YogaHelp: Leveraging Motion Sensors for Learning Correct Execution of Yoga With Feedback

Ashish Gupta and Hari Prabhat Gupta

***Abstract*—By leveraging motion sensors, the study of physical activities has become effortless and convenient. Yoga is an excellent form of physical activity or exercise that involves all parts of the body. Regular practice of yoga improves both the physical and mental health of the people, if the practitioner performs it correctly. Though the existing exercise recognition methods have shown quite a good performance using deep neural networks, they fail to assess the correctness of execution. Feedback describing correctness level is very useful to an amateur practitioner. In this article, we aim to develop a YogaHelp system to help amateurs for learning the correct execution of yoga without any supervision of a trainer. The importance of such a system is much higher during a pandemic situation, where the trainers can not be availed due to the risk of catching an infection. YogaHelp leverages the motion sensors including accelerometer and gyroscope, to recognize 12 linked-steps of sun salutation (Surya Namaskar) yoga along with their correctness level. YogaHelp employs a deep learning model built using convolutional layers for yoga step recognition without explicit feature extraction. The novelty of the system lies in the feedback that describes the speed of execution and angular deviation of the posture from the standard ones. We develop a prototype for collecting a training dataset from eight professional yoga trainers and for validating the effectiveness of the system.**

***Impact Statement*—Recently, yoga has received an unprecedented popularity all over the world. It is a wonderful way of performing physical exercise at home. Surya Namaskar is also a form of Yoga that incorporates a sequence of 12 linked steps and involves stretching of almost every part of the body. With the ubiquity of motion sensors, monitoring the execution of yoga steps has become convenient by collecting motion data. We develop a novel YogaHelp system that leverages the motion data for recognizing the different steps and for identifying how correctly they were executed by the practitioner. YogaHelp facilitates the amateur practitioners to learn the correct execution of yoga without any trainer within four weeks.**

***Index Terms*—Motion sensors, physical exercise, sun salutation, yoga.**

I. Introduction

PHYSICAL exercise is an important daily routine activity for human beings as it offers numerous health benefits including a strong immune system, healthy muscles, and improved cardiovascular fitness. In the pandemic scenario (*e.g.,* COVID-19), regular exercise becomes essential for strengthening the immune system of the people and consequently reducing the chance of catching an infection. As most outdoor physical activities (*e.g.,* gym workout, sports) get restricted during the pandemic, people are advised to perform in-home exercises but without any supervision of a trainer. Yoga is an excellent form of in-home exercise that not only strengthens the muscles but relaxes the mind also. As yoga involves stretching multiple body parts together to attain the right posture, amateur practitioners may find it difficult to follow by merely imitating a video. A monitoring system is required to help the practitioners by informing them about correct posture.

With the ubiquity of sensors in smartphones and wearable, physical exercises or activities can be monitored on a single tap by recording the body movements of the user with an accelerometer and gyroscope [1]–[3]. Fig. 1 illustrates an example of an exercise or yoga recognition system, where motion sensors are attached to different parts of the user's body and transmit the data to a smartphone when the user performs the yoga steps. The data are later transmitted to a cloud server, which identifies the steps using a recognition model and computes their correctness level. Finally, a feedback report is sent back to the user. Keeping the health benefits of regular exercise in mind, the researchers have put their efforts on developing physical activity recognition systems using smartphone sensors [3], [4], wearable [2], [5]–[7], WiFi signals [8], [9], and radio frequency (RF) signals [10]. Apart from that some prior studies have focused on recognizing body posture while performing yoga steps [11]–[15], squat [16], arm motion [17], [18], and playing badminton [19]. Though the recognition accuracy was claimed quite high in the existing work, the assessment of how correctly the activities was executed is not done, and as a consequence the amateur practitioners may be reluctant to adopt such solutions.

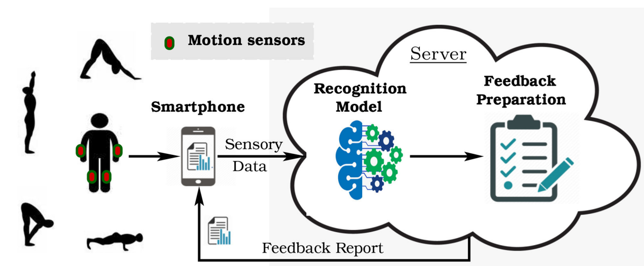


Fig. 1. Illustration of a yoga step recognition system with feedback of correctness using motion sensors.

Manuscript received March 10, 2021; revised May 12, 2021; accepted June 26, 2021. Date of publication July 9, 2021; date of current version October 13, 2021. This paper was recommended for publication by Associate Editor Qinglai Wei upon evaluation of the reviewers' comments. *(Corresponding author: Ashish Gupta.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, Varanasi 221005, India (e-mail: ashish.gupta582@gmail.com; hariprabhat.cse@iitbhu.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAI.2021.3096175.

Digital Object Identifier 10.1109/TAI.2021.3096175

A. Motivation

This work is motivated by the limitations of existing work.

1) *Correctness of execution:* Although the previous studies [2], [5], [12], [14] can recognize exercise or yoga step quite accurately, they do not assess how correctly the steps were executed. Merely imitating the steps from the video could help in practicing yoga at home but it can not inform the user about the correctness of execution. Incorrect execution may result in severe muscular injuries. Thus, a yoga recognition system should incorporate a mechanism to assess the correctness level.
2) *Feedback Score:* It helps in-home exercise or yoga practitioners (trainees) to improve upon the way of execution without any supervision of a trainer. Though a few existing work including [11], [16], [18] have attempted to provide feedback to the user, they do not consider the correctness level. Moreover, their feedback report is not informative enough to an amateur practitioner.
3) Along with the above limitations, the camera-based solutions [12], [13], [20] can identify the exercise steps only if a trainee performs them in predefined settings with a specific background, light, *etc.* On the other hand, an expensive setup is also required in RFID and WiFi signals based solutions [9], [10]. Such solutions impose cost and adaptability related issues on the user.

While addressing the above limitations, we present a YogaHelp system for practicing yoga (*e.g.,* Sun Salutation) correctly without any supervision of a trainer. Particularly, we address the problem: *how to recognize yoga steps with correctness level of their execution by leveraging motion sensors?*

B. Major Contributions

This is the first work toward assessing the correctness level of yoga execution. We make the following major contributions:

1) We propose a novel YogaHelp system to monitor and assess the correctness level of linked-steps of the yoga (i.e., Sun Salutation). YogaHelp leverages the motion data obtained from accelerometer and gyroscope, to recognize the steps and assess their correctness level.
2) The system builds a deep learning based recognition model that learns from a labeled dataset collected from professional yoga trainers. We also design two meaningful parameters for describing the correctness level of execution. YogaHelp computes the parameters based on the speed of step execution and angular deviation of current posture against the trainer's ones.
3) We incorporate a feedback mechanism to YogaHelp system to inform user about the correctness level of each step independently along with an overall score (in percent) after complete execution of each round of the yoga.
4) Finally, we validate YogaHelp system on a well known Sun Salutation (*Surya Namaskar*) yoga consisting of 12 linked-steps that are performed in a single round. The experimental results show a substantial improvement ($\sim$ 20%) in overall correctness score within four weeks.

The remainder of this article is organized as follows. Next section discusses the existing related work. Section III presents an overview of YogaHelp system. Section IV describes the training dataset collection and Section V proposes the deep learning based recognition model. Computation of step correctness parameters and feedback score are detailed in Sections VI and VII, respectively. We evaluate the proposed system in Section VIII and concludes this this in Section IX.

