2022 International Conference on Innovations in Science and Technology for Sustainable Development (ICISTSD) | 978-1-6654-9936-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICISTSD55159.2022.10010550

Detection and Classification of Indian Classical Bharathanatyam Mudras Using Enhanced Deep Learning Technique

1st Sneha Haridas
Dept. of Computer Science and Engineering
Vidya Academy Of Science and Technology
Thrissur, India
snehaharidas2898@gmail.com

2nd Dr. Ramani Bai V
HOD,Computer Science and Engineering
Vidya Academy Of Science and Technology
Thrissur, India
ramani.b.v@vidyaacademy.ac.in
ramani.research@gmail.com

***Abstract*—Indian Classical dance forms like bharathanatyam are composed of advanced hand gestures, facial expressions moreover as body moments. Because of its complexity, identifying each mudra in bharathanatyam is extremely difficult. This paper demonstrates a massive Convolutional Neural Network (CNN) that was trained on Google Colaboratory using a single step model, You Only Look Once version 3 (YOLOv3), to analyze images in the dataset and detect and classify the mudras. Open datasets of mudras are not presently available. So Bharatanatyam mudra dataset of single hand gesture images of 28 classes was created. This proposed system is, as far as we know, the first attempt in this subject. YOLOv3 was never used to detect mudras. YOLOv3 divides the image into sectors, predicts bounding boxes, and calculates probability for each. These bounding boxes are then weighted according to the projected probability, and the model is then able to detect the object based on the final weights. The neural network was able to correctly generate test data after being trained, with a mean average precision (mAP) of 73%.**

***Index Terms*—Convolutional Neural Network(CNN), Classification, Deep Learning, YOLOv3, Mudras**

I. Introduction

Dance could be a sort of communication that's fueled by music and adheres to a collection of rules that adjust reckoning on the form. A series of distinct and elementary activity units (action elements) mix with music to embody a plan or specific emotions in dance [3]. Bharatanatyam, a classical dance type that originated in India's southern states, is on the approach of being absolutely mechanised, because of a severe shortage of competent and actuated teachers/gurus. Teachers are the authority who will perceive the precise linguistics which means and feelings of those dance syllabus. Each of the inner and outer feelings that was sent by the dancers is troublesome for novice learners and also for the general audience to understand. The Indian classical dance repository Natyashastra mentions regarding the dance syllables which may be employed in dance forms. It mentions the dance postures referred to as as karanas, Nrita Hastas, Asamyukta hastas, and Samyukta hastas. Bharathanatyam have twenty eight asamyukta hastas and twenty three samyukta hastas. The hasta is conjointly termed as mudra. To perform or to represent Samyukta gestures both the hands are used whereas for representing Asamyukta mudras just one hand is used. Every hasta will be accustomed represent a range of thoughts, ideas, and objects. Shapes of mudras plays an important role within the mudra classification system. several works are planned for form descriptions that are employed in varied applications. within the dance forms, to convey the story line visual expressions, hand gestures and facial expressions are used.

People who are unfamiliar with the significance of dancing gestures will find this helpful. The work is a troublesome one as a result of there are mudras that tally different mudras and will lead to mudra misclassification. The aim of this study is to form a dataset of twenty eight hand gestures and to classify the images into corresponding classes. The study focuses on each single hand gestures. To classify the images, a deep learning technique called Yolov3 is used.

The following is a list of the remaining components of this paper: Section II is devoted to a review of the literature. The methods employed in this system is introduced in Section III. The implementation section is shown in Section IV. The findings is discussed in Section V and the paper is concluded in Section VI.

II. Literature Survey

Mampi Devi et al. [1] focused on classifying the asamyukta hastas of Sattriya dance form. The work consists of a classification mechanism which runs in two phases. Based on Structural similarity and Medical Axis Transformation(MAT), the images from the data set were classified in to 29 classes. As mentioned in the previous works, the application of this

978-1-6654-9936-1/22/$31.00 ©2022 IEEE

work also lies in the self learning and e-learning domains. The work focused on improving the recognition accuracy. So PBF kernel is cincorporated with the SVM model and the accuracy was found to be 97.24%.

