Implementation of Machine Learning Technique for Identification of Yoga Poses

Yash Agrawal*, Yash Shah*, Abhishek Sharma†
*†*Department of Electronics and Communication Engineering*
The LNM Institute Of Information Technology, Jaipur, Rajasthan, India
17uec136@lnmiit.ac.in, 17uec139@lnmiit.ac.in, abhisheksharma@lnmiit.ac.in

***Abstract*—In recent years, yoga has become part of life for many people across the world. Due to this there is the need of scientific analysis of y postures. It has been observed that pose detection techniques can be used to identify the postures and also to assist the people to perform yoga more accurately. Recognition of posture is a challenging task due to the lack availability of dataset and also to detect posture on real-time bases. To overcome this problem a large dataset has been created which contain at least 5500 images of ten different yoga pose and used a tf-pose estimation Algorithm which draws a skeleton of a human body on the real-time bases. Angles of the joints in the human body are extracted using the tf-pose skeleton and used them as a feature to implement various machine learning models. 80% of the dataset has been used for training purpose and 20% of the dataset has been used for testing. This dataset is tested on different Machine learning classification models and achieves an accuracy of 99.04% by using a Random Forest Classifier.**

***Index Terms*—YOGI - YOga Gesture Identification dataset, Computer Vision, Machine Learning, Classification, Gesture Recognition.**

I. INTRODUCTION

Yoga is originated in ancient India and it is a group exercise associated with mental, physical and spiritual strength. Yoga and sports have been attracting peoples from so many years but from the last decade, a large number of people are adopting yoga as part of their life. This is due to the health benefits. It is important to do this exercise in right way specially in right posture. It has been observed that sometime due to lack of assistance or knowledge people don't know the correct method to do yoga and start doing yoga without any guidance, thus they injure them-self during self-training due to improper posture. Yoga should be done under the guidance of a trainer but it is also not affordable for all the peoples. Nowadays people use their mobile phones to learn how to do yoga poses and start doing that but while doing that they don't even know that the yoga pose they are doing is in the right way or not. To overcome these limitations, many works have been done. Computer vision and data science techniques have been used to build AI software that works as a trainer. This software tell about the advantages of that pose. It also tell about the accuracy of the performance. Using this software one can do yoga without the guidance of a trainer. To use machine learning and Deep learning modules a Large number of image dataset has been created which contain 10 yoga pose (Vriksasana, Utkatasana, Virabhadrasana I, Parsva Urdhva Hastasana, Baddha Konasana, Standing, Bhujang asana, Sukhasana, plank, and Virasana). Features have been extracted using computer vision and tf-pose Algorithm. This Algorithm draws a skeleton of a human body (shown in figure 4) by marking all the joint of a body and connects all the joints which give a stick diagram known as the skeleton of a body. Coordinates and the angles made by the joints can be extracted using this algorithm and then used that angles as features for machine learning models. Several machine learning models has been used to calculate the test accuracy of the model. Random Forest classifier gives the best accuracy among all the models.

II. RELATED WORK

Researches have been done on yoga pose detection and correction[1],[2]. Some researchers have used a Kinect device to form a human posture [3]. This device used to capture the images but the important part is that this device contains an inbuilt infrared laser projector, a multiarray microphone and an RBG camera used to capture the color and depth images. This device also has a tool that makes a human body skeleton in 3D space which gives the information about the coordinates of the joint of the body. This method is good but the main disadvantage of this method is that the Kinect device is expensive and not user-friendly. To avoid this problem tf-pose Algorithm has been used. This Algorithm creates a skeleton of a human body and gives the desired information about the joint in the human body. Using this one can find the coordinates of the joints and use that as a feature to detect the posture of a body. Paula Pullen, William Seffens [4] used visual Gesture Builder feature of kinect sensor which used to capture yoga postures with high accuracy. Edwin W. Trejo, Peijiang Yuan[5] has also used Microsoft Kinect v2 which gives more accuracy and precision but for a more complex model, it requires more computational time. The Kinect camera is a device which works on 3 things (depth,color, and body tracking) [6],[7],[8]. Using all the features of the Kinect device they develop a Computer Interaction system for training purpose an Adaboost Algorithm has been used to recognize 6 common yoga pose. Many authors have worked on different applications of kinect device and came to a conclusion that it works well for depth, color and body tracking [9],[10],[11].. Another technique for