II. Related Work

Physical activity recognition has been a well-researched topic in the last two decade [21]. Some of the existing work that are closely related to our work are discussed as follows.

A. Physical Activity Recognition

Literature indicates a large body of work on physical activity recognition. Based on the data used for the recognition, we divided the existing work into two groups.

First group leverages the motion sensors (*e.g.,* accelerometer and gyroscope) of wearable or smartphone. The work in [5] designed an inertial measurement unit (IMU) based smart earring to motivate the users for regular exercise. A sensor fusion model, called as SenseHAR, is developed in [1] to recognize basic locomotion activities regardless of the sensors' locations on body parts, sampling rates, and their availability. SenseHAR exploited deep learning techniques to map the raw sensory data to a shared latent space, where different devices build their own fusion model for identifying human activities. The work in [2] developed an unsupervised learning approach for weight training exercises using wearable sensors. A hybrid two-layer framework is introduced in [6] to recognize free weight exercises and count their repetitions. The first layer uses a support vector machine to distinguish free weight exercises from nonfree weight while the second layer uses a neural network for identifying different free weight exercises. Recognizing complex human activities such as sitting on sofa and talking while standing, has also drawn the attention of researchers [3], [4]. Further, the information about routine exercises can also be used to assess the quality of sleep by using generalized boosted regression models [22].

Another group of solutions exploit other modalities such as WiFi signals, RF signals, and visual information captured through a depth camera. In the work [23], an RF-identification (RFID) based platform is developed for monitoring a variety of free-weight exercises such as concentration biceps curl, seated triceps press, *etc.* This work employed dynamic time warping for recognizing the exercises using their Doppler profiles. The authors in [8] used channel state information of WiFi signals for monitoring human dynamics such as people's walking direction and speed in indoor venues. The authors in [9] also leveraged WiFi signals to recognize indoor human activities with high accuracy. Physical activities can also be recognized quite accurately with depth cameras [20], [24] by exploiting the capabilities of deep learning techniques.

B. Posture Recognition

Identifying the posture or arrangement of body skeleton is an active area of research in ubiquitous computing. Some recent related work is discussed as follows. In [16], the authors

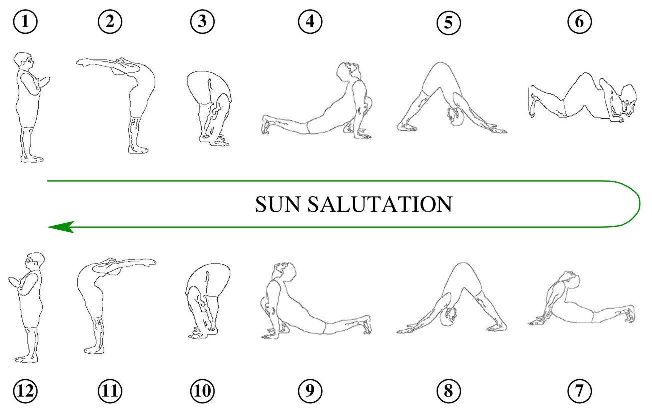


Fig. 2. Illustration of 12 linked-steps of Sun Salutation that need to be performed in the mentioned order.

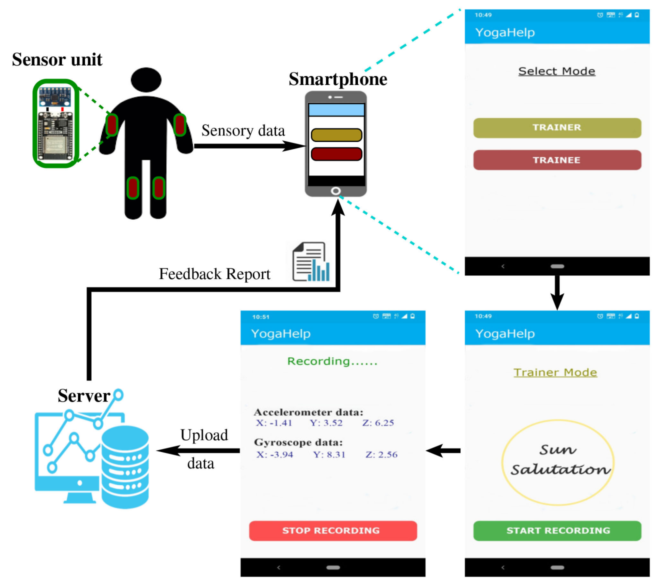


Fig. 3. Prototype of YogaHelp system using sensor units and android application for smartphone.

developed a deep learning based approach for squat posture recognition using IMUs. A camera-based solution is proposed in [25] to assess the consistency of steps of Sun Salutation using hidden Markov models. Recently, the authors in [11] attempted to assess the correctness of yoga posture using several IMUs placed on the user's body. The assessment is done based on the angular difference from currently attained posture to the standard one (learned from yoga trainers) for different body parts. The work [15] developed a wireless network of infrared and thermal sensors to construct an image of yoga posture being performed. The image is later given as input to a deep learning based image classifier to recognize the type of posture in a noninvasive manner. Jain *et al.* [14] leverages smartphone camera for yoga posture recognition using a convolutional neural network. Some studies [12], [13] have also exploited deep neural networks for posture recognition in different home environments.

III. YogaHelp System

While developing YogaHelp system, we mainly focus on *Sun Salutation* yoga as it is an excellent form of exercise for strengthening the whole body muscles and relaxing the mind. One round of Sun Salutation consists of 12 powerful linked-steps (Asanas) that are necessarily to be performed in a particular sequence, as shown in Fig. 2. In YogaHelp system, we focus on providing a detailed feedback describing each step of the Sun Salutation with its correctness level of execution. This work uses the terms “yoga” and “Sun Salutation” interchangeably.

A. Prototype

As Sun Salutation brings the motion in both parts (*i.e.,* lower and upper limb) of body, we require multiple sensor units to record the body movements precisely. YogaHelp system uses four sensor units that are attached to different parts of body as shown in Fig. 3. Each sensor unit consists of a NodeMCU module (ESP32) and two motion sensors including accelerometer and gyroscope. NodeMCU module works as controller and can transmit sensory data through Bluetooth. Accelerometer and gyroscope are availed from nine-axis IMU sensor (MPU9250) and their sampling rate is set to 50 Hz. Further, we develop an android application to facilitates the data collection on user's smartphone. The application receives the motion data through Bluetooth and transfers it to a cloud server (personal computer) once the whole round is executed. The android application works in following modes:

1) *Trainer Mode*: It is selected by the trainer during training data collection. As soon as trainer presses the start button, the smartphone starts recording the samples in a local file till the *stop recording* button is not pressed. Once one round of Sun Salutation is completed, the trainer presses the *stop recording* button, which will automatically transfers the data file to the server. This process is repeated for each round of Sun Salutation.
2) *Trainee Mode*: It is used by the trainees, who wish to practice yoga without the supervision of trainer. Similar to the trainer mode, the trainee can start and end the Sun Salutation by pressing the *start recording* and *stop recording* button of the application, respectively, and the data are transferred to the server once the current round is completed. At the server, the recognition model identifies each step of the yoga. In addition, a feedback report is prepared to describe the correctness level, which is sent back to the trainee's smartphone, as shown in Fig. 3.

B. Overview of the System

YogaHelp system consists of four major components. This section presents an overview of each component and also provides a list of notations in Table I for better understanding of the work.