Soumitra Samanta et al. [2] used YouTube videos to create their own Indian Classical Dance dataset. Each course contains 30 videos of varying resolutions (maximum resolution: 400x350). Based on the proposed sparse representation, the dictionary learning technique entails using a movement descriptor based on HOOF to represent each frame of the video (Histogram of Oriented Optical Flow). When using SVM to classify video frames, 86.67% is achieved.

Kishore et al. [3] proposes classifying actions of Indian classical dance with the assistance of a convolutional neural net- works (CNN). By taking the offline and online videos of dance steps, act recognition is done. The web videos were taken from live performances and YouTube. The offline videos are created by cinematography with the help of subjects playacting two hundred mudras or acquainted dance steps. During this process, ten subjects were chosen to perform the dance steps in varied backgrounds. Eight totally different samples of variable size is employed for coaching the CNN model. Every sample has different subject set. Out of 10 subject samples, the remaining 2 samples were mounted to check the CNN model. The model consists of five, two and one each layers for corrected Linear Unit(ReLU), random pooling, dense and SoftMax layers respectively.

Anami and Bhandage et al. [4] proposed a method for identifying images of Bharatanatyam dance mudra. Various mudras and vertical-horizontal intersections were used as features in this model. Mudras were classified as either conflicting or nonconflicting. A rule based classifier is deployed in this work to classify images into 24 classes of Bharatanatyam. And the average reported accuracy is 95.25%.

Lai et al. [5] proposed a model for automated hand gesture recognition that combines the power of CNN and RNN. The model used both the depth and skeleton data. Both the types of data can be used to train the neural networks which are intended to recognise the hand gestures seperately. This work focused on applying CNN to extract the prominent spatial information from the depth data taken. To extract the temporal and spatial information, various combinations of skeleton and depth information were conducted. Overall accuracy of 85.46% is obtained for the 14/28 dataset which is a dynamic dataset. In the future, the model could be extended to recognise human activity, which could be used in variety of human assistance applications and scenarios.

Basavaraj and Venkatesh et al.[6] approached the matter with a 3 stage mechanism to classify the mudras. In second stage, the Humoments, chemist values and intersections are extracted out within the third stage, the mudras are classifed mistreatment an ANN.

S. Masood et al. [7] propose a vision-based mechanism to decrypt Argentine Republic signing gestures. They achieved a 95.217 % accuracy. This accuracy worth gave an insight that spatial and temporal features will be simply extracted by incorporating CNN with RNN. The prediction achieved an accuracy of 80.87 % properly recognising 370 gestures out of 460 within the check set, whereas the pool layer approach achieved a score of 95.21 % by correctly recognising 438 gestures.

Nikitha religious belief [8] classified totally different dance forms. In this, they used a deep convolutional neural network model using ResNet50. As per the authors, an accuracy score of 0.911 is obtained for the work. They planned a piece that categorises dance forms thoroughly into eight categories. Furthermore, in terms of performance evaluation, the model outperformed some recent works. Image thresholding and sampling is completed whereas feeding the model with input images. The dataset that they need used consists of 800 images.

Lakshmi Tulasi Bhavanam et al. [9] proposed a model which combines the power of Support Vector Machine(SVM) and Convolutional Neural Network(CNN) to classify the images of dance form into appropriate classes. The work focused on classifying and recognising Kathakali dance mudras into 24 classes. The data set used for the work is a self made one. In Kathakali, the performer uses 24 different types of hand gestures to tell the story. The data set consisted of 654 images of the mudras of Kathakali, with 27 images for each mudra. Both the left and right hands were used to show the mudras. There are two steps in SVM classification - preprocessing and feature extraction. Using the output of the feature extraction phase, the images were classified. The model showed an accuracy of 74%. By examining these existing works, it was discovered that no work had been done in Yolov3.

