Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 1, pp. 143-155
DOI : 10.47974/JIOS-1859

Enhancing wellness through AI-powered yoga assistant : A human activity recognition application

Neha Gupta [#]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Yogita Arora [†]
Department of Electronics & Communication Engineering
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Sarita Yadav [§]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Neera Aggarwal [‡]
Department of Electronics & Communication Engineering
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

[#] *E-mail:* neha.06.gupta@gmail.com
[†] *E-mail:* yogitamarora@gmail.com
[§] *E-mail:* sarita1320@yahoo.co.in
[‡] *E-mail:* neera711@gmail.com

Sangeeta Gupta [@]
Department of Instrumentation and Control Engineering
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Meenakshi Gupta *
Department of Electronics & Communication Engineering
Manav Rachna University
Faridabad 121001
Haryana
India

Prakhar Priyadarshi [$]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Surinder Kaur [^]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
Paschim Vihar 110063
New Delhi
India

Abstract

Prioritizing health is crucial, yet it remains unfulfilled and requires increased attention. To address this issue, an automated artificial intelligence (AI) application that monitors postures and actions during exercise would be highly beneficial. In the proposed article, we offer a remote AI powered yoga trainer. This trainer incorporates OpenCV for live image capture and processing, which helps in identifying the key poses. Advanced AI techniques, such as deep learning, have proven beneficial in monitoring applications. In the proposed article, we proposed three AI yoga trainer models using deep neural network (DNN), convolutional neural network (CNN), and multilayer perceptron (MLP) models. We evaluated the proposed model's performance on a publicly available dataset sourced

[@] *E-mail:* **sangeeta.gupta@bharatividyapeeth.edu**
* *E-mail:* **meenakshigupta@mru.edu.in (Corresponding Author)**
[$] *E-mail:* **priyadarshi.prakhar@bharatividyapeeth.edu**
[^] *E-mail:* **kaur.surinder@bharatividyapeeth.edu**

from Kaggle, where we focused on obtaining yoga pose images, as well as their captions. Data processing, which includes data cleaning and augmentation, is performed to ensure the diversity of the dataset. A carefully curated set of yoga poses, such as chair, cobra, and shoulder stand, helps to provide beginner-friendly and health-beneficial practices remotely, thereby preventing the neglect of physical health.

Subject Classification: *68T07.*

Keywords: *HAR, Yoga Trainer, CNN, DNN, MLP.*

1. Introduction

With the increasing screen time and people being glued to electronic devices, it is even more concerning that physical health is something that is neglected tremendously. Poor physical health, in turn, leads to chronic conditions. Since everyone is also busy with their own work and schedule, it becomes even more difficult to go to a trainer and coordinate everything according to their own convenience. But in situations like these, a HAR based remote trainer who can fit according to anyone's schedule is a game changer. We have built an AI-based yoga trainer [1], [2] .

To monitor exercise Adeyemi *et al.* proposed a AI based yoga trainer which make use of OpenCV for identifying the key poses which a person is doing yoga pose [1]. Similarly, Ding *et al.* proposed a free weight exercise monitoring application using radio frequency identification tags for monitoring the postures [3]. Kanase *et al.* proposed an OpenPose based application where OpenPose is a pre trained multi-stage CNN to detect a user's posture. [4]

In the proposed article, we offer an AI-powered yoga trainer application that includes the TensorFlow MoveNet model. We also utilize the OpenCV library for real-time image capture and processing, while JavaScript powers the frontend. With our trainer, people will have fewer excuses and more reasons to take good care of their physical health. Since yoga also has a ton of health benefits, including flexibility, strength, stress control, and improving the overall quality of life, a remote trainer such as ours goes with individuals everywhere and can fit into any schedule, no matter how busy it gets.

