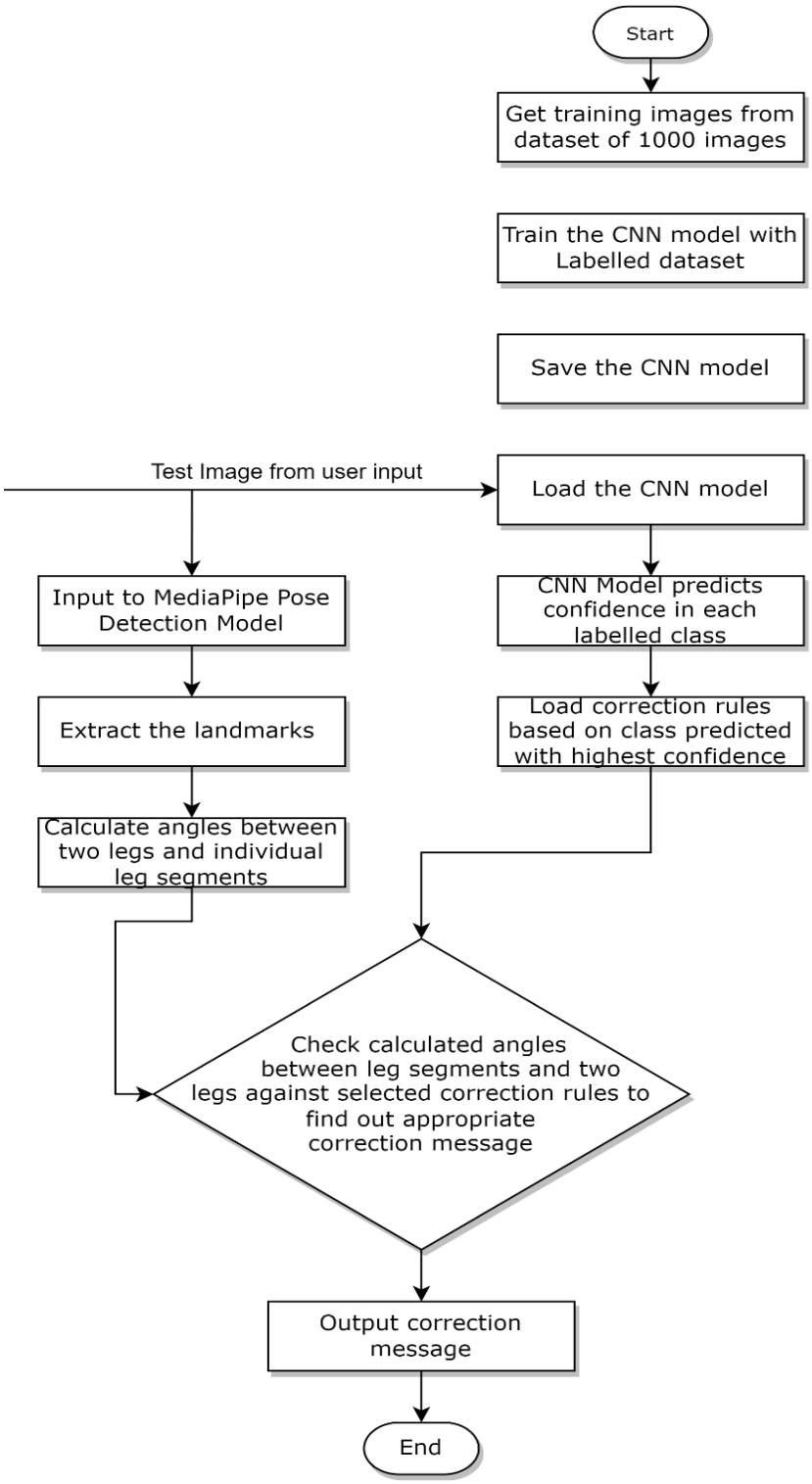

Fig. 3 Dance Move Correction System Flowchart

set a goal of focusing on improving the accuracy of any dance move correction system. I chose this 3D model to interpret the dance move correctly in its entirety as dance moves cover all 3 dimensions. I evaluated the MediaPipe BlazePose GHUM 3D (33 landmarks) and Posenet (17 landmarks) and finally chose MediaPipe model because it provides a 3-dimensional map of the body with the largest set of 33 joints with a tracking speed of about 18 fps5.

