

Kathak Dance Step Mapping Using LSTM

Sehar Randhawa[1]

Grade 12, Indian School Al Ghubra, Muscat
Oman

Reetu Jain [2]

Chief-Mentor and Founder of On My Own Technology
Private Limited Mumbai,
reetu.jain@onmyowntechnology.com

Abstract: *We have proposed an interesting application of human action recognition to dance style, more specifically to the Indian classical dance form 'Kathak'. The paper focuses on developing an unbiased system that can rate the steps performed by a dancer without the help of human intervention. Kathak is a field that has not been researched extensively from an Image processing and machine learning standpoint. This makes the availability of proper training data a major issue. The first half of the paper focuses on collecting data in video form, applying appropriate pre-processing, and then making this data available publicly. The second half of the paper focuses on training an LSTM model using the collected data to be able to perform multi-class classification on the different dance steps with 100% accuracy and then be able to use the confidence metric of the prediction to score the steps that were performed. The scoring done by the model using this methodology is then compared to the scoring done by a professional kathak dancer to validate the model's performance.*

Keywords: Kathak, Machine learning, Pose estimation, Motion Analysis, Classical Dance forms

1. Introduction

Kathak is one of India's most respected traditional dance styles. The sole dance from the north of India, Kathak is one of the eight Indian classical dance styles and is a work of art with an enigmatic allure, captivating footwork, and elegant motions. Kathak started in the villages of Northern India, where the locals recounted their life stories. These storytellers, known as Kathakkars, traveled from village to village and kingdom to kingdom, sharing their craft. These Kathakkars would periodically stop at temples in these locations to relax, and it was here that they began to enact stories from the great Indian epics, as well as stylize the art by adding a classical touch. With the arrival of the Mughals, who seized the temple riches, these Kathakkars were also brought into the Mughal courts. The emphasis on religion shifted as well, with Emperor appeasement taking priority. Kathak reflected on the changes that occurred with the foundation of the Mughal Empire. Manuscripts, music, jewelry, costumes, and dancing methods all evolved and changed rapidly. Urdu was included in the art form. However, the arrival of the British compelled the great dancers of the Mughal era to abandon their profession since dancing as an art form was prohibited by the authorities. Because there were no other options for safeguarding this unwritten and unrecorded art form, these artists began home tutoring by teaching their own family members in an effort to preserve the history of this old art form.

The Evaluation aspect of, how elegantly/completely a particular mudra or an action has been performed is solely done by an artist who has/had years of experience performing Kathak. The same is true when any other dance form is taken into consideration. Although inspection by human eyes can be considered the best way to evaluate something as complex as dance forms. We have suggested a Machine learning based more scientific approach for evaluating and rating dance moves.

Action recognition has been a field of active study in recent years with the emergence of pre-trained models allowing one to track the joints in the human body. Few methods explore the ways in which video data can be fed directly to an RNN model to be able to classify the action being done. But these models suffer from the quality and the low quantity of the data available. Modern methods for action recognition involve first training a model that can detect and provide the skeletal mapping of the person performing the action and then using the data from the skeletal mappings to detect the action being performed. Although this adds an extra layer of complexity, the output we get is way more accurate and reliable. A Few of the popular libraries that allow for easy joint/skeletal tracking involves

1. OpenPose
2. PoseDetection
3. DensePose
4. AlphaPose
5. MediaPose

MediaPipe is a collection of easy-to-use pre-trained models developed and maintained by Google. There are several models that allow users to access and use the model via simple-to-use APIs. Media Pose is one of the implementations that is provided by MediaPipe. It provides 33 landmarks. These landmarks can be later used to train an action recognition model. The model is very optimized and can process a video playing at approximately 30fps without any lag. This makes it ideal for live tracking of actions being performed. There have been several research works where the normalized data from the Media Pose model was used to train a second model that uses the angle data as the input and predicts the particular action being performed. Also, action recognition using RNN and LSTM methods has been used. This allows the model to predict the action based on the previous frames of input that are being fed to the model. The camera positioning, scale, and closeness to the camera are other factors that affect the affection of the action recognition using joint information. But the Media Pose model allows us to compensate for these changes using the API.

