




Article

Recognition and Repetition Counting for LME exercises in Exercise-based CVD Rehabilitation: A Comparative Study using Artificial Intelligence Models

Ghanashyama Prabhu ^{*,1,2,4} , Noel E. O'Connor ^{1,2} , and Kieran Moran ^{1,3} 

¹ Insight Centre for Data Analytics, Dublin City University, Dublin, Ireland; ghanashyama.prabhu2@mail.dcu.ie (G.P.); noel.oconnor@dcu.ie (N.C.); kieran.moran@dcu.ie (K.M.)

² School of Electronic Engineering, Dublin City University, Dublin, Ireland

³ School of Health and Human Performance, Dublin City University, Dublin, Ireland

⁴ Manipal Academy of Higher Education, Manipal, India; gs.prabhu@manipal.edu

* Correspondence: ghanashyama.prabhu2@mail.dcu.ie

Abstract: Exercise-based cardiac rehabilitation requires patients to perform a set of certain prescribed exercises a specific number of times. Local muscular endurance (LME) exercises are an important part of the rehabilitation program. Automatic exercise recognition and repetition counting, from wearable sensor data is an important technology to enable patients to perform exercises independently in remote settings, e.g. their own home. In this paper we first report on a comparison of traditional approaches to exercise recognition and repetition counting, corresponding to supervised machine learning and peak detection from inertial sensing signals respectively, with more recent machine learning approaches, specifically Convolutional Neural Networks (CNNs). We investigated two different types of CNN: one using the AlexNet architecture, the other using time-series array. We found that the performance of CNN based approaches were better than the traditional approaches. For exercise recognition task, we found that the AlexNet based single CNN model outperformed other methods with an overall 97.18% F1-score measure. For exercise repetition counting, again the AlexNet architecture based single CNN model outperformed other methods by correctly counting repetitions in 90% of the performed exercise sets within an error of ± 1 . To the best of our knowledge, our approach of using a single CNN method for both recognition and repetition counting is novel. In addition to reporting our findings, we also make the dataset we created, the INSIGHT-LME dataset, publicly available to encourage further research.

Keywords: CVD rehabilitation; Local muscular endurance exercises; Exercise-based rehabilitation; Deep Learning; AlexNet; CNN, SVM, kNN, RF, MLP, PCA, multi-class classification, INSIGHT-LME dataset

1. Introduction

Cardiovascular disease (CVD) is the leading cause of premature death and disability in Europe and worldwide [1]. Exercise-based cardiac rehabilitation is a secondary prevention program which has been shown to be effective in lowering the recurrence rate of CVD and reducing the need for medicines [2,3]. Cardiac rehabilitation is organised in three phases, with the third phase targeting long-term exercise maintenance by attending community-based rehabilitation programs or through home-based exercise self-monitoring programs. However, a significant challenge is that the uptake and adherence of community-based cardiac rehabilitation is very low, whereby only 14% to 43%

of cardiac patients participate in rehabilitation programs [4,5]. Key reasons for this include lack of disease-specific rehabilitation programmes, long travel time to such programmes, scheduling issues, and low self-efficacy associated with a perception of the poor 'body image' or poor exercise technique [6]. Part of the solution is delivering an alternative, such as a technological platform that can motivate the user to engage with exercise-based cardiac rehabilitation and enable them to do so in any environment. In an ideal scenario, people would undertake a variety of exercise programs, either specifically prescribed or based on personal preference, that suits their goals and that allows them to avoid exercise associated with comorbidities, for example arthritis of the shoulder. In this scenario of "exercising anywhere", where people may be exercising in their own homes on their own, it is extremely important that they receive feedback on the exercises in order to help them track their progress and stay motivated. A technological approach may facilitate this by integrating a single wearable sensor for assessing exercise movement into an appropriate smartphone application (i.e. eHealth and eRehabilitation). However, two key challenges are presented with this approach. Firstly, it is important to be able to automatically recognize which exercises are being completed and secondly, once recognised to provide the number of repetitions as quantitative feedback on the amount of exercises performed to build the user's confidence. This would also allow people to complete elements of their own training program disbursed over the day in any environment. For example, someone could complete different exercises in home or in the workplace [7,8].

In recent years, wearable sensors have been used in the assessment of human movement in many domains including health [9], wellness [10], musculoskeletal injury and sport [11]. Research on CVD patients found that most (67% ~ 68%) patients showed interest in mobile phone-based CR [5]. Accelerometer and Gyroscopes along with magnetometer in wearables are accurate in measuring translatory and rotary movements are frequently found in wearable sensor units. Multiple wearable sensors have been deployed on the body to recognize the day-to-day activities like walking, jogging, climbing stairs, sitting, standing [12–15], exercise feedback [16] and in qualitative evaluation of human movements [17], gym activities [18], and rehabilitation [19]. However, keeping in mind the likelihood of technology adoption combined with the aesthetics of wearing multiple sensors, it is more likely that individuals would be willing to wear a single sensor with the wrist being the ideal worn location. Single wearable sensor usage studies, either phone-based or using inertial measure units, have been conducted in recent years. Applications include recognising day-to-day activities [20–25], recognition of multiple complex exercises [26] or single exercises like lunges [16], squats [27] as well as repetition counting [8,26,28]. Detailed related work is discussed in the next section 2: Related Work.