1) Training Data Collection: At first, we collect the training data from eight professional yoga trainers including six males and two females. The data are collected via smartphone from four sensor units that are strapped on following body parts: left forearm, right forearm, left leg, and right leg. The collected data are processed at the server to create a training dataset, where each instance is a single step of the yoga.

TABLE I
NOTATIONS USED IN THIS WORK

Symbol	Description
$\mathbf{X}$	24-dimensional time series
$\mathbf{D}$	Dataset with instances of type $\mathbf{X}$
D_i	Dataset with only i^{th} dimension time series
$\mathcal{X}^j$	j^{th} instance of D_i
$\mathcal{Y}^j$	Class label associated with $\mathcal{X}^j$
N	Number of instances in D_i
$\boldsymbol{\mathcal{X}}$	Set of N instances of D_i without class labels
$\boldsymbol{\mathcal{Y}}$	Set of N class labels of D_i
m	Length of an instance in D_i
k	Total number of classes in D_i
$\mathbf{Y}$	Set of k class labels
y_c	c^{th} class label in $\mathbf{Y}$
θ	Weight parameters of the recognition model
θ^*	Optimal set of weight parameters
$\mathcal{L}(\cdot)$	Loss function of the model
$\sigma(\cdot)$	Softmax function

2) Recognition Model: Given the training dataset, we develop a deep learning based model to recognize each step of the Sun Salutation independently. This work employs convolutional neural network for constructing the model.

3) Step Correctness Parameters: We estimate the correctness parameters for each step of the Sun Salutation by using the training dataset. The correctness parameters include speed of step execution and angular position of the sensor units once a new posture is attained.

4) Step Recognition and Correctness Feedback: When YogaHelp system works in trainee mode, the server first recognizes the steps by using the built recognition model and then prepares a feedback report based on the step correctness parameters. The feedback is finally sent to the trainee.

IV. TRAINING DATA COLLECTION

A primary step toward developing YogaHelp system is the training data collection by employing professional yoga trainers. This section describes the complete process of building the training dataset.

A. Data Collection

We interviewed eight professional yoga trainers (six males and two females) with an average age of 29 years and then collected the training data with their consent. The trainer first places the sensor units on body as shown in Fig. 3 and then performs the Sun Salutation yoga by selecting the *trainer* mode on the smartphone. For each round of Sun Salutation, the smartphone records a six-dimensional sequence of time stamped samples (corresponding to three-axis accelerometer and three-axis gyroscope) from each of the four sensor units. In total, a 24-dimensional time series is sent to the server. While performing the yoga, the trainer stays at each posture for around 5 s without any movement and thus the resultant time series also includes nonmovement samples. We use these samples to segment the time series into step wise subtime series.

During data collection, each trainer performed 20 rounds of Sun Salutation for 30 days, which generated total $20 \times 30 = 600$ time series. As the data are collected from eight trainers, the server received total $8 \times 600 = 4800$ time series by the end of 30 days. Each collected time series consists of 24 dimensions. Apart from these time series, we recorded a video of the trainer while performing the Sun Salutation. The video data are used for obtaining the ground truth of the steps.

B. Segmentation

We segment the time series into step wise subtime series and each of that subtime series becomes an instance of the training dataset. For segmentation, we utilize the fact that the fluctuations in sample values during body movement are much higher than nonmovement. As the trainer remains in each posture for around 5 s without any movement, the time series has almost steady values during nonmovement (or stable) period. However, we observed from the collected data that the stable period varies from 2 to 7 s between different steps. As Sun Salutation consists of 12 steps, the time series should contain 13 stable periods when one round is ended. The first and last stable periods occur before first step and after last step, respectively. If the samples taken during stable periods are removed then we obtain 12 disconnected segments, which correspond to 12 linked-steps. Now, the objective of the segmentation is to find the stable periods in a given time series for one round of Sun Salutation.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_{24}\}$ be a time series received from the four sensor units for a single round of Sun Salutation, where X_i is one-dimensional time series of length m, *i.e.*, $X_i \in \mathbb{R}^m$ for $1 \leq i \leq 24$. The complete process to obtain the stable periods in $\mathbf{X}$ is as follows.

1) At first, we divide the time series $\mathbf{X}$ into n nonoverlapping windows, where n is equal to m/l and l denotes the length of window.
2) For $i = 1$ to n:
   a) Compute variance of ith window as $\mathrm{Var}(\mathbf{w}_i) = \{\mathrm{Var}(w_{i1}), \mathrm{Var}(w_{i2}), \ldots, \mathrm{Var}(w_{i24})\}$, where $\mathrm{Var}(w_{ij}) = \frac{1}{l}\sum_{k=1}^{l}(\overline{w_{ij}} - w_{ij}[k])^2$ with $1 \leq j \leq 24$ and $\overline{w_{ij}}$ denotes the mean value.
   b) If $\mathrm{Var}(w_{ij}) < \theta$ for all j then mark $\mathbf{w}_i$ window as stable, where θ is the maximum allowable variance threshold and it is learned empirically from the given time series.
3) Considering the trainer stay in one posture for 2–7 s and window length $l = 10$, we need to locate 10–35 consecutive stable windows to make a stable period, because 50 samples were collected per second. We now identify the stable periods as follows.
   a) Find at least 10 consecutive windows, *i.e.*, $\mathbf{w}_i, \mathbf{w}_{i+1}, \ldots, \mathbf{w}_{i+9}$, that are marked as stable.
   b) If found then we identify a stable period from $\mathbf{w}_i$ to $\mathbf{w}_{i+s}$, where $\mathbf{w}_{i+s}$ is the last stable window in the continuation and $s \geq 9$.
   c) Search for next stable period between $\mathbf{w}_{i+s+1}$ to $\mathbf{w}_n$, by using steps $3(a) - 3(c)$.

Finally, if the number of stable periods is equal to 13 then we remove them from the time series $\mathbf{X}$, which in turn provides 12 disconnected subtime series (segments).

C. Training Dataset

Once segmentation is done for all the collected time series, we create a dataset $\mathbf{D}$ by storing each segment as an instance

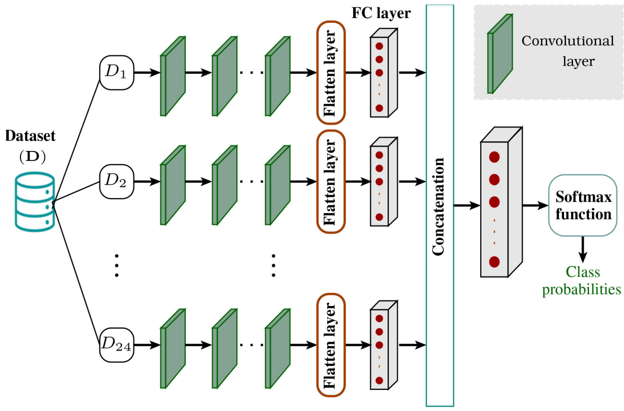


Fig. 4. Overview of the yoga step recognition model.

Algorithm 1: Recognition model.