We have used a MediaPose-based model to extract the pose landmarks from a video of a particular kathak action. And then the data is normalized and scaled to be able to feed it to a model that can predict the scoring based on the previously stored action. Later the prediction done by the model and scoring done by a Kathak teacher was compared to comment on the performance of the model.

2. Literature Review

Most common hurdle for creating any proper ML model in this field is the non-availability of proper training data. But various approaches have been applied over years to tackle this problem. Soumitra Samanta et al.[1] apply image processing to classify classical dance. They had created their own data set for training and testing the model. YouTube videos were edited to be able to create the data set. Their novelty was to develop a new action detector specifically for Indian classical dance. Accuracy of 86.67% was achieved. They also tested their model against the KTH data set and they claim that the model performance is at par with the state-of-the-art technology available. Shailesh S et al.[2] talks about how image processing can be used to create intelligent information retrieval systems for Indian classical dance. The paper proposes various ways of preserving cultural heritage. Their focus is the preservation of classical dance. A video archive was created by recording trained dancers and by using the existing videos available online. They created a deep pose estimator coupled GRU model and found that their model outperformed various CNN and CNN-LSTM-based models even when the video quality was low. Nikita Jain et al.[3] has proposed a deep convolutional neural network based ResNet50 model. The model cannot directly process the input video instead a bit of image processing which includes thresholding operations has to be done for the model to perform accurately. They achieved an accuracy score of 91%. Vinay Kaushik et al.[4] discusses the complications involved in classifying Indian classical dance.

These complications are majorly due to the fact that the dance form is a complex combination of hand, body, and facial expressions. They have suggested a model that uses the joint relationship instead of the traditional ways in which the data from the landmark is directly used to train the model. The outcome is a classifier that can classify between 6 dance forms. They achieved a class accuracy of 87.93% for Kathak. The lowest class accuracy was for the Odissi dance form. The accuracy was about 63.76%. They suggest that the lower accuracy was due to the fact that there exists a lot of similarities between different dance forms in terms of steps being performed. Ashwini Dayanand Naik et al.[5] has proposed a 3D geometric data-based classification model as opposed to the standard 2D data-based model. They specify how conventional CNN-based models cannot work with non-euclidean data and hence a PointNet model was used to classify the 3D point cloud data into 5 dance forms which were Bharatanatyam, Odissi, Kathak, Kathakali, and Yakshagana. Authors argue that analyzing something as complex as dance forms should not be done with 2D data as it can cause the model to learn the complexity of the task being performed. Rajisha P et al.[6] have generated a hand gesture data set for kathak. The data set consisted of 24 hand mudras. An attempt was made to predict the mudras that were being performed by the artist.

3. Proposed Methodology

3.1. Data acquisition

Since there has been very little research related to Kathak and data science, there is no proper data set that has been already collected and published. So creating a data set was one of the challenges. Three Kathak dance steps were selected. There were no specific reasons for selecting these dance steps. They were just randomly picked for ease of data collection. 5 individuals performed the selected dance step for 10-20 repetitions in a single go. The continuous videos captured were later edited and made into a data set containing just a single repetition. Dance steps done by two highly qualified individuals were selected as the baseline for the dance steps that were going to be evaluated. This choice brings in a bit of human intervention which is not ideal provided that the entire idea behind the project is to make a model that is not biased by the human decision-making process. But choosing an ideal dance step allows us to set a baseline for the model for it to be able to judge the other videos in the training and the test dataset. Evaluation of each video was done by a professional Kathak dancer who scored the dance steps on a scale of 1-10 this data was later used to compare the model's performance.

3.2. Data Pre-processing

Using the collected videos to train the ML model would not be ideal. So pre-processing steps were applied to the collected data to ensure that the machine learning algorithm performs well. The recorded videos all had different frame rates. To solve this problem we select 30 equally spaced frames from each step being performed. This is a mandatory step as the LSTM model that is being used requires the shape of the training array to be constant across the window.