Based on the Figure 3 flowchart of the overall system, the accuracy mainly depends on two points:

1. CNN model validation accuracy for a specific dance move
2. Correction criteria applied for an extracted feature such as the angle between the legs calculated using the landmarks extracted from the human pose detection model.

steps were implemented:

Changing hyperparameters and Parameter Initializers: All input images used for training or validation were rescaled to a common size to ensure that all images that are input into the system are standardized so that the convolution results are meaningful.

Kernel/Filter Size: Filters are two-dimensional matrices with values mentioned as derived weights (using back propagation/gradient descent) that scan the input image during convolution[6]. Smaller size filters gain more local information while bigger filters provide high-level representative information. In general, filters are used in convolution to detect edges in the image. We chose a smaller filter size for the first Conv2D layer.

Padding: Padding is the process of adding zeros onto a previously processed image (either from a previous convolution layer or previous pooling layer)6. It keeps the spatial size of the image the same which can retain more information at the borders. If the "same" option is chosen for the padding layer, it will keep the size of the input image the same as the output image while if the "valid" option is chosen, the image shrinks to **Eq. 1** $ceil(\frac{n-f+1}{s})$[7] where 'n' is input dimension, 'f' is filter size and 's' is stride length. We used stride length = 2, which reduced the activation size when we move from one conv2D layer to the next Conv2D layer.

Stride: Stride is used to skip pixels both in the horizontal and vertical axis direction to reduce the size of the input image without sacrificing significant accuracy6. It is using the same equation as mentioned before to find out the output image size: $ceil(\frac{n-f+1}{s})$[7]

Other Parameters: A rescaling ratio of 1/255 and input shape to 180 x 180 x 3 was chosen for the image. A CNN architecture in which there were 2 Convolution Layers followed by 1 MaxPooling Layer followed by 2 Convolution Layers followed by 1 MaxPooling Layer, consisting of 6 layers, was used.

A smaller filter size was used to begin with in order to collect the maximum local information possible from the image. Gradually, the filter size was either kept the same or increased in order transition from a wider and shallower feature space to a narrower and deeper feature space of the image.

The dropout ratio was configured to 0.25 to avoid overfitting.

The kernel initializers were configured to "random_normal" and bias initializers to "normal" for quick convergence of the model.

The linear activation function (relu) was used for the first few layers and the "softmax" activation function (non-linear) was used at the output layer.

The hyperparameters were chosen in such a way that there is a general trend of the activation value decreasing without resulting in a negative activation value at the end. The Table 2 shows the exact configuration and the calculation of activation values.

Calculation of activation shape and value after each convolution layer or Pooling layer where padding is valid is done with Eq 2. Use Eq. 1 to form Eq. 2 Eq 2:

$(ceil(\frac{N-f+1}{s}), (ceil(\frac{N-f+1}{s}), \text{Number of filters}) \rightarrow (ceil(\frac{N-f+1}{s}) \times (ceil(\frac{N-f+1}{s}) \times \text{Number of filters}$

Calculation of activation shape and value after each convolution layer or pooling layer where padding is same:

$(N, N, \text{Number of filters}) \rightarrow N \times N \times \text{Number of filters}$

Fixing the human pose detection model to extract specific features such as the angle between body segments:

As shown in Table 3 below, these are the specific criteria to identify a specific move in this case among different ballet poses:

The criteria are chosen in such a way that even though there is some error in placing landmarks on the right locations due to inadequate accuracy of the human pose detection model, MediaPipe, the criteria would still correctly validate the dance pose.

article amsmath

The angle calculation for the angle test criteria for each dance move is done as shown below: First, the intersection point between two legs is found:

As shown above in Figure 4, Line 1 consists of the right hip and right ankle.

For example: Right hip is located at (x_1, y_1) and right ankle is located at (x_2, y_2) Eq. 3: The equation of right leg (Line 1): $(y - y_1) \cdot (x_2 - x_1) = (y_2 - y_1) \cdot (x - x_1)$

As shown above in Figure 4, Line 2 consists of the left hip and left ankle. For example: Left hip is located at (x_3, y_3) and Left ankle is located at (x_4, y_4) Eq. 4: The equation of left leg (Line 2): $y - y_3 = \frac{y_4 - y_3}{x_4 - x_3} \cdot (x - x_3)$

Implemented the function *findIntersection()* function using the above information.