Input: Training dataset $\mathbf{D}$;
Output: Optimal set of weight parameters θ^*;
1 Divide dataset $\mathbf{D}$ into 24 sub-datasets D_i, where $1 \le i \le 24$.
/* θ denotes a set of weight parameters for all the layers */
2 Initialize θ with random weights.
3 Prob_dist = [].
4 **for** $j \leftarrow 1$ *to* N **do**
  /* For each D_i perfom Steps $5-6$ in parallel using θ */
  /* $\mathcal{X}^j$ is an instance of D_i */
5   Op_conv$_i$ $\leftarrow$ Pass $\mathcal{X}^j$ via 3-convolutional layers.
6   Op_FC$_i$ $\leftarrow$ Pass Op_conv$_i$ via a flatten and FC layer.
7   Op_concate $\leftarrow$ Concate$_{1\le i\le 24}$(Op_FC$_i$).
8   Op_FC $\leftarrow$ Pass Op_concate via another FC layer.
9   Probabilities $=$ $Softmax$(Op_FC) using Eq. 2.
10   Prob_dist $\leftarrow$ $Append$(Probabilities).
11 Compute accuracy loss using Eq. 3 and update θ.
12 $\theta^* \leftarrow$ Minimize loss by repeating Steps $4-12$.
13 **return** θ^*.

---

along with its step number (label). During segmentation, 40 out of 4800 time series were discarded due to nonavailability of 13 stable periods and thus we get total $12 \times 4760 = 57120$ labeled instances in $\mathbf{D}$. Each instance is also a 24-dimensional subtime series (segment) corresponding to one particular step of Sun Salutation. We found that the length of instances vary from 80 to 130 samples. To use this dataset for training the model, we make the length of all instances equal to 130 by padding mean values in the shorter instances.

V. Recognition Model

Given training dataset, this section constructs a recognition model that can recognize each individual step of the Sun Salutation yoga when performed by a trainee. This work employs deep neural networks to construct the recognition model. The reason for choosing the deep neural networks is their capability of extracting the most suitable features automatically from the raw data [26]. An overview of the recognition model is shown in Fig. 4.

As the dataset $\mathbf{D}$ contains 24-dimensional instances, we first split it into 24 datasets, denoted by $D_1, D_2, \ldots, D_{24}$, where each D_i consists of the samples of one particular axis of the used sensor and $1 \le i \le 24$. Each of this dataset is passed through a set of convolutional layers (connected sequentially) followed by a flatten and a fully connected (FC) layer, as shown in Fig. 4. Later, the resultants of FC layers are concatenated and passed through another FC layer followed by a *softmax* function that returns a set of class probabilities for all the steps of Sun Salutation.

In the model, all the convolutional layers are one-dimensional with equal number of filters and an input shape $(1, m)$, where m denotes the length of the instance ($m = 130$ in particular). This work uses 12 neurons at each FC layer to obtain 12 features as the total number of class labels (steps) in the dataset $\mathbf{D}$ is 12.

Now, we present the mathematical formulation of the model. Let $D_i = \{\mathcal{X}, \mathcal{Y}\} = \{\mathcal{X}^j, \mathcal{Y}^j\}_{j=1}^{N}$ be a training dataset, where $\mathcal{X}^j$ denotes a time series instance with its class label $\mathcal{Y}^j$. Let the dataset contain total k class labels, denoted as $\mathbf{Y} = \{y_1, y_2, \ldots, y_k\}$. For each dataset D_i, the objective of recognition model is to learn optimal set of weight parameters by minimizing a loss function $\mathcal{L}(\cdot)$ as

$$\theta^* = \underset{\theta}{\text{argmin}}\ \{\mathcal{L}(\mathcal{Y}, \sigma_{\mathbf{Y}}(\theta\mathcal{X}))\} \tag{1}$$

where $\sigma_{\mathbf{Y}}(\theta\mathcal{X}) = p(\mathbf{Y}|\mathcal{X}, \theta)$ and $\sigma(\cdot)$ is a *softmax* function that transforms a vector into probability distribution over its elements. For a vector $z \in \mathbb{R}^k$, $\sigma(\cdot)$ is given as

$$\sigma_c(z) = \frac{\exp(z_c)}{\sum_{c=1}^{k}\exp(z_c)} = p(c|z) \quad \forall c \in \{1, 2, \ldots, k\}. \tag{2}$$

This work employs cross-entropy loss function as it is widely adopted for solving recognition problems [26]. The loss function $\mathcal{L}(\cdot)$ for the dataset D_i, is given as

$$\mathcal{L}\left(\mathcal{Y}, \sigma_{\mathbf{Y}}(\theta\mathcal{X})\right) = -\frac{1}{N}\sum_{j=1}^{N}\sum_{c=1}^{k}\mathbb{I}(\mathcal{Y}^j, y_c)\log p\left(\mathcal{Y}^j = y_c|\mathcal{X}^j, \theta_c\right) \tag{3}$$

where $\mathbb{I}(\mathcal{Y}^j, y_c) = 1$ if $\mathcal{Y}^j = y_c$ and 0 otherwise; and θ_c denotes the model parameters for y_c class label. As the cross-entropy loss function is convex and differentiable [26], the recognition model applies stochastic gradient decent (SGD) to solve the optimization problem (defined in Eq. 1) to get an optimal set of weight parameters θ^*. With θ^*, the model becomes ready to predict the class label of a time series instance. The steps of building the recognition model are illustrated in Algorithm 1.

VI. Step Correctness Parameters

This section estimates correctness parameters for each step of Sun Salutation by using the training dataset. This work considers two parameters, duration of step execution and angular position of the sensor units, for indicating the correctness level. These parameters are used to assess the correctness of step execution when a trainee performs the Sun Salutation.

A. Duration of Step Execution

The time period (duration) that an user takes to execute a single step, is an important parameter for measuring the speed

TABLE II

DURATION OF STEP EXECUTION AND ANGULAR POSITION OF THE SENSOR UNITS FOR DIFFERENT STEPS OF SUN SALUTATION OBTAINED USING TRAINING DATASET. [LW: LEFT WRIST, RW: RIGHT WRIST, LC: LEFT CALF, RC: RIGHT CALF]

Step No. (class label)	**Length of instances** $[L_{min} - L_{max}]$	**Duration** $[d_{min} - d_{max}]$	**Position on LW** $[\rho, \varphi, \theta]$	**Position on RW** $[\rho, \varphi, \theta]$	**Position on LC** $[\rho, \varphi, \theta]$	**Position on RC** $[\rho, \varphi, \theta]$
1	$80 - 90$	$1.6 - 1.8$	$30, 110, 210$	$60, 90, 200$	$45, 30, 170$	$20, 80, 100$
2	$80 - 120$	$1.6 - 2.4$	$250, 40, 30$	$50, 120, 330$	$45, 120, 140$	$30, 60, 310$
3	$100 - 130$	$2.0 - 2.6$	$50, 40, 115$	$70, 30, 10$	$40, 125, 210$	$40, 55, 90$
4	$100 - 120$	$2.0 - 2.4$	$65, 80, 190$	$20, 45, 80$	$30, 215, 100$	$50, 90, 65$
5	$100 - 120$	$2.0 - 2.4$	$60, 75, 90$	$135, 60, 30$	$30, 75, 90$	$110, 210, 180$
6	$90 - 130$	$1.8 - 2.6$	$80, 95, 60$	$75, 60, 85$	$110, 115, 20$	$45, 75, 110$
7	$80 - 110$	$1.6 - 2.2$	$200, 110, 80$	$110, 115, 60$	$120, 35, 70$	$75, 60, 85$
8	$80 - 120$	$1.6 - 2.4$	$65, 80, 190$	$20, 45, 80$	$30, 215, 100$	$50, 90, 65$
9	$90 - 110$	$1.8 - 2.2$	$60, 75, 90$	$135, 60, 30$	$30, 75, 90$	$110, 210, 180$
10	$90 - 120$	$1.8 - 2.4$	$50, 40, 115$	$70, 30, 10$	$40, 125, 210$	$40, 55, 90$
11	$110 - 130$	$2.2 - 2.6$	$250, 40, 30$	$50, 120, 330$	$45, 120, 140$	$30, 60, 310$
12	$80 - 100$	$1.6 - 2.0$	$30, 110, 210$	$60, 90, 200$	$45, 30, 170$	$20, 80, 100$

of step execution. The duration of step execution helps to know whether the user is performing the step at a right speed or not. If the user is performing the steps too fast then it may cause severe muscular pain. On the other hand, if the steps are executed too slowly then the user may not get intended benefits of the yoga. We use the training dataset for estimating the suitable duration of step execution.