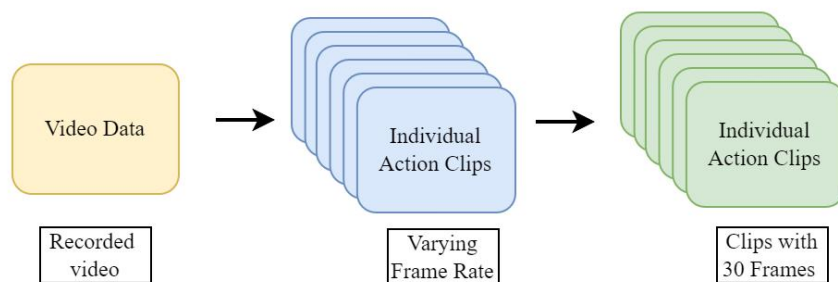


Fig 1. Pre-processing to limit the number of frames in each step

Mediapose library is used to extract the landmarks from each frame. Landmarks returned by the MediaPose library were normalized to the frame width and height. This is an issue as different videos in the data set have different frame widths and frame heights. Apart from this different dancers can have different heights and can be standing at varying distances from the camera. This causes the person to appear big or small causing the landmark position to be different. So a landmark normalization technique was used which compensates for the above-mentioned issue. Pose embedding provided by the MediaPose library was used as a starting point for feature engineering and the customization of the pose embedding was done to come up with appropriate features that were going to be used in training the ML model.

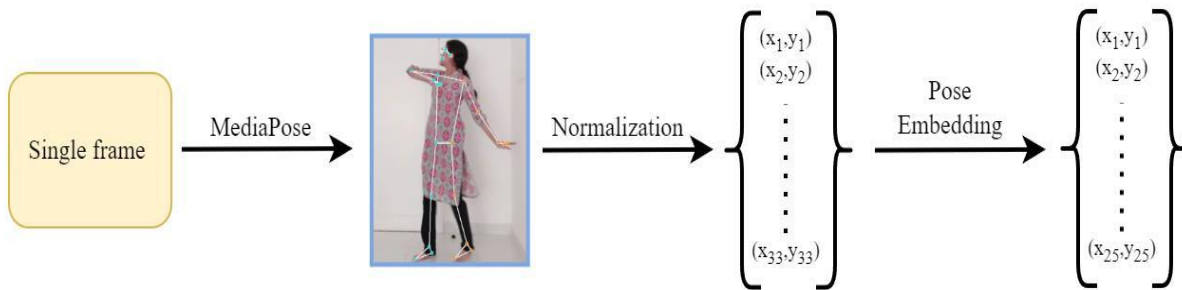


Fig 2. Pre-processing used to get the Normalized data after Pose Embedding

Pose embedding is a very simple technique by which we first normalize the size of the body by taking the distance between the two hip joints and then scale down all the other joints as per this value. All 33 joint coordinates are scaled with this method. Instead of using the normalized and the scaled joint coordinates directly, feature engineering was used to convert the 33 landmark coordinates into 25 distance metrics with both the x and y distance values. A 3 dimensional NumPy array was created for each pose from the video files. This array was created such that it can directly be used with the LSTM model which requires the data to be in this shape (*window, frames, features*).

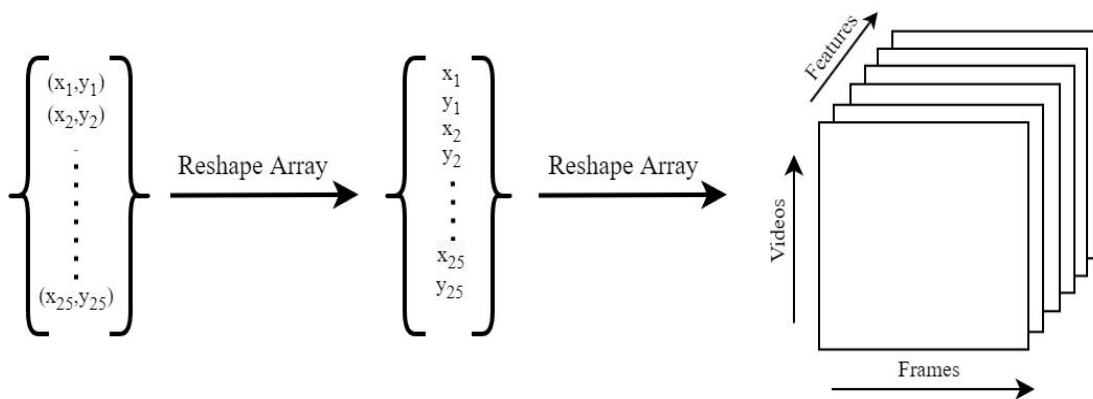


Fig 3. Reshaping image data for the LSTM model

Every window is a single row in the NumPy array containing data of a single action being performed. The 25 distance metrics which include the x and the y values were stored in a 2D NumPy array with shape (25,2). Since the LSTM model required the features to be a single-dimensional array the distance metric array was unwrapped and converted into a 1D array with shape (50,). The Final Numpy array for the 3 poses was converted into a .npy file and uploaded to GitHub. The link for the same is provided in the Data source section. The pre-processed train and test files have

been provided as .npy files. The number of rows in the NumPy array denotes the number of videos that were used to create the array. Every row is composed of 30 individual frames and every frame is composed of 50 distance features.