Fig. 1. Asamyuktha hastas

III. Methodology

A. Proposed System

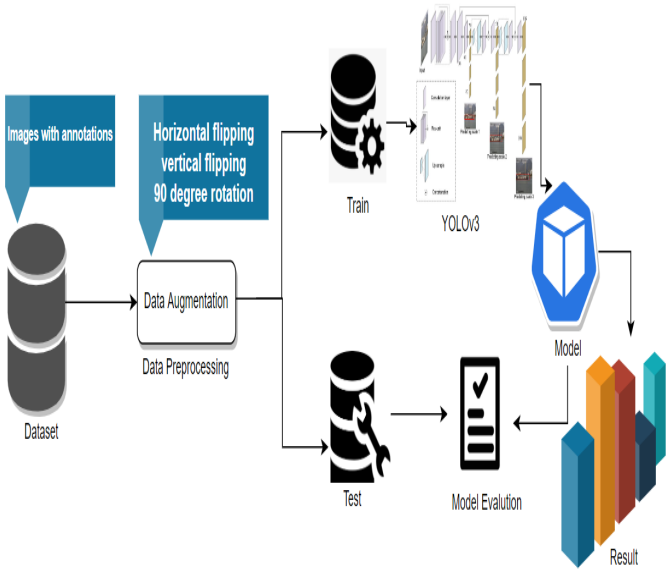

Fig. 2. Proposed System

Bharathanatyam single hand mudras are classified using yolov3. Yolov3 is a object detection algorithm that identifies each and every specific objects in videos, images etc. This work uses images as input. Dataset was created for this. This dataset was split into training and testing set. Then the images are trained for 10 to 14 hours and thus a model was created. This model was used to classify the images.

B. Dataset Preparation

Open datasets are not available. So a dataset has been created for this. The dataset consist of 5000 images and their annotated text. Images are annotated using labelimg. The annotated text are in Yolo darknet form. Image augmentation is a technique that is used to expand the training dataset and it will create different versions of similar contents. In this proposed system, the images are horizontally as well as vertically flipped and rotation of 90° is performed with clockwise, counter-clockwise, upside down directions. Images are preprocessed and resized into 416*416.

C. Model Architecture

YOLO is one of the fastest and accurate object detecting algorithm as compared to R-CNN, Fast R-CNN, Faster R-CNN[11]. You Only Look Once Version 3 (YOLOv3) uses a variant of Darknet,which originally has 53 layer network trained on Imagenet. For detection, 53 more layers are put on top of it, giving YOLOv3 a 106-layer fully convolutional underlying architecture. Yolov3 makes detections at 3 different scales. Prediction of an image is done in a single algorithm.

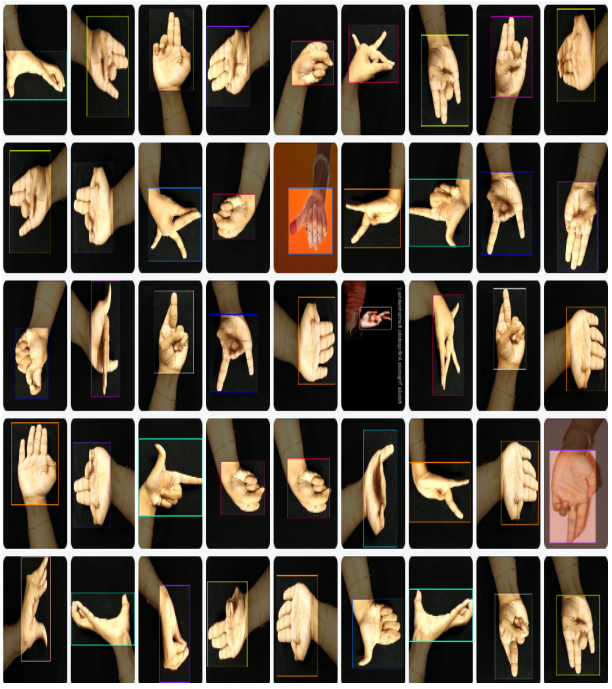

Fig. 3. Annotated images using bounding box

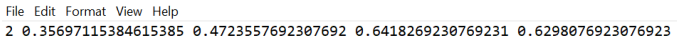

Fig. 4. Coordinates of bounding box with in the image

1) Darknet-53: Darknet-53 is used to extract features. It is primarily made up of 3 x 3 and 1 x 1 filters with skip connections, similar to ResNet's residual network. Darknet-53 is a deeper feature extractor architecture used in YOLOv3.