Table II illustrates the length of instances and the duration of execution (in seconds) for the steps of Sun Salutation. The length of instances is obtained from the training dataset and the duration is calculated as $\frac{\text{length}}{50}$ seconds, where 50 is the sampling rate of the sensors. Minimum and maximum length of the instances are denoted by $L_{\min}$ and $L_{\max}$, respectively. Similarly, $d_{\min}$ and $d_{\max}$, respectively, denote the minimum and maximum duration of step execution.

B. Angular Position of Sensor Units

By executing a step, the user basically switches from current posture to next. Once the next posture is attained, it is important to know that how correctly the posture is attained by the user. Such correctness can be quantified by estimating the position of the sensor units that are strapped on the user's body. In this work, the position of each unit is given in terms of tilt angles (pitch ρ, roll φ, yaw θ) corresponding to the three axes of accelerometer. The tilt angles can be computed by using three axes of the accelerometer samples.

For the dataset $\mathbf{D}$, we first separate out the instances of one particular step and then consider only three-dimensions, which are obtained from a particular accelerometer. For each three-dimensional instance, we first remove the padding from the instance and then consider only its last sample for computing the tilt angles (pitch, roll, yaw). We chose only the last sample of the instance as it can correctly capture the user's posture. Later, the position of the sensor unit for one particular step is given by the average tilt angles, where averaging is done over all the instances belonging to that step. Algorithm 2 illustrates the complete process for estimating the angular position. The angular positions of all the units are shown in Table II. As the position of sensor units is obtained using the trainer's data, we use them as reference (or standard) for quantifying the correctness of the trainee's posture during yoga execution.

Algorithm 2: Estimating position of a sensor unit.

Input: Training dataset $\mathbf{D}$;
Output: Position of a sensor unit for all 12 steps;
for $i \leftarrow 1$ *to* 12 **do**
  /* N_i is the number of instances for i^{th} step (class) */
  $\mathbf{P} \leftarrow []$
  for $j \leftarrow 1$ *to* N_i **do**
    /* Let $A^j \in \mathbf{D}$ be a 3-dimensional instance obtained from the accelerometer for i^{th} step */
    Remove padding of A^j.
    Find last sample of A^j as (A_x, A_y, A_z).
    /* Compute tilt angles */
    Pitch $(\rho) = tan^{-}\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right)$.
    Roll $(\varphi) = tan^{-}\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right)$.
    Yaw $(\theta) = tan^{-}\left(\frac{\sqrt{A_x^2 + A_y^2}}{A_z}\right)$.
    Position $P_j = (\rho, \varphi, \theta)$.
    $\mathbf{P} \leftarrow Append(P_j)$.
  /* Position of the sensor unit for i^{th} step */
  $(\rho, \varphi, \theta)_i = Average(\mathbf{P})$.
  $(\boldsymbol{\rho}, \boldsymbol{\varphi}, \boldsymbol{\theta}) \leftarrow Append((\rho, \varphi, \theta)_i)$.
return $(\boldsymbol{\rho}, \boldsymbol{\varphi}, \boldsymbol{\theta})$.

VII. STEP RECOGNITION AND CORRECTNESS FEEDBACK

Given the recognition model and step correctness parameters, YogaHelp system can recognize the steps of Sun Salutation and their correctness level for a trainee. In this section, we discuss the working of YogaHelp application for trainees. At first, the trainee needs to wear four sensor units (as shown in Fig. 3) and start recording the data through YogaHelp application on the smartphone by selecting the trainee mode. When one round is completed, the trainee should stop the recording. The recorded data are transferred to the server from the smartphone. Later, the server performs following steps:

1) Segmentation: The collected time series are segmented by using the segmentation process discussed in Section IV-B.

2) Step Recognition: Next, the obtained segments are given as input to the recognition model to predict a class label (step number) for each of the segments.

3) Correctness Feedback: Finally, the server prepares a feedback report for each predicted step by using the estimated

correctness parameters and sends it to the trainee's smartphone. The feedback describes each predicted step s based on following two parameters:

a) *Speed of Execution:* We identify the speed of execution of any particular step with respect to its reference d_{min} and d_{max} values (see Table II). Let d' denote the duration of step s. YogaHelp provides the speed of execution as

$$\text{Speed of execution} = \begin{cases} \text{Too slow}, & \text{If } d' < d_{\text{min}} - 1 \\ \text{Too fast}, & \text{If } d' > d_{\text{max}} + 1 \\ \text{Correct}, & \text{Otherwise.} \end{cases}$$

YogaHelp system allows the trainee to increase or decrease the duration of step execution up to one second only. If the duration goes beyond one second then the step was either executed too slow or too fast.

b) *Deviation of the Posture:* We first compute the position of the four sensor units by using the segment (time series) of step s as discussed in Section VI-B. Let ρ', φ', and θ' be the computed tilt angles that represent the position of a sensor unit after executing the step s. Let ρ, φ, and θ be the reference tilt angles obtained from Table II for the same unit. Now, the deviation of the posture is given as

$$\text{Deviation} =$$

$$\begin{cases} \text{Acceptable}, & \text{If } (\rho', \varphi', \theta') \in (\rho \pm 5, \varphi \pm 5, \theta \pm 5) \\ \text{Low}, & \text{If } (\rho', \varphi', \theta') \in (\rho \pm 10, \varphi \pm 10, \theta \pm 10) \\ \text{Moderate}, & \text{If } (\rho', \varphi', \theta') \in (\rho \pm 15, \varphi \pm 15, \theta \pm 15) \\ \text{High}, & \text{If } (\rho', \varphi', \theta') \in (\rho \pm 20, \varphi \pm 20, \theta \pm 20) \\ \text{Wrong}, & \text{Otherwise.} \end{cases}$$

Based on the conducted experiments we know that if the trainee is executing the step correctly then the posture should not be deviation more 5° from the reference posture in all three angles. Thus, YogaHelp system allows the deviation of the posture up to 5° in pitch, row, and yaw. However, if the deviation is more than 20°, the trainee might attained a wrong posture.

Overall Correctness Score (OCS): Along with above two parameters, YogaHelp system also provides an OCS for the whole round of Sun Salutation performed by the trainee. While computing OCS, we consider two cases:

1) If number of segments in the collected time series is exactly 12 then OCS is computed as

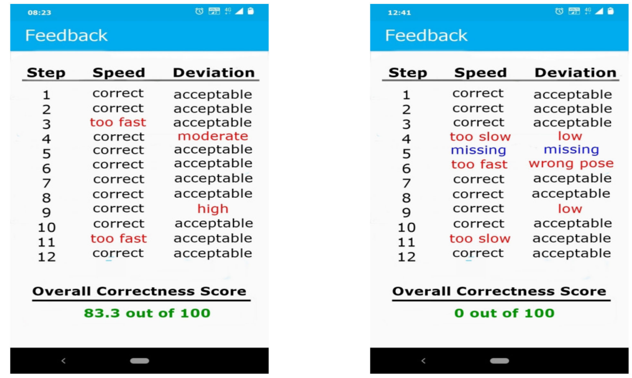
$$\text{OCS} = \frac{0.5 \times a + 0.5 \times b}{12} \times 100 \quad (4)$$

where a denotes the number of steps that are executed at *correct* speed and b denotes the number of steps that have *acceptable* deviation of the postures. A feedback screenshot of the YogaHelp application is shown in part (a) of Fig. 5.