4. Machine learning model

Three Dance steps were selected during the dataset creation phase. Test and Train data sets (.npy files) for all three steps have been provided on the GitHub page. For training the model only 2 of these steps were selected as the amount of data available for the 3rd step was not adequate so this step was not included as it can introduce some bias during the training phase. The data that was selected for training the model was created by two separate Professional Kathak dancers. This was important as the trained model is also required to score the test videos later after the classification.

Long short-term memory (LSTM) was used to train the model. Input to the LSTM layer was the array created in the pre-processing step. LSTM requires the input array to have (*timestep, feature*) as the shape. Our data has 30 timesteps in each window and 50 features for every timestep. LSTM is similar to the RNN model where the model learns not just based on the data from the current time step but also retains the data pertaining to the previous timestep. This allows the model to learn complex actions that are being performed which span across multiple frames in a video. We have used the Keras sequential model in which the first layer is composed of the above-said LSTM model with an output space of 20. This is followed by a dropout layer with a 50% dropout rate. Dropout layers are important while training an LSTM model as they have a tendency to overfit the data. The dropout layer is followed by a dense layer with an output dimensionality of 20. The dense layer is a deeply connected neural network layer. This is again followed by a dropout layer with 50% dropout and this is connected to the final Dense layer which has 2 output dimensionality which corresponds to the number of poses being trained.

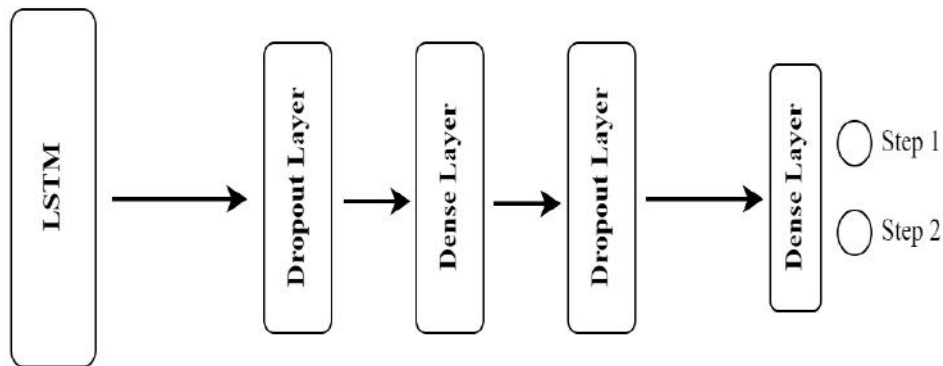


Fig 4. Sequential model created using Keras API's

Once the Sequential model was prepared it was trained on the training dataset. Adam's optimizer was used during the training process. Since the model does not need to learn the relationship between the different videos the training data set was shuffled before the model was trained. Input data containing the labels of the dance step was also converted to categorical as the LSTM model requires data in terms of 0s and 1's for it to learn better.

Once the model was trained and an acceptable accuracy was achieved a simple standard scaling-based approach was used to convert the confidence level of the LSTM classification model into scores that ranged from 1-10. The minimum and the maximum values from the confidence score that was achieved during the testing of the model were used to scale the confidence score and achieve the required scoring.