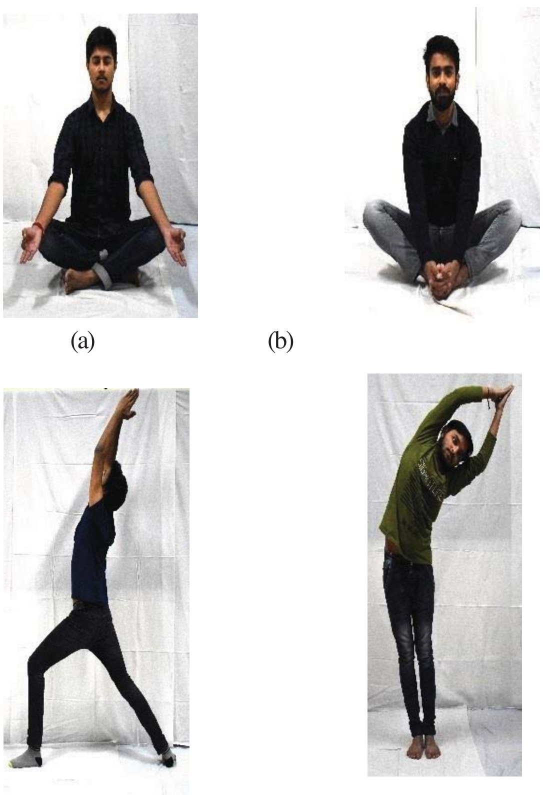
978-1-7281-4976-9/20/$31.00 ©2020 IEEE
DOI: 10.1109/CSNT.2020.08

pose detection is to use a Convolution Neural Network. In [12] authors used a dataset of 6 yoga asana. They used across deep learning model of CNN and LSTM to recognize a yoga pose where features are extracted using CNN from the points acquired by Open Pose and LSTM is used to detect yoga pose. Using this module they reach an accuracy of 99.04% on a single frame.

III. DATASET COLLECTION

Currently, finding a precise and effective yoga-pose dataset on the web is a challenge in itself. YOGI dataset is a mixture of both standing poses and sitting poses, it makes use of the whole body in depicting any yoga pose. The poses have a variety of different hand and leg folds which make it difficult for posture detection algorithm to work efficiently.

Realizing the above problem, YOGI dataset consist of10yoga poses which were captured using burst feature of the DSLR camera. The images were taken with high precision and accuracy. There are 10 yoga poses, each class containing around 400 to 900 images. The complied Colour Image dataset consists of 5459 images. Yoga-pose of four classes are shown in Fig.1



(a) (b)

(c) (d)

Fig 1: Images of the YOGI dataset

A. The procedure followed in collecting YOGI dataset

Collecting data manually of a huge volume requires a lot of effort, attention and precision. Listed below is the procedure.

- A closed room was used to avoid any use of direct sunlight to get images without any reflection or glare.
- The camera was mounted and adjusted on a tripod with an appropriate frame centering the person performing the yoga poses, and the distance was maintained around 4 to 5m between the camera and the person.
- The background was kept plain white to enhance and distinguish the yoga poses done by the person.
- Images of every pose were clicked from multiple directions and angle with the motion to capture every form of the pose, This helped to create a mixed real-time dataset.
- Images of each pose were clicked in continuous mode ,25 images in a single go.

IV. EXPERIMENTS AND RESULTS

After collecting YOGI dataset the following steps were followed as shown in Fig 2.

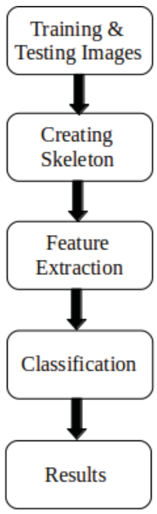
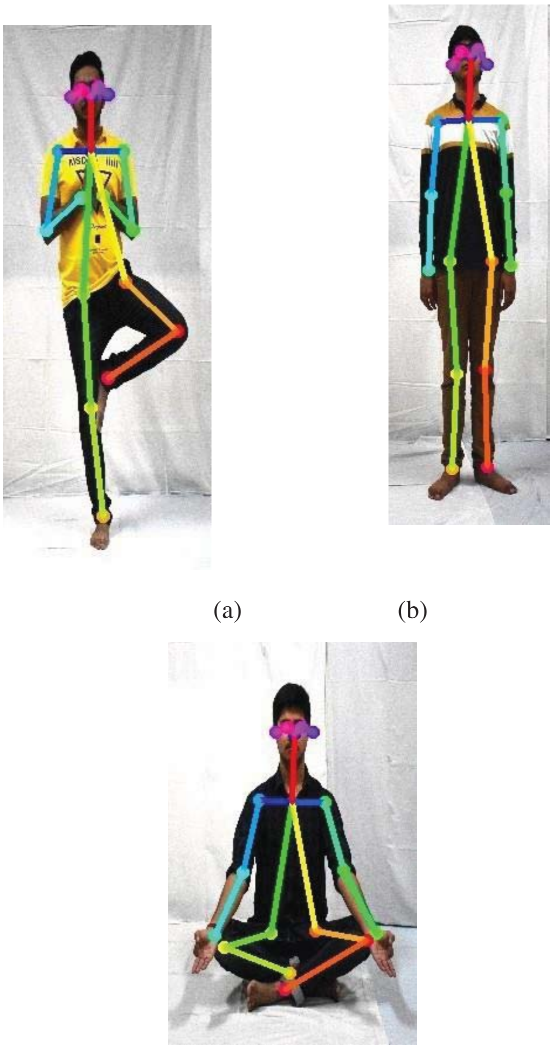