2. Literature Review

Upon reviewing further publications related to trainers built under the umbrella of human activity recognition, we noted the following:

Adeyemi et al. utilizes OpenPose to locate joints and determine key points [1]. Sensor based HAR models utilizes smartphone magnetometers, gyroscopes, and accelerometers to create a wearable device that monitors posture real time, which is also beneficial for back-related problems [4]–[6]. The vision-based approach to yoga posture detection, utilizing a mobile camera and deep learning techniques [7]–[10]. The next category we studied explores the evolutionary aspect of posture and its impact on human health to understand the relationship between daily posture and bone health. Based on the types of papers we studied, we observed that the use of the Tensorflow MoveNet model is efficient and suitable for lightweight real-time pose estimation. While there are benefits and drawbacks to every approach, MoveNet's high accuracy and quick inference speeds are a major asset to our project [11].

3. Proposed AI Yoga Trainer Model

3.1 *Dataset*

We have focused on a specific set of yoga poses in this project. The source of the data used is Kaggle, a machine learning and data science community. Table 1 shows the yoga poses we have considered in training the AI powered Yoga trainer application. There are two different categories in the data from Kaggle. The first section consists of pictures of yoga poses to identify the proper form. We can categorize them into multiple groups through the pictures. Then the second Kaggle component is the English-language data to deal with poses.

Table 1

Dataset Description

Activities	# Train Samples	# Test Samples
Tree Pose	418	192
Chair Pose	400	168
Dog Pose	400	180
Warrior Pose	400	218
Triangle Pose	315	70
Cobra Pose	400	232

3.2 *Data pre-processing*

The dataset contains many different poses and environments from public sources. We started the preprocessing of the data by cleaning up the data and making sure that any missing or incorrect values in the dataset are removed. Deleting any inconsistent or unnecessary data points and removing any noisy data points can have a negative impact on the training of the model. Then, we used data augmentation to improve the diversity and robustness of the dataset using noise addition, flips, and rotations. Next, to enhance model performance, we normalized the image pixel values to a standard range. We concluded with setting up a pipeline that will automate and organize the preprocessing tasks. To have consistent training, validation, and testing data we use this pipeline.

3.3 *Proposed Methodology*

In this section the proposed methodology for Yoga pose identification is presented. Fig. 1 illustrates the flowchart of proposed AI-powered Yoga trainer.

User Interaction: User selects a yoga pose from the dropdown menu and clicks "Start Pose." The webcam captures live video input from the user.

Pose Detection and Classification: Frontend captures video frames from the user's webcam. TensorFlow.js pre-processes the video frames. It sends video frames to the Pose Detection Backend for real-time pose estimation

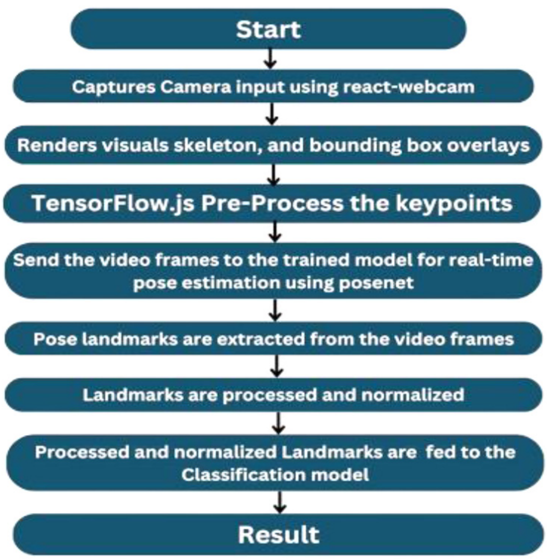


Figure 1

Steps for Yoga pose detect

using PoseNet. Pose landmarks and keypoints are extracted from the video frames.

Visual Feedback: The Canvas Component overlays visual feedback on the webcam feed, highlighting key points, skeleton, and bounding box based on pose detection results. Skeleton color changes to indicate correct pose alignment.

Performance Tracking: The application tracks pose duration and best performance metrics. Audio feedback provides cues when the user achieves correct pose alignment and maintains it for a specific duration.