5. Test Results

Table 1. Comparison of Scorings done by a professional dancer vs. the scoring done by ML model

	Pose 1		Pose 2	
	Professional	ML model	Professional	ML model
1	6	7.2872	6	6.35729
2	6	8.1741	6	6.00324
3	7	7.6037	7	5.98732
4	6	7.7348	7	7.19581
5	6	6.048	7	6.45277
6	7	7.373	7	7.14884
7	6	6.0247	7	7.20689
8	6	6.2935	6	6.35943
9	7	7.8361	6	6.8144
10	8	7.9178	6	5.97475
11	6	6.2828	6	7.51892
12	6	5.5902	6	6.89343
13	6	6.3931	7	6.84384
14	8	5.71	7	7.18967
15	8	6.6804	7	7.33325
16	6	5.2809	7	6.07429
17	7	6.436	7	6.75163
18	7	6.9516	6	6.59994
19	7	6.7632	6	6.9832
20	7	7.494	6	8.0354

20 steps from each pose were selected randomly for testing. The model predicted the steps accurately as beginning to two separate classes. The model's confidence score for each prediction was then converted into the scores. They were recorded along with the scores that were provided to the same steps by a professional Kathak dancer.

6. Conclusion

Comparing the test results shows that the model's prediction and the score are very similar to the scores that were given to the individual steps by a professional trainer. For steps that were scored poorly by the professional dancer the ML model also scored them low. However, in all the instances when the professional dancer rated the steps above 7, the model rated them below 6. This may be due to the fact that the ML model can pick out minute changes that otherwise would not be noticeable by human eyes. The speed at which a particular step is being performed can affect the decision of a human. But the ML model can make out differences that exist even in fast performed steps causing the model to give a lower score to these steps.

6.1. Future Scope

Kathak as a dance form is heavily dependent on expressing feelings through expression. Every pose has a different expression that is performed by the artist. So detecting the expressions and predicting how accurately a particular action was performed during the dance can be incorporated into the model. This can be accomplished by using the Mediapipe Face mesh model. Although it might require more data to get trained properly as the number of landmarks available on the face is very large. But with proper training, we can pretty much predict the expression being imitated. We have used a base video of a dance step being performed to compare the other dance step videos that were performed

by the artist. This is not ideal as this brings in a human bias to the entire process. Instead, we can replace the base video with a different approach that uses the angles that are being formed during a particular action and only track the joint motion during the dance step. This can also be later modified to take into account mathematical equations and the resulting curves made by the equation. These equations need to be crafted properly. Once the equations of the motions are defined then we compare the curves made by the dance steps to the curves made by the equations that we defined earlier. This will give us a more mathematical and scientific approach to the Kathak dance style.

Acknowledgment

I would like to express my gratitude to the mentors of On My Own Technology Pvt. Ltd. for extending their help in carrying out this project. I would also like to express my gratitude to Ms. [Vishakha Apte](#) (Mentor, On My Own Technology Pvt. Ltd) and her colleagues at the dance center, for helping me collect the necessary videos required to generate the train and test data for training the machine learning model.

Data repository

All the Machine learning code and the data collected which is required to recreate the work done have been uploaded to the following <https://github.com/sehar4679/Kathak> git repository. The data folder has the test and the train data for all the 3 dance steps although only 2 dance steps have been used to train the model. The data is in the form of a .npy file containing a 3D NumPy array. The 'Kathak_model' folder has the trained model. The model is stored in the Keras SavedModel format.

References

1. Soumitra Samanta, Pulak Purkait, Bhabatosh Chanda. Indian Classical Dance classification by learning dance pose bases. *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*.
2. Shailesh S, JudyM.V. Understanding dance semantics using Spatio-temporal features coupled GRU networks. *Entertainment Computing Volume 42*, May 2022, 100484.
3. Nikita Jain, Vibhuti Bansal, Deepali Virmani, Vedika Gupta, Lorenzo and Laura Garcia. An Enhanced Deep Convolutional Neural Network for Classifying Indian Classical Dance Forms. *Human-Computer Interaction for Industrial Applications*, Applied sciences June 2021.
4. Vinay Kaushik, Prerana Mukherjee, and Brejesh Lall. Nrityantar: Pose oblivious Indian classic dance sequence classification system. *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP December 2018 Article No.: 65, Pages 1–7.
5. Ashwini Dayanand Naik and M. Supriya. Classification of Indian Classical Dance 3D Point Cloud Data Using Geometric Deep Learning. *Computational Vision and Bio-Inspired Computing* pp 81–93, June 2021.
6. Rajisha P, Jithendra K B. Classification Of Kathakali Hand Gestures Using Machine Learning And Deep Learning Techniques –A Review. *IJCRT Volume 10, Issue 3 March 2022*.