2) Convolution layers in YOLOv3: It has 53 convolutional layers, each of which is followed by a batch normalisation layer and the activation of the Leaky ReLU. CNN is used to predict the class as well as bounding box simultaneously. Numerous filters are convolved on the pictures using the convolution layer, which results in multiple feature maps. There is no pooling and the feature maps are down sampled using a convolutional layer with stride 2. It aids in the prevention of low-level feature loss, which is frequently linked to pooling.

3) Bounding Box: It is an outline that highlights the objects in the images. Each bounding box have 4 attributes.

- Height (bh)
- Width (bw)
- Bounding box center (bx,by)
- class (c)

Bounding boxes are created by annotating the images. A text file is thus created including the XMin, XMax, YMin, and YMax coordinates of the annotated bounding boxes of the mudra. The bounding boxes are weighted using the probabilities, and the model utilises the final weights to do detection.

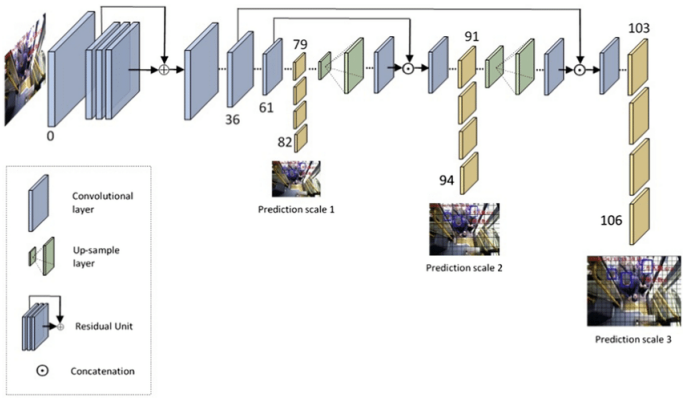

Fig. 5. Network structure of YOLOv3 [11]

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 6. CNN architecture of Darknet-53

IV. IMPLEMENTATION

A. Parameters Initialization

YOLOv3 has a configuration file that provides information such as the class number, max-batches, filters, batch, subdivision, and steps, among other things. For this proposed model, I created a custom cfg file. There are 28 classes in all. As a result, the filter value is 99. (number of classes + 5)*3 is the formula for determining the filter. For training, the batch size is 24 and the subdivision is 16. For example, the batch size in the default yolov3.cfg file is 64 and subdivision is 16, meaning 4 images will be loaded at once, and it will take 16 of these mini batches to complete one iteration.

Create a ”object.names” file that contains the names of the classes that the model is looking for. Then there’s an object.data file with 28 classes, a train data directory, test data, ”object.names,” and a weights path that will be saved in the backup folder.

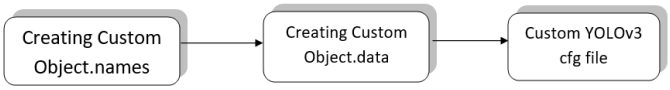

Fig. 7. Configuration Steps

yolov3_custom.cfg

```
[net]
# Testing
#batch= 24
#subdivisions=16
 Training
 batch=24
 subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1
```

Fig. 8. YOLOv3 custom configuration file

B. Training

Yolov3 custom configuration file and yolov3 default weight file were used to train. After each round of training, a weight file is generated, which is then used for testing.

```
8000: 0.084377, 0.092855 avg loss, 0.000100 rate, 2.683526 seconds, 128000 images,
Saving weights to backup/yolov3_custom_8000.weights
Saving weights to backup/yolov3_custom_last.weights
Saving weights to backup/yolov3_custom_final.weights
```