Angle between these two intersecting segments is found as follows:

Table 2 The Effect of Number of Filters, Filter Size on Activation and Activation Size

Layer	Number of filters	Padding	Activation Shape	Activation Size
Input Image	-	-	(180,180,3)	97200 (180 x 180 x 3 = 97200)
Conv2d (f=3, s=1)	16	same	(180,180,16)	518400 (180 x 180 x 16 = 518400)
Conv2d (f=3, s=1)	16	same	(180,180,16)	518400 (180 x 180 x 16 = 518400)
MaxPool (f=2, s=2)	16	valid	(90,90,16)	129600 (90 x 90 x 16 = 129600)
Conv2d (f=3, s=1)	32	valid	(88,88,32)	247808 (88 x 88 x 32 = 247808)
Conv2d (f=3, s=1)	32	valid	(86,86,32)	236672 (86 x 86 x 32 = 236672)
MaxPool (f=2, s=2)	32	valid	(43,43,32)	59168 (43 x 43 x 32 = 59168)
Dropout (0.25)	-	-	(43,43,32)	59168 (43 x 43 x 32 = 59168)
Flatten	-	-	(59168, 1)	59168
Dense	-	-	(64,1)	64
Dense	-	-	(128, 1)	128

f = filter size= f x f
s = strides

Table 3 Criteria for each Dance Move & the Correction Message based on Criteria Match

Dance Move	Criteria	If Criteria Matches	If Criteria Does Not Match
Arabesque	90 <= angle between left and right leg <= 125	"Beautiful Arabesque"	"Lift Your Leg Higher"
Passe	(Angle between right hip -> right knee -> right ankle > = 160) AND (Angle between left hip -> left knee -> left ankle < = 60)	"Beautiful Passe"	"Right leg needs to be straight" or "Left leg needs to bend more"
	(Angle between left hip -> left knee -> left ankle > = 160) AND (Angle between right hip -> right knee -> right ankle < = 60)	"Beautiful Passe"	"Left leg needs to be straight" or "Right leg needs to bend more"
Leap	Angle between left and right leg >= 170	"Beautiful Leap"	"Stretch your legs, point your feet. Push off the back foot with more power".
Sous sous	Angle between left and right leg <= 40	"Beautiful sous sous"	"Bring your legs together"

Find angle between two intersecting line segments; [0][1] indicates = [x][y] co-ordinates

3 landmarks: $a[0][1]$, $b[0][1]$, $c[0][1]$

Eq. 5: slope of segment ab $m_1 = \tan(\theta_1) = \frac{b[1]-a[1]}{b[0]-a[0]}$ where $\theta_1 = \arctan\left(\frac{b[1]-a[1]}{b[0]-a[0]}\right)$

Eq. 6: slope of segment bc $m_2 = \tan(\theta_2) = \frac{c[1]-b[1]}{c[0]-b[0]}$ where $\theta_2 = \arctan\left(\frac{c[1]-b[1]}{c[0]-b[0]}\right)$

Difference between these two angles adjusted for a negative angle is implemented in the *getAngle()* function, which returns the angle.

Conclusion

The objective of this research was how can the overall validation accuracy of the machine learning-based self-correction dance system be improved to at least 80% without sacrificing its ability to adapt to a variety of dance moves. The research focused on both improving the validation accuracy of the convolution neural network and correction system that implements the rules

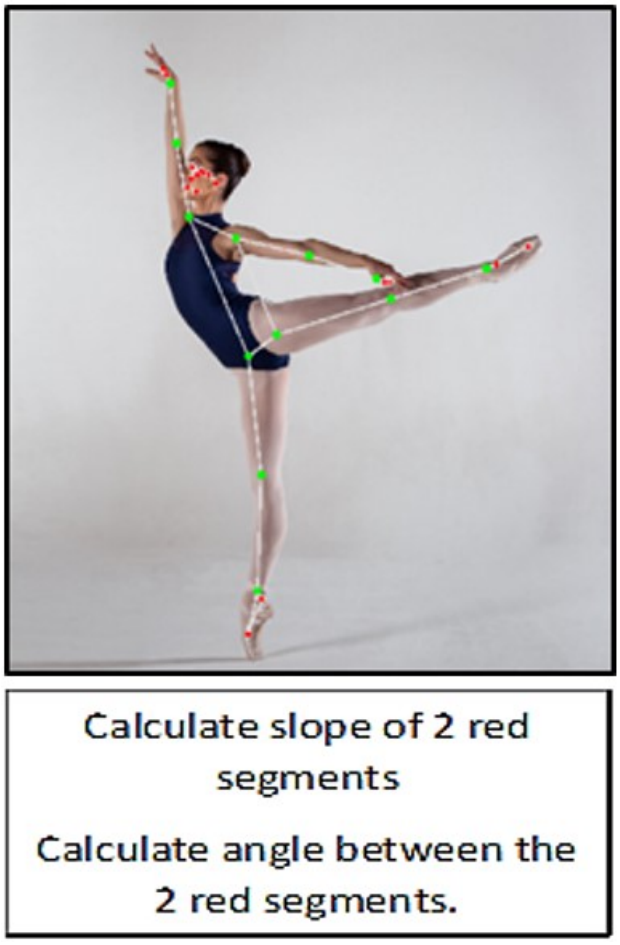

Fig. 4 Dancer with Landmarks, Line Segments, and Angle annotated.

for specific dance moves.