2) If number of segments is not equal to 12 then OCS is set to 0, indicating the wrong execution of the whole round, as shown in part (b) of Fig. 5.

VIII. Evaluation

In this section, we carried out several experiments to evaluate the performance of the proposed YogaHelp system. Essentially, we discuss the performance evaluation metrics, implementation, and parameter settings of the recognition model, and experimental results, in detail.



(a) (b)

Fig. 5. Screenshots of feedback report in YogaHelp system.

A. Performance Metrics

This work uses following performance metrics:

1) F_1 score: Let T_p^i, F_p^i, and F_n^i represent the true positive, false positive, and false negative, respectively, for ith class. Now, F_1 score of the recognition model is computed as $\frac{1}{k}\sum_{i=1}^{k}\frac{2\times T_p^i}{T_p^i+F_p^i+F_n^i}$, where k denotes the number of classes in the dataset.

2) Accuracy: It is computed as $\frac{1}{k}\sum_{i=1}^{k}\frac{T_p^i}{T_p^i+F_n^i}$.

3) OCS: This metric helps in determining the overall correctness of whole Sun Salutation round when performed by the trainee. The detailed description of OCS is covered in Section VII.

B. Implementation and Parameter Settings

YogaHelp system involves the implementation of a smartphone application and a deep learning based recognition model. The application is implemented in Android Studio version 4.0 and the recognition model is implemented in Python language using functional API of Keras. The model is built around the combination of convolutional and fully connected layer, as discussed in Section V. During the experiments, we set the loss function to "categorical_crossentropy," batch size to "200," number of convolutional layers to "3," filters per layer to "64," learning rate to "0.01," and epochs to "50," as these settings prove to be most effective among all combinations.

C. Results on Testing Data

This section presents the results on testing data using the trained recognition model with preset parameters. In particular, we attempt to answer following questions: 1) What is the step recognition accuracy of YogaHelp system for trainers and trainees? 2) How does the performance of trainees improve over time? 3) What is the data distribution in terms of speed of execution and angular deviation of the posture? 4) How does OCS feedback help in improving the step execution of trainees?

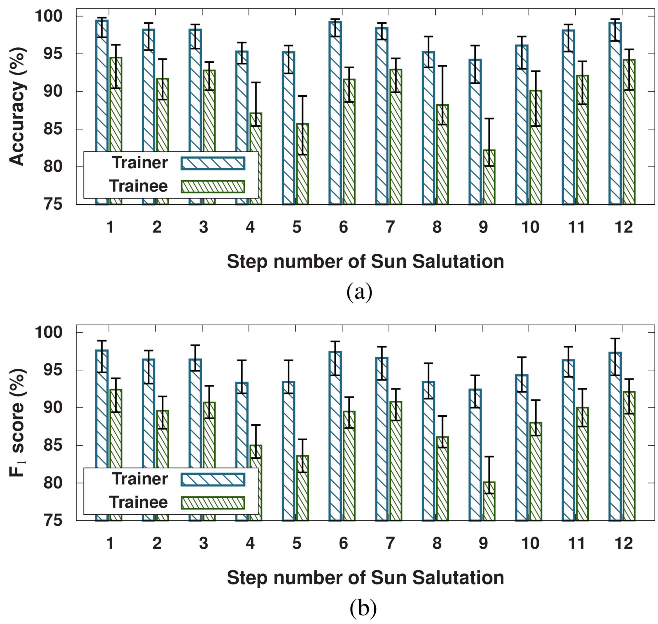


Fig. 6. Step recognition performance for trainers and trainees. (a) Accuracy. (b) F_1 score.

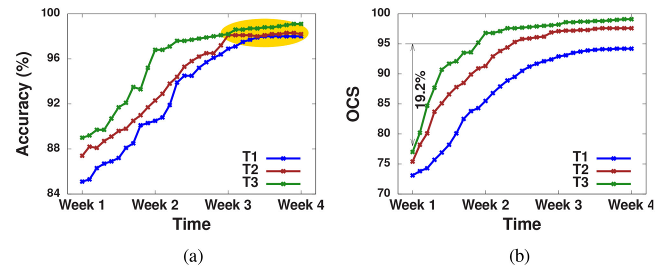


Fig. 7. Illustration of performance improvement of three different trainees (T1, T2, and T3) over a period of four weeks. (a) Accuracy. (b) Overall correctness score.

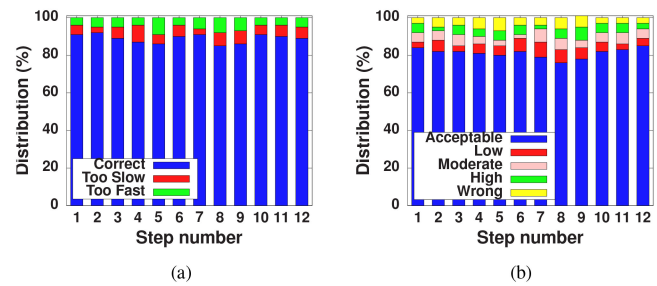


Fig. 8. Step-wise distribution of correctness level using test data collected from the trainees over a period of four weeks. (a) Distribution based on speed. (b) Distribution based on deviation.

5) What is the step recognition accuracy with fewer sensor units?
6) How better YogaHelp system is than state-of-art methods?

1) Step Recognition Accuracy and F_1 Score: At first, we evaluate the step recognition performance of YogaHelp system for trainers and trainees. Test data are collected for a period of four weeks from 20 participants, including 5 trainers and 15 trainees. Each participant is asked to perform at least 5 and at most 15 rounds of Sun Salutation every day. The collected test time series are segmented as discussed in Section IV-B. Fig. 6 illustrates the accuracy and F_1 score with which the system has recognized the steps for trainers and trainees. We observed that step 9 is quite hard to recognize by the built deep learning model as its time series contains fewer identifiable patterns than other steps. On the contrary, most of the steps are recognized with an accuracy more than 96% and 93% for trainers and trainees, respectively, as shown in part (a) of Fig. 6. Similarly, part (b) of Fig. 6 illustrates F_1 score for trainers and trainees, respectively. Further, we observe from the results that trainees can learn correct execution of the yoga steps with an average accuracy of 91.4% and F_1 score of 89.3% within four weeks.

2) Performance Improvement Over Time: Next, we investigate how the performance of trainees improve over time. We monitored three trainees with their consent for a period four weeks. Each of them was directed to perform a fixed number of rounds of Sun Salutation as follows: Trainee T1 – 5, Trainee T2 – 10, and Trainee T3 – 15 rounds. The recognition accuracy and OCS for different trainees are shown in parts (a) and (b) of Fig. 7, respectively. The results show that four weeks are sufficient to improve the overall correctness of yoga execution by using YogaHelp system. Further, it is easy to observe in part (b) that if the trainee performs more rounds each day then higher accuracy can be achieved early. The correctness of step execution has been greatly improved in four weeks only; for instance, from 77.3% to 98.9% for trainee T3. It clearly indicates the effectiveness of the feedback.