Fig. 9. Generated Weight File

C. YOLOv3 Algorithm Applied

An image is passed into the YOLOv3 model as an input. This object detector searches through an image for the coordinates that are present. It basically divides the input into a grid and examines the target object’s attributes from that grid. The features that were recognised with a high confidence rate

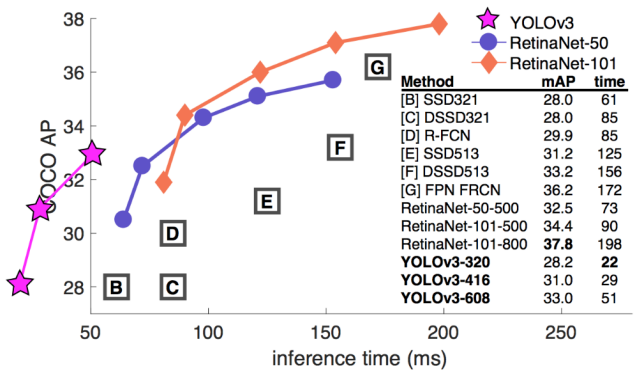

Fig. 10. Comparison with other detection methods[10]

in nearby cells are combined in one place to provide model output.

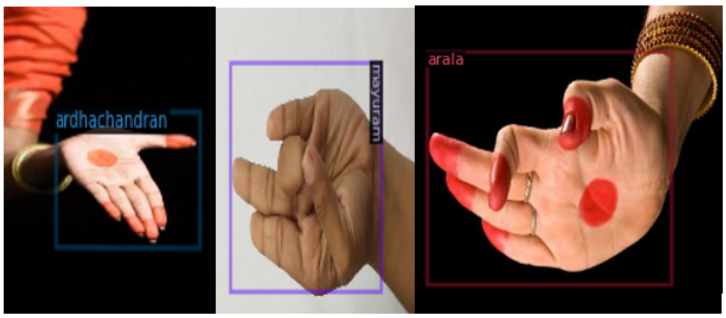

Fig. 11. Mudra Detected

V. EXPERIMENTAL RESULT

A. Environmental Setup

To do this, Opencv and Python 3.7.13 were used to train the algorithm. Version 11.2 of CUDA is utilised. On a personal PC with a local 4GB GPU, the algorithm was trained. The Colab notebook was utilised. On a GPU, training the neural network model on dataset took about 10 to 11 hours.

TABLE I
SPECIFICATIONS OF YOLOV3

Specifications	Parameters used
Batch size	24
Subdivision	16
Filters	99
Classes	28
Max-batches	8000
Loss function	Binary cross-entropy

B. Test Result

Twenty of the 28 classes have precision ranging from 70 to 100 %. Others have a precision of less than 50%. This proposed system has 73 percentage mean average precision (mAP@0.50). Yolov3 final weight file and custom configuration file were used for testing.