3.4 *Deep Learning Algorithms*

3.4.1 Multi-Layer Perceptron

MLPs are a feed-forward neural network employed for a variety of tasks, like classification, pattern recognition etc. They are capable of processing both structured and unstructured data. The MLP network consists of three layers: input, hidden, and output which is illustrated in Fig 2. Every neuron in one layer is linked to each neuron in the subsequent layer.

$$h(r) = \Phi(r) = \rho(V_1 g(r) + c_1$$

$$g(r) = \psi(V_2 h(r) + c_2$$

The MLP demonstrates excellent accuracy on both training and test datasets, making it a viable option. However, it does not perform as well as DNNs and CNNs in terms of validation accuracy.

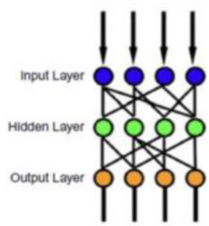


Figure 2

Multilayer Perceptron Model (MLP)

3.4.2 Deep Neural Networks (DNN)

DNNs are neural networks with multiple hidden layers is presented in Fig 3. DNN has the ability to identify patterns and relationships within the data. Even after having lower accuracy than CNN the true capability of the DNN is demonstrated by its generalization to unknown test data. Its performance likely stems from its ability to identify complex relationships within the data and the regularization advantages provided by dropout layers.

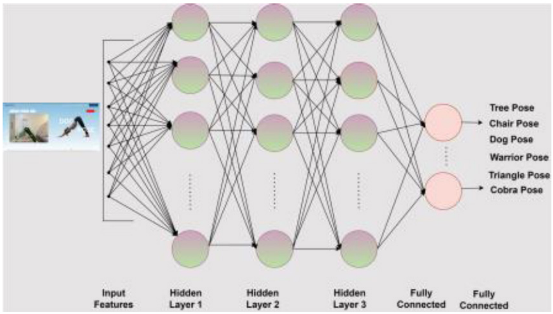


Figure 3

Deep Neural Network (DNN).

3.4.3 CNN

Convolutional Neural Networks, or CNN, is a deep learning algorithm primarily utilized for tasks involving image recognition and spatial data analysis. CNN consists of multiple layers like convolutional layers, pooling layers, and fully connected layers. CNNs draw inspiration from the way the human brain processes images and they are best for capturing patterns and spatial data. Fig 4. Shows the proposed CNN architecture for Yoga pose detection. The convolution layer is the core of CNN. It contains the main portion of the network's load. Convolutional layers use filters to extract features like edges, textures, and forms from the input data

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

The pooling layer reduces the spatial dimensions, which decreases the required amount of computation and weights. It achieves it by retaining important information while reducing computational complexity.

$$W_{out} = \frac{W - F}{S} + 1$$

Fully connected layers are used at the end of CNNs for classification tasks, where they combine extracted features to make predictions. It helps to map the representation between the input and the output.

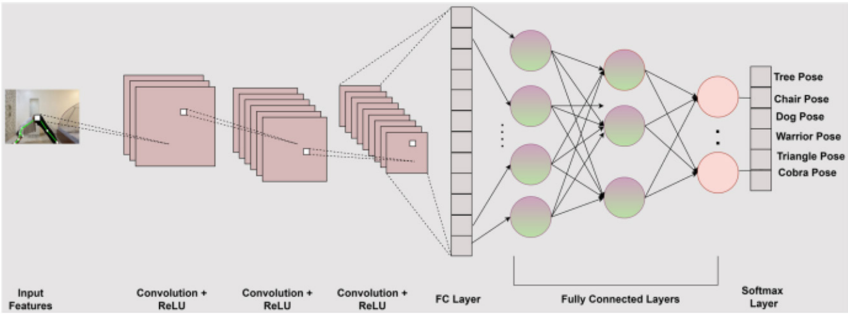


Figure 4

Convolutional Neural Network (CNN).