3) Evaluation of Step Correctness: Fig. 8 shows the distribution of testing rounds performed by the trainees over a period of four weeks. The distribution is reported in terms of speed of step execution and deviation of posture attained after complete execution. We observed that the trainees were able to perform the steps at *correct* speed for 90% of the total rounds, however, they experienced difficulty in attaining the *acceptable* posture. Besides that the trainees took *wrong* posture for 3%$\sim$7% of the total rounds.

4) Impact of OCS Feedback on Step Execution: We also conduct an experiment to analyze the impact of OCS feedback on the performance of steps performed by the trainee. The obtained results are reported in Fig. 9 using accuracy, F_1 score, variation in speed of execution, and angular deviation from the trainers' posture. The data used in this experiment are collected from 15 trainees over a period of four weeks. Part (a) of Fig. 9 shows a clear improvement of 11%$\sim$15% on the recognition accuracy if the feedback is given. Recognition accuracy for step number 9 is quiet low (*i.e.,* 72.6%) if the trainees do not know about how correctly they are executing the steps, and it increases to 83.5% with feedback, which indicates the effectiveness of the OCS feedback in YogaHelp system. A similar pattern of performance with minimal compromise is observed for F_1 score in part (b) of Fig. 9. Further, we also observe a substantial gain in the *correct* speed of execution and *acceptable* level of angular deviation of sensor unit positions, as depicted in part (c) and (d) of Fig. 9, respectively.

5) Using Partial Data: Finally, we present the accuracy and OCS results using the data of fewer sensor units not all four in Fig. 10. We take the data of two sensor units in the experiment

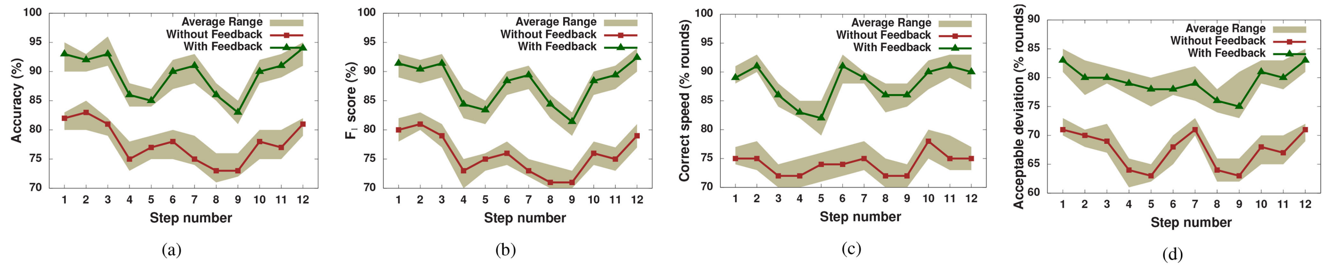


Fig. 9. Impact of OCS feedback on step execution of the trainees. (a) Impact on accuracy. (b) Impact on F_1 score. (c) Impact on speed. (d) Impact on deviation.

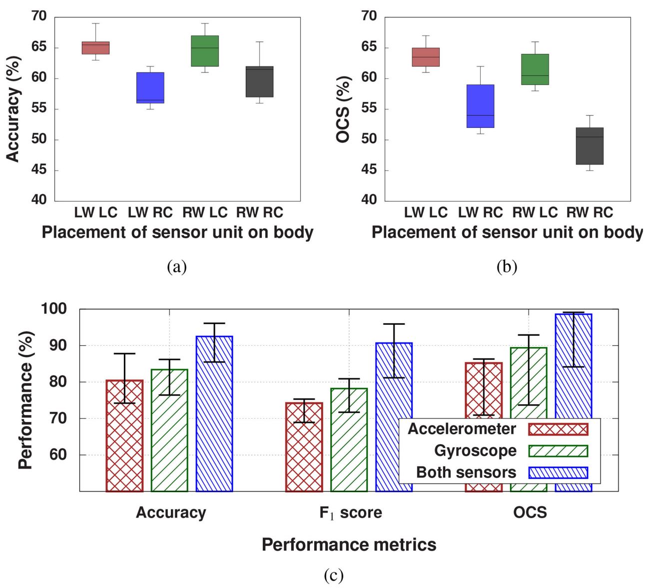


Fig. 10. Performance using the data of fewer sensors or units. [LW: Left Wrist, RW: Right Wrist, LC: Left Calf, RC: Right Calf]. (a) With two sensor unit. (b) With two sensor unit. (c) With data of one (Accelerometer or Gyroscope) and both sensors.

where four combinations are possible, as shown in parts (a) and (b) of Fig. 10. It is interesting to observe that any combination with the sensor unit placed on left calf (LC) gives better accuracy and OCS than other combinations. Although the median accuracy reaches to around 65% with the combination of sensor units placed on LW and LC, it is not reliable enough. Thus, YogaHelp system uses four sensor units to capture every movements of upper and lower limb while performing the steps of yoga. Next, part (c) of the figure illustrates the contribution of individual sensor (Accelerometer or Gyroscope) in the performance of the recognition model. We can clearly observe that the model performs better with Gyroscope data. Further, the model gains approximately 10%$\sim$15% on the all the performance metrics if it combines Gyroscope data with Accelerometer.

D. Comparison With Existing Work

We compare YogaHelp system with two state-of-art methods including SenseHAR [1] and HuMAn [3]. The comparison is carried out on step recognition accuracy and its improvement over four weeks, for trainers and trainees.

TABLE III
COMPARISON OF YOGAHELP SYSTEM WITH EXISTING WORK ON AVERAGE RECOGNITION ACCURACY (IN %)

Step No.	SenseHAR [1]		HuMAn [3]		YogaHelp	
	Trainer	Trainee	Trainer	Trainee	Trainer	Trainee
1	82.8	72.9	88.5	76.2	99.4	94.5
2	85.5	73.2	85.6	73.4	98.2	91.7
3	84.1	74.3	87.2	77.2	98.3	92.8
4	72.1	68.2	81.2	70.2	95.3	87.1
5	73.7	67.1	79.7	71.2	94.9	85.7
6	80.6	72.8	83.4	72.6	99.1	91.6
7	84.1	73.2	88.1	77.1	98.4	92.9
8	77.3	70.1	78.9	69.1	95.2	88.2
9	75.1	67.1	81.2	71.5	94.2	83.2
10	83.4	72.2	88.7	79.1	96.1	90.1
11	82.2	71.2	86.0	78.8	98.6	92.1
12	85.2	73.4	89.8	76.1	99.0	94.2

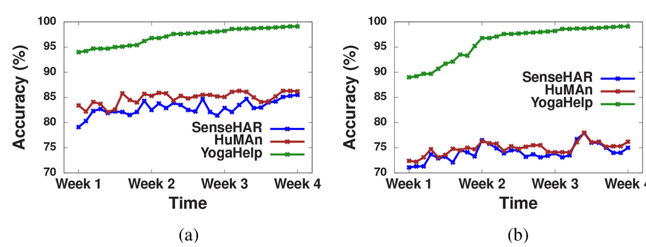


Fig. 11. Accuracy improvement for YogaHelp and existing methods over a period of four weeks. (a) For trainer. (b) For trainee.