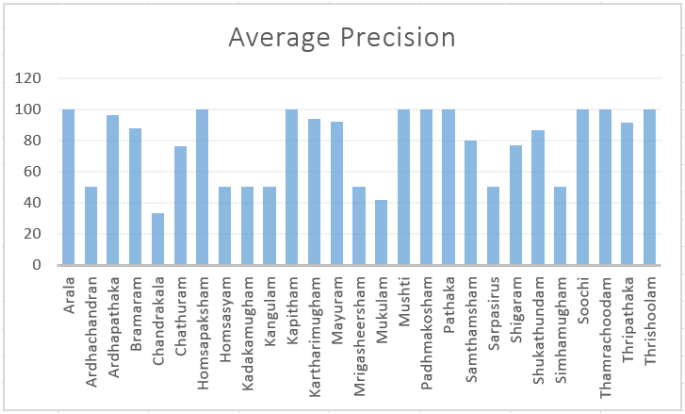

Fig. 12. Average precison chart

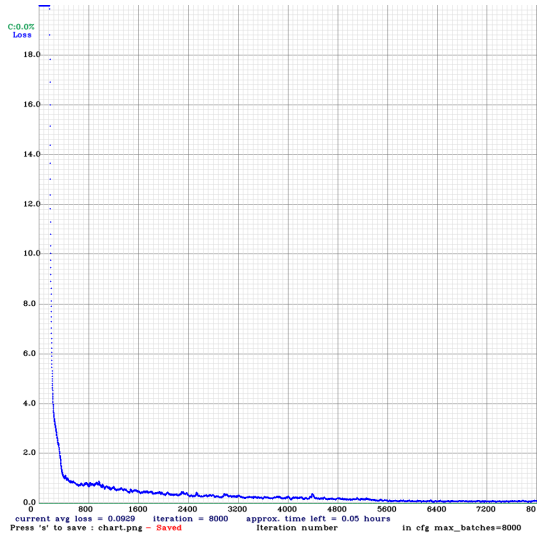

Fig. 13. Graph of losses Vs iterations

VI. CONCLUSION AND FUTURE WORK

Indian classical dances such as Kathakali and Bharathanatyam are made up of hand gestures, body movements, and facial expressions which are inlined with the background music. Mudras are the basic elements of Bharathanatyam. Understanding the bharathanatyam mudras are very much difficult because of the complexities arises with the hand-gestures. Public dataset availability is comparatively lower, so an expert dancer is required to create the dataset. Bharathanatyam has 28 asamyuktha mudras that are classified with an accuracy of 73% through this work. This work is mainly focused on mudras. It can be extended where the mudras go along with the adav. In future, an extension of this methodology can be done to check whether the mudras are correct or not along with adavus. So it will be helpful for those who want to learn the dance online. The foreigners

who is interested to know and learn the Indian dance forms can get the meaning or decode the mudras of Bharathanatyam dance form. .

REFERENCES

[1] M. Devi and S. Saharia, "A two-level classification scheme for single-hand gestures of Sattriya dance," ICADW, Guwahati, 2016.

[2] S. Samanta, "Indian Classical Dance classification by learning dance pose bases," IEEE Workshop on the Applications of Computer Vision.

[3] K.V.V. Kumar, P.V.V. Kishore, "Indian Classical Dance Mudra Classification Using HOG Features and SVM Classifier," International Journal of Electrical and Computer Engineering (IJECE), 2017

[4] Anami BS, Bhandage VA, " A vertical-horizontal-intersections feature based method for identification of bharatanatyam double hand mudra images," Springer,2018.

[5] Kenneth Lai and Svetlana N. Yanushkevich, "CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition," IEEE, 2018.

[6] Basavaraj S. Anami and Venkatesh A. Bhandage, "A Comparative Study of Suitability of Certain Features in Classification of Bharatanatyam Mudra Images Using Artificial Neural Network," part of Springer Nature, 2018.

[7] Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal and Musheer Ahmad, "Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN," Advances in Intelligent Systems and Computing, Springer

[8] Nikita Jain,Vibhuti Bansal, Deepali Virmani, Vedika Gupta,Lorenzo Salas-Morera and Laura Garcia-Hernandez, "An Enhanced Deep Convolutional Neural Network for Classifying Indian Classical Dance Forms," MDPI,2020.

[9] Lakshmi Tulasi Bhavanam and Ganesh Neelakanta Iyer, "On the Classification of Kathakali Hand Gestures Using Support Vector Machines and Convolutional Neural Networks," International Conference on Artificial Intelligence and Signal Processing (AISP), 2020.

[10] Nurul Iman Hassan,Fadhlan Hafizhelmi Kamaru Zaman,Nooritawati Md. Tahir, Habibah Hashim, "People Detection System Using YOLOv3 Algorithm," IEEE , 2020.

[11] Joseph Redmon, Ali Farhadi, "YOLOv3: An Incremental Improvement," University of Washington,2020.

[12] Abdullah Mujahid,Mazhar Javed Awan,Awais Yasin,Mazin Abed Mohammed, "Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model," Applied Sciences, 2021.