3.4.5 MoveNet

Based on the comparative analysis, it implies that CNN algorithm performs better than other algorithms in terms of metrics like accuracy, precision, recall, F1 score etc. CNN has features like Spatial Hierarchical Features, Parameter Efficiency, Reduced Overfitting and are highly effective for tasks on data with images, making CNN our best option. CNN aligns with our goal of obtaining accurate and reliable pose detection while offering a robust user experience. The MoveNet is a deep learning model made by Tensorflow. We can use the generic CNN model for pose estimation, but we would rather use MoveNet due to its lightweight and efficient design, enabling real-time performance. It's tailored for single person pose estimation results in faster results with lower resource consumption. MoveNet is an extremely fast and precise model that identifies 17 key points on a body. MoveNet which is based on the CNN model is one of the best options out there to get accurate predictions and detection. There are two versions of MoveNet: Lightning which is performance-oriented, and Thunder which is accuracy-oriented. In our case we chose to use MoveNet Thunder to get the best accuracy. Fig 5. Presents the MoveNet architecture where the input tensor with size [B, H, W, 3], where B stands for the batch size and it shows # images processed in an iteration, H represents each image height, whereas W shows the image width, and 3 represents the # channels in image [11]. Scales s4 to s32 present the extracted features resolution where low-resolution(s32) is for broader context such as key pose estimation and high-resolution(s4) is for very specific objects. 3*3 convolutional layer is for extracting larger scale features whereas 1x1 repossesses finer details.

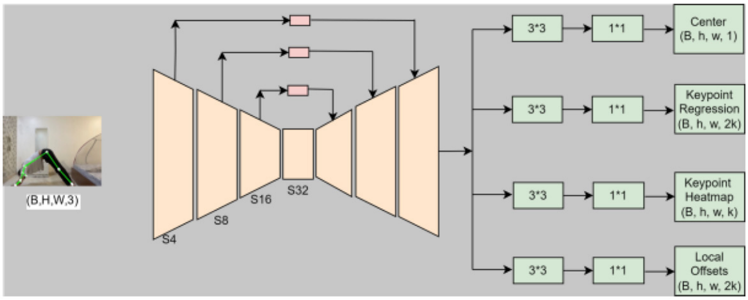


Figure 5

MoveNet Model: B: Batch size, H: image height, W: image width, S4-S32: scales, k: # keypoints

3.4.6 Model Training

There were multiple steps in the model training and testing. At first, the dataset was loaded from a CSV file using pandas, and the input data was normalized, unnecessary columns were eliminated, and labels were converted to categorical format. For Training and Testing sets a ratio of 85-15 was devised. After that, a custom function was created to take the pose landmarks and extract key points and then convert them into a pose embedding. The neural network model was made with layers using

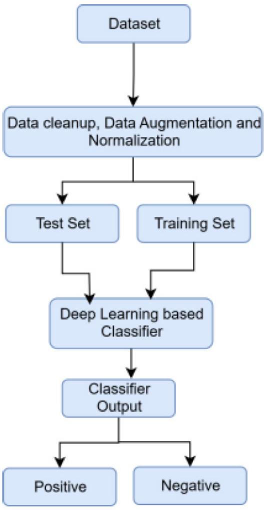


Figure 6

Model train and test

TensorFlow's Keras API. A loss function and an Adam optimizer were used in the compilation of the model. During training, a checkpoint callback was added to save the model weights corresponding to the highest validation accuracy achieved. We trained the model with a batch size of 16 over 200 epochs. After training, the model was evaluated on the test set to calculate loss and accuracy. The model's performance was saved in TensorFlow.js format for deployment in web applications. Fig. 6 illustrates the proposed models train and test strategies.

4. Performance Evaluation

4.1 *Yoga Pose Detection*

This trainer incorporates OpenCV for live image capture and processing, which helps in identifying the key poses. Fig 7. shows the output of OpenCV for all 6 yoga poses where 17 keypoints are identified in each. Fig. 7 shows following 6 poses: (a) Tree pose, (b) chair pose, (c) Dog Pose, (d) Warrior pose, (e) Triangle pose, (f) Cobra pose. Frontend captures video frames from the user's webcam. TensorFlow.js pre-processes the video frames. It sends video frames to the Pose Detection Backend for real-time pose estimation using PoseNet. Pose landmarks and keypoints are extracted from the video frames. Landmarks are then processed and normalized for pose size and orientation.