Specifically, while choosing hyperparameters of the CNN model, the smallest filter size for the first Conv2D layer (2 x 2) was used to capture the maximum local details. Next, increasing the filter size or maintaining the same filter size for the next Conv2D layer combination while increasing the number of filters to capture a global and high-level representation of the given image helped to improve validation accuracy. By using pooling layers with stride length of more than 1, the activation size could be reduced. This helped to improve the convergence speed. The accuracy of the CNN classification of the dance image/video increased to 83% and the human pose detection model and the associated angle correction message accuracy improved to 97.2%.

Human pose detection model BlazePose MediaPipe 3D model is a pre-trained model from Google with PSK accuracy (Percentage of Correct Key Points) equal to 97.2%[8], and if the landmarks determined are correct, then with the correction algorithm accuracy of 99.9% will result in overall accuracy of 97.2% for pose detection model and subsequent correction. After following each step mentioned in "Methods," the overall correction system accuracy improved to 80% (0.83 * 0.972 = 0.80) because first step accuracy is 83% while second step accuracy is 97.2%. Thus, the objective of the research was achieved.

This research focuses on decoupling the identification of dance moves to the correction of dance moves by implementing a convolution neural network to identify specific dance moves while using a pose detection model to determine the landmarks on the human body and then using particular angle criteria for already identified dance move to check its correctness. That way, one can train a model to a variety of dance moves or human poses, and still, each can be corrected by applying specific rules using angles between legs, etc., to correct the dance moves. It allows the expansion of this dance correction system to practically any type of human body movement and its associated correction if the rules of correction are clearly defined.

Potential challenges of implementing this dance move correction system include that the system requires a clear and high-quality image or video to process in order to correctly place the landmarks on the dancer's body. If a given image or video is low-quality or taken in darker surroundings, it would be difficult for the dance move to be seen, classified, and corrected. An ethical consideration of this system is that this dance move system requires pictures and videos of actual people dancing in a private setting, and these images or videos will be saved on a computer or in the cloud. Receiving permission from the dancers to store those images and ensuring that those stored images are not misused in the present or future are also ethical considerations to be made. In the case of incorporating this dance move correction system in real-world dance education, certain policies should be created that consider the privacy of the dancer and assure that the images are used only for the purpose of correction.

Limitations & Future Work

Although the current dance correction system is trained for only 4 dance poses of Ballet, it is not a limitation of the correction system. If the CNN model is trained for a variety of different dances or human pose images and the associated correction message angle criteria is written for a specific classified dance or human pose, this correction system can be used for any dance move or human pose. The dance correction system currently works on a single dancer's image and can be further expanded to accommodate multiple dancers in a group setting.

As further research, we will evaluate the possibility of expanding the set of images used for training to accommodate data augmentation. In implementing such a dance correction system where individual dancers' pose is corrected in the real world, thorough testing in different background settings is required.

The ability of this dance move correction system to have a high accuracy while correcting a dancer's technique to help them improve enables this dance move correction system to be installed on a Raspberry Pi board with a Raspberry Pi camera that would take pictures or video of the dancer.

Acknowledgments

Malak Sadek

References

1 L. Zhang, *Mathematical Problems in Engineering*, 1–11.

2 L. Pan, 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC.

3 *Ballet images – browse 289,990 stock photos, vectors, and video*, `https://stock.adobe.com/search?k=ballet`, Adobe Stock. (n.d.).

4 `https://www.istockphoto.com/search/2/image?phrase=ballet`, n.d.). Dancers in White Tutu synchronized dancing stock photo. iStock.

5 Google, *Pose landmark detection guide — mediapipe — google for developers. Google*, `https://developers.google.com/mediapipe/solutions/vision/pose_landmarker/`.

6 R. Pramoditha, *Medium*.

7 S. Ramesh, *Medium*.

8 *GitHub, github.com/google/mediapipe/blob/master/docs/solutions/pose.md. Accessed*.