Fig 2: Flowchart of the procedure followed on the dataset.

A. Creating Skeleton

In the first stage, the Brightness of every image was increased and the parameter enhance was set to a value of 2.0 for uniformity. Followed by this, the new images were resized to 500x500 resolution to best fit the pose estimation algorithm for the accurate and precise outcome.

The tf-pose-estimation algorithm is used to create a skeleton of the person performing the yoga poses, the algorithm marks each joint of the body and connects it with a skeleton/stick diagram as shown in Fig 3. The algorithm works very accurately in real-time as well



(a) (b)

(c)

Fig 3: Skeleton made using Tf-pose estimation

B. *Feature Extraction*

In the next stage, using tf-pose-estimation algorithm the coordinates of the joints are extracted, the number of joints are shown in Fig 4. The coordinates are used to calculate 12 different angles which will be used as features to detect and correct the yoga poses. The formula for calculating the angle [13] is shown below.

Here's an equation:

$$a^2 = b^2 + c^2 - 2bc\cos A \quad (1)$$

Where,

a = Distance between point p1 and p2
b = Distance between point p2 and p3
c = Distance between point p1 and p3
A = Angle made by point p2

To find the distance between two points [14]

$$a = \sqrt{(x1 - x2)2 + (y2 - y1)2} \quad (2)$$

Where,

(x1,y1) is the coordinate of point p1 (x2,y2) is the coordinate of point p2

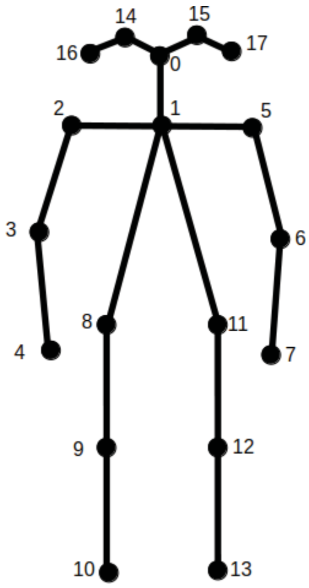


Fig 4: Stick Diagram using tf-pose Algorithm

C. *Classification*

In the final stage, the features are stored in a CSV file and labeled accordingly. The data is then split into training and test dataset in 80:20 ratio as shown in Table-I. Six classifica- tion models of machine learning namely Logistic Regression, Random Forest, SVM, Decision Tree, Naive Bayes and KNN with different parameters were used as a point of reference for the dataset. After the final evaluation, 25 results of accuracy were found. Following results are mentioned in Table-II.

TABLE I
DESCRIPTION OF DATASET

Category	*Number of Example*
Training set	4367
Test set	1092

TABLE II
BENCHMARK OF YOGI DATASET

Classifier	*Description*	*Accuracy*
Logistic Regression	IterationNumber: 1000, Solver: Newton-cg	0.8215
	IterationNumber: 1500, Solver: Newton-cg	0.8302
	IterationNumber: 2000, Solver: Newton-cg	0.8379
	IterationNumber: 2500, Solver: Newton-cg	0.8316
Random Forest	n estimator: 30, MaxDepth: 7	0.9926
	n estimator: 30, MaxDepth: 10	0.9972
	n estimator: 30, MaxDepth: None	0.9990
SVM	Kernal Function: Linear, Loss Function: Hinge	0.8791
	Kernal Function: Polynomial, Loss Function: Hinge	0.9358
	Kernal Function: Radial Basis Function, Loss Function: Hinge	0.9871
Decision Tree	MinSampleLeaf: 1, MinSampleSplit: 2	0.9771
	MinSampleLeaf: 2, MinSampleSplit: 2	0.9670
	MinSampleLeaf: 3, MinSampleSplit: 2	0.9679
	MinSampleLeaf: 1, MinSampleSplit: 3	0.9752
	MinSampleLeaf: 2, MinSampleSplit: 3	0.9670
	MinSampleLeaf: 3, MinSampleSplit: 3	0.9761
Naive Bayes	Distribution: Normal	0.7475
KNN	Neighbours: 3, DistanceWeight: Equal, Distance: Euclidean	0.9899
	Neighbours: 3, DistanceWeight: Inverse, Distance: Euclidean	0.9826
	Neighbours: 5, DistanceWeight: Equal, Distance: Euclidean	0.9853
	Neighbours: 5, DistanceWeight: Inverse, Distance: Euclidean	0.9901
	Neighbours: 7, DistanceWeight: Equal, Distance: Euclidean	0.9826
	Neighbours: 7, DistanceWeight: Inverse, Distance: Euclidean	0.9890
	Neighbours: 9, DistanceWeight: Equal, Distance: Euclidean	0.9725
	Neighbours: 9, DistanceWeight: Inverse, Distance: Euclidean	0.9891