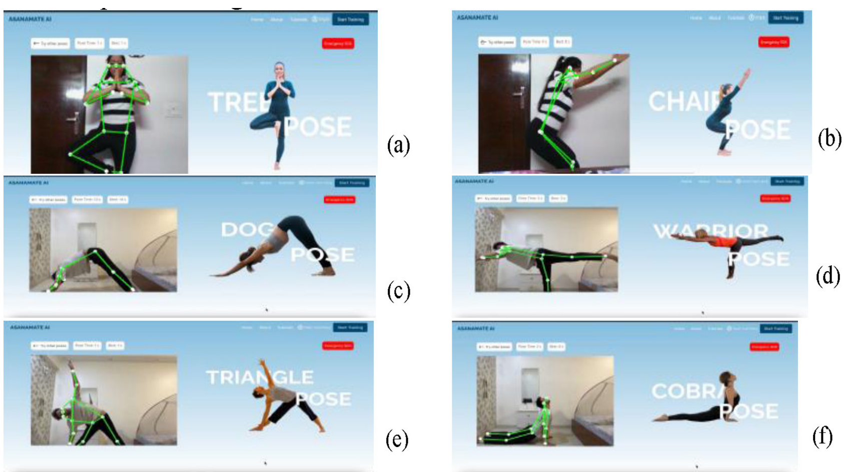


Figure 7

Yoga poses detected with OpenCV: (a) Tree pose, (b) Chair pose, (c) Dog Pose(d) Warrior pose, (e) Triangle pose, (f) Cobra pose

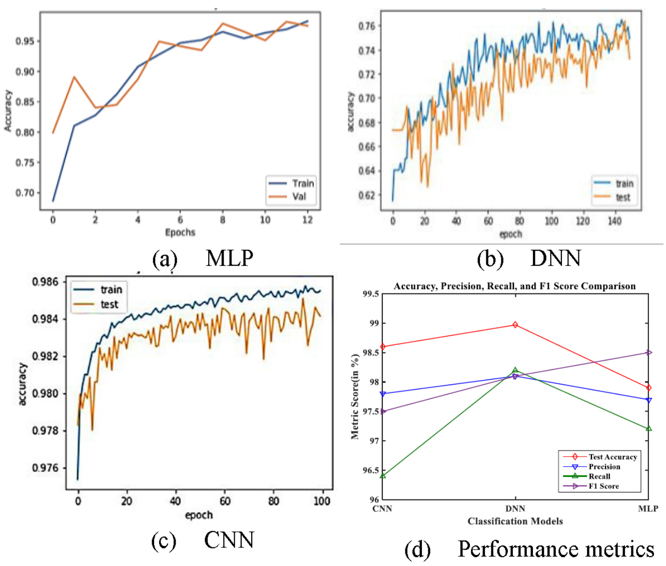
4.2 *Results*

In the proposed article, we have evaluated the AI Yoga trainer performance on the following metrics: accuracy, precision, recall and F1 score. Table 2. Illustrates the performance of the proposed model, MoveNet and CNN are best performed models with the accuracy of 98.67% and 98.41% respectively. Fig. 8 shows the performance MLP, DNN, and CNN models accuracy plots for training as well as for test data, Fig. 8 (a) to (d) presents the performance of proposed MLP, DNN and CNN models in terms of accuracy, precision, recall and F1 score. As per our analysis and graphs presented, all three models has remarkable performance with accuracy between 97.9% to 98.6%.