1) Comparison on Step Recognition Accuracy: The step recognition accuracy is compared for trainer and trainee using the data collected from 5 trainers and 15 trainees over a period of four weeks. Table III illustrates the comparison results. We observed following points from the table:

a) YogaHelp outperforms the state-of-art methods with an accuracy of around 15% and 18% for trainer and trainee, respectively. It clearly indicates the effectiveness of feedback provided to the users in YogaHelp system.
b) Accuracy drop from trainer to trainee is lesser for YogaHelp system than that of existing methods. It shows the reliability of the system for the trainees.
c) Similar to YogaHelp, SenseHAR, and HuMAn both find difficulty in recognizing steps 4,5,8, and 9, and thus result in lower accuracy than that of other steps.

2) Comparison on Accuracy Improvement: We also conduct an experiment to compare the proposed system with existing methods on how accuracy improves over the period of four weeks for trainers and trainees. The obtained results are illustrated in Fig. 11. It is easy to observe in the results that YogaHelp system helps to improve the accuracy of yoga step execution from 93.7% to 98.7% for trainers and from 88.7% to 98.3% for trainees, within four weeks. On the contrary, SenseHAR and HuMAn are hardly able to manage 85% and 75% of accuracy for trainers and trainees, respectively.

IX. Conclusion

In this article, we presented a YogaHelp system for amateur practitioners to learn the correct execution of 12 linked-steps of Sun Salutation without any supervision of a trainer. The system leveraged motion sensors for identifying how correctly the steps were executed by the user. Deep learning based recognition model has improved the accuracy of the proposed system without any explicit feature extraction. We designed two most meaningful parameters, speed of step execution and angular deviation of posture, to describe the correctness level of step execution in the feedback report. The experimental results revealed the usefulness of the feedback for improving the performance within four weeks. We believe that the proposed system is reliable enough for training correct postures of Sun Salutation yoga to amateurs.

Currently, YogaHelp system exploited only the training data collected from the yoga trainers while building the recognition model. In future, we are planning to incorporate incremental learning, where the test data may be utilized to retrain the model for better generalization. Further, we believe that the use of cameras along with motion sensors can improve the performance of YogaHelp system but at the cost of complexity. Thus, maintaining a tradeoff between the improvement and the complexity of the system also opens a new research direction.

References

[1] J. V. Jeyakumar, L. Lai, N. Suda, and M. Srivastava, “SenseHAR: A robust virtual activity sensor for smartphones and wearables,” in *Proc. SenSys*, 2019, pp. 15–28.
[2] Y. Kowsar, E. Velloso, L. Kulik, and C. Leckie, “LiftSmart: A monitoring and warning wearable for weight trainers,” in *Proc. UbiComp/ISWC*, 2019, pp. 298–301.
[3] P. Bharti, D. De, S. Chellappan, and S. K. Das, “Human: Complex activity recognition with multi-modal multi-positional body sensing,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 857–870, Apr. 2019.
[4] A. Gupta, H. P. Gupta, B. Biswas, and T. Dutta, “A fault-tolerant early classification approach for human activities using multivariate time series,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1747–1760, May 2021.
[5] S. Jeon, H.-J. Yoon, Y. S. Lee, S. H. Son, and Y. Eun, “Biometric gait identification for exercise reward system using smart earring,” in *Proc. SenSys*, 2018, pp. 367–368.
[6] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, “A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1384–1393, Apr. 2019.
[7] N. Manoharan, K. Sha, and J. Lu, “Exercise evaluation using wrist sensors,” in *Proc. CHASE*, 2017, pp. 268–269.
[8] X. Guo, B. Liu, C. Shi, H. Liu, Y. Chen, and M. C. Chuah, “WiFi-enabled smart human dynamics monitoring,” in *Proc. SenSys*, 2017, pp. 1–13.
[9] F. Wang, W. Gong, and J. Liu, “On spatial diversity in WiFi-based human activity recognition: A deep learning-based approach,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2035–2047, Apr. 2019.
[10] Y. Wang and Y. Zheng, “Modeling RFID signal reflection for contact-free activity recognition,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 1–22, 2018.
[11] Z. Wu, J. Zhang, K. Chen, and C. Fu, “Yoga posture recognition and quantitative evaluation with wearable sensors based on two-stage classifier and prior Bayesian network,” *Sensors*, vol. 19, no. 23, 2019, Art. no. 5129.
[12] Y.-H. Byeon, J.-Y. Lee, D.-H. Kim, and K.-C. Kwak, “Posture recognition using ensemble deep models under various home environments,” *Appl. Sci.*, vol. 10, no. 4, 2020, Art. no. 1287.
[13] S. K. Yadav, A. Singh, A. Gupta, and J. L. Raheja, “Real-time yoga recognition using deep learning,” *Neural Comput. Appl.*, vol. 31, no. 12, pp. 9349–9361, 2019.
[14] S. Jain, A. Rustagi, S. Saurav, R. Saini, and S. Singh, “Three-dimensional CNN-inspired deep learning architecture for yoga pose recognition in the real-world environment,” *Neural Comput. Appl.*, vol. 33, no. 12, pp. 6427–6441.
[15] M. Gochoo *et al.*, “Novel IoT-based privacy-preserving yoga posture recognition system using low-resolution infrared sensors and deep learning,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7192–7200, Aug. 2019.
[16] J. Lee, H. Joo, J. Lee, and Y. Chee, “Automatic classification of squat posture using inertial sensors: Deep learning approach,” *Sensors*, vol. 20, no. 2, 2020, Art. no. 361.
[17] Y. Liu, Z. Li, Z. Liu, and K. Wu, “Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors,” in *Proc. MobiSys*, 2019, pp. 287–299.
[18] S. Shen, H. Wang, and R. Roy Choudhury, “I am a smartwatch and i can track my user’s arm,” in *Proc. MobiSys*, 2016, pp. 85–96.
[19] N. Promrit and S. Waijanya, “Model for practice badminton basic skills by using motion posture detection from video posture embedding and one-shot learning technique,” in *Proc. AICCC*, 2019, pp. 117–124.
[20] Z. Luo, B. Peng, D. Huang, A. Alahi, and L. Fei-Fei, “Unsupervised learning of long-term motion dynamics for videos,” in *Proc. CVPR*, 2017, pp. 7101–7110.
[21] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, “Sensing technology for human activity recognition: A comprehensive survey,” *IEEE Access*, vol. 8, pp. 83 791–83 820, 2020.
[22] X. Liu, S. Tamminen, T. Korhonen, J. Röning, and J. Riekki, “Prediction of sleep efficiency from big physical exercise data,” in *Proc. UbiComp/ISWC*, 2019, pp. 1186–1189.
[23] H. Ding *et al.*, “FEMO: A platform for free-weight exercise monitoring with RFIDs,” in *Proc. SenSys*, 2015, pp. 141–154.
[24] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+ D: A large scale dataset for 3D human activity analysis,” in *Proc. CVPR*, 2016, pp. 1010–1019.
[25] H. Jain and G. Harit, “A framework to assess Sun salutation videos,” in *Proc. ICVGIP*, 2016, pp. 1–8.
[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Cambridge, Massachusetts, US, 2016.

Ashish Gupta received the M.Tech. degree in computer engineering from the Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India, in 2012, and the Ph.D. degree in computer science and engineering from the Indian Institute of Technology (BHU), Varanasi, India, in 2020.

His research interests include sensor data analytics, machine learning, and IoT.

Hari Prabhat Gupta received the Ph.D. degree in computer science and engineering from the Indian Institute of Technology Guwahati (IITG), Guwahati, India, in 2014.

He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India. His research interests include IoT, Ubiquitous Computing, and WSN.