V. CONCLUSION

In this paper, a system is suggested that classify ten yoga poses and the dataset upholds on six classification models of machine learning. The yoga pose is detected based on the angles extracted from the Skeleton joints of TF pose estimation algorithm. 94.28% accuracy altogether was attained of all machine learning models. The data preprocessing and model training was done on Google Colab and Ubuntu 18.04.4 LTS terminal. Future ideas also includes expansion of YOGI dataset on more yoga poses and implement deep learning modules for better performance. In addition to that an audio guidance system will also be included.

ACKNOWLEDGMENT

We would like to thank L-CST(LNMIIT Centre for Smart Technology) for the guidance. Also would like to thank our colleagues Ritik Bansal, Anupum Shah, Akshat Jaiman and Rohit Munjal for helping in data preparation.

REFERENCES

[1] Muhammad Usama Islam ; Hasan Mahmud ; Faisal Bin Ashraf ; Iqbal Hossain ; Md. Kamrul Hasan ”Yoga posture recognition by detecting human joint points in real time using microsoft kinect.” IEEE Region 10 Humanitarian Technology Conference (R10-HTC). pp.1-5, 2017.

[2] Hua-Tsung Chen, Yu-Zhen He, Chun-Chieh Hsu, Chien-Li Chou, Suh-Yin Lee, Bao-Shuh P. Lin, “”Yoga posture recognition for self-training.” International Conference on Multimedia Modeling. Springer, pp.496-505, 2014.

[3] Xin Jin ; Yuan Yao ; Qiliang Jiang ; Xingying Huang ; Jianyi Zhang ; Xiaokun Zhang ; Kejun Zhang, “Virtual personal trainer via the kinect sensor” IEEE 16th International Conference on Communication Technology (ICCT). pp.1-6, 2015.

[4] Pullen, Paula, and William Seffens. “Machine learning gesture analysis of yoga for exergame Development.” IET Cyber-Physical Systems: Theory Applications, vol.3, no.2, pp.106-110, 2018.

[5] Trejo, Edwin W., and Peijiang Yuan. ”Recognition of Yoga poses through an interactive system with Kinect device.” 2nd Inter- national Conference on Robotics and Automation Sciences (ICRAS), 2018.

[6] Kodama, Tomoya, Tomoki Koyama, and Takeshi Saitoh, “Kinect sensor based sign language word recognition by mutli-stream HMM.” 201756th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp.1-4, 2017.

[7] Călin, Alina Delia.” Variation of pose and gesture recognition accuracy using two kinect versions.” 2016 International Symposium on Innovations in Intelligent Systems and Applications, pp.1-4, 2016.

[8] Qifei Wang; Gregorij Kurillo; Ferda Ofli; Ruzena Bajcsy; “Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect” International conference on healthcare informatics, pp.1-11, 2015.

[9] Gonzalez-Jorge, H., Diaz Vilarino L., “Metrological Comparison Between Kinect I and Kinect II Sensors.” Elsevier Measurement, Vol.70, pp.21=26, 2015.

[8] Khoshelham, Kourosh, and Sander Oude Elberink. “Accuracy and Resolution of kinect Depth Data for Indoor mapping applications”, Sensors, Vol. 12, No.2, pp.1437-1454, 2012.

[9] Yang Lin, Zhang Longyu; Dong Haiwei, ”Evaluating and improving the depth accuracy of Kinect for Windows v2.” IEEE Sensors Journal, Vol.15, No.8, pp. 4275-4285, 2015

[10] S.K. Yadav, . Singh, A. Gupta, J. L. Raheja, “Real-time Yoga Recognition using Deep Learning” vNeural Computing and Applications, Vol.31, pp. 9349-9361, 2019.

[11] https://en.wikipedia.org/wiki/Law of cosines.

[12] https://en.wikipedia.org/wiki/Euclidean distance