Table 2

Performance evaluation

Model	Train Acc	Val Acc	Test Acc	Precision	Recall	F1 Score
MLP	97.73	98.1	97.9	97.7	97.2	98.5
DNN	98.57	97.94	98.97	98.1	98.2	98.1
CNN	98.41	99.1	98.6	97.8	96.4	97.5
MoveNet	98.67	98.91	98.6	98.4	98.2	98.5



(a) MLP (b) DNN

(c) CNN (d) Performance metrics

Figure 8

Models Accuracy and Loss

6. Conclusion

In the proposed article, we investigated the implementation of an AI-powered yoga trainer using human activity recognition (HAR) and pose estimation. With the increasing screen time and people being glued to electronic devices, it is even more concerning that physical health is something that is neglected tremendously, with this project we tried to solve the problem. By leveraging TensorFlow's MoveNet model and real-time pose detection, we developed a system capable of detecting and evaluating yoga poses from a webcam feed. We did a comprehensive analysis on many algorithms and concluded Convolutional Neural Networks (CNNs) as the best option, as it is efficient in working on spatial features, which is required for image-based tasks like pose detection. MoveNet is an extremely fast and precise model that identifies 17 key points on a body and based on CNN was chosen to enhance real-time performance. The proposed AI-powered yoga trainer will make a significant advance in fitness technology by offering people a flexible and customized method of maintaining their physical and mental health. The user-friendly interface and real-time response open new possibilities for the home fitness, healthcare, and wellness sectors.

References

[1] J. Omotoyosi Adeyemi, "'Enhancing Elderly Wellness through AI-Powered Yoga and Exercise Support Systems,'" *Biomed. J. Sci. Tech. Res.*, vol. 54, no. 4 (Jan. 2024), doi: 10.26717/BJSTR.2024.54.008579.

[2] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, *Human activity recognition in artificial intelligence framework: a narrative review*, no. 0123456789. Springer Netherlands (2022).

[3] H. Ding, S. R. K. Bran, J. A. Paradiso, and H. W. M. H. Arif, "FEMO: A platform for free-weight exercise monitoring with RFIDs," SenSys 2015 - Proc. 13th ACM Conf. Embed. Networked Sens. Syst., pp. 141–154 (2015), doi: 10.1145/2809695.2809708.

[4] R. R. Kanase, A. N. Kumavat, R. D. Sinalkar, and S. Somani, "Pose Estimation and Correcting Exercise Posture," *ITM Web Conf.*, vol. 40, p. 03031 (Aug. 2021), doi: 10.1051/itmconf/20214003031.

[5] W. Qi and A. Aliverti, "A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity,"

IEEE J. Biomed. Heal. Informatics, vol. 24, no. 8, pp. 2199–2207 (2020) doi: 10.1109/JBHI.2019.2963048.

[6] A. Shah, W. Patel, and H. Koyuncu, "Empowering healthcare innovation : IoT-enabled smart systems and deep learning for enhanced diabetic retinopathy in the telehealth landscape," *J. Interdiscip. Math.*, vol. 27, no. 2, pp. 355–367 (2024), doi: 10.47974/JIM-1836.

[7] T. Singh and D. K. Vishwakarma, "A deeply coupled ConvNet for human activity recognition using dynamic and RGB images," *Neural Comput. Appl.*, vol. 33, no. 1, pp. 469–485 (2021), doi: 10.1007/s00521-020-05018-y.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and D. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691 (2017), doi: 10.1109/TPAMI.2016.2599174.

[9] A. Abobakr, M. Hossny, and S. Nahavandi, "A skeleton-free fall detection system from depth images using random decision forest," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2994–3005 (2018), doi: 10.1109/JSYST.2017.2780260.

[10] M. Sivalakshmi, K. R. Prasad, and C. S. Bindu, "Convolutional-based variational autoencoders for face privacy protection in video surveillance," *J. Discret. Math. Sci. Cryptogr.*, vol. 27, no. 4, pp. 1205–1214 (2024), doi: 10.47974/JDMSC-1975.

[11] B. Jo and S. Kim, "Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices," *Trait. du Signal*, vol. 39, no. 1, pp. 119–124 (Feb. 2022), doi: 10.18280/ts.390111.

Received August, 2024