In this work, we focus on exercise recognition and repetition counting using a single wrist-worn inertial sensor for 10 local muscular endurance (LME) exercises that are specifically prescribed in exercise-based CVD rehabilitation. We present a comparative analysis between different traditional supervised machine learning algorithms and the current state of the art methods in deep learning architectures. As the novelty of this work, we claim following novel contributions. Firstly, we propose the use of a single CNN model for the repetition counting task of different exercises. Secondly, the dataset we created, the INSIGHT-LME dataset, is made publicly available to encourage further research on this topic (<https://drive.google.com/open?id=124ugOPzzoXFTmSJv9K37NC18Lz1FnFPO>).

2. Related Work

Machine learning and deep learning are artificial intelligence methods that employ statistical techniques to learn underlying hidden distributions from observed data. Technology advances in sensor manufacturing and micro-miniaturization has resulted in low-cost micro-sensor devices like wearables, that are capable of effective lossless streaming and/or storing translatory and rotary movement information for further processing. The application of machine learning methods to study data from human movements and activities with a view to detecting and understanding these activities is referred to as human activity recognition (HAR).

Within the HAR literature, many machine learning and deep learning-based models have been used to study day-to-day routine activities like walking, jogging, running, sitting, drinking, watching TV, etc., [9,13,14,29–32] and to assess sporting movements [11] and indoor based exercises for strength training [33]. Machine learning methods have been predominantly used for exercise recognition using multiple wearable sensors [17,18,34,35], and specifically in the areas of free weight exercise monitoring [36], the performance of lunge evaluation [16], limb movement rehabilitation [19], and intensity recognition in strength training [37]. However, the use of multiple sensors is non-ideal in practice because of cost, negative aesthetics and potential user uptake [31]. Therefore, in our research, we would like to use only a single wrist-worn inertial sensor for exercise recognition and repetition counting.

Several studies [13,14,20,21,28,31] use a single wrist-worn wearable or phone-based sensor for recognition of various day-to-day activities with machine learning models. However, very few studies have been completed in recognizing exercises, and even then only with a limited number of exercises: lunges evaluation [16], correct squat measurement [8,16,38]; single best axis for exercise repetition counting [28]. A very recent article on recognition and repetition counting for complex exercises uses deep learning [26] which recognizes a set of CrossFit exercises using a single CNN for exercise recognition, but multiple CNN models for individual the repetition counting. However, no other studies appear to have studied a wide number of exercises, and none specifically for CVD rehabilitation through LME exercises.

To date, the vast majority of HAR studies detailed above have used traditional machine learning approaches such as decision trees, Naive Bayes, random forest, perceptron neural networks, k-nearest neighborhood, and support vector machines. There is, however, a growing interest in the potential use of deep learning methods in the field of activity recognition mainly using CNN [26,39–42], and recurrent models [40,43]. A small number of studies [39,40,42,44] have shown the significant advantage of using deep learning models in the general area of human activity recognition. However, very few studies appear to have used deep learning models in exercise recognition and repetition counting. In addition, to date no studies have done a comparative study of using traditional machine learning methods and state of the art CNN methods to identify the best possible method for exercise recognition and repetition counting. Furthermore, to the best of our knowledge, there are no works reported using single deep a CNN model for exercise recognition and repetition counting following recognition. The use of a single model for repetition counting is attractive as it eliminates the need for an exercise specific repetition counter and reduces the dependency on the total number of resources required in repetition computation.

3. Data Set

Currently, there exists no publicly available data sets with a single wrist-worn sensor for endurance-based exercises that are commonly prescribed in cardiovascular disease rehabilitation (CVD) programs. Therefore, we collected a new data set of local muscular endurance exercises for this purposes Table 1. In the data collection process, consenting participants performed the ten LME exercises in two sets (constrained set and unconstrained set) and some common movements which were observed by any exerciser in between two exercises. The constrained set of exercises involves participants performing the exercises while observing demonstrative videos and following the limb movement actions relatively synchronous with the demonstrator in the video. The unconstrained set of exercises involved participants performing the set of LME exercises without the assistance of demonstrative videos. Inclusion of the non-exercise movements were essential that the built models can distinguish the actions corresponding to the exercises movements from that of non-exercise movements. The data set was then used for training, validating and testing different machine learning and deep neural network models.

Table 1. List of LME exercises with the acronym

Exercise Type	Exercise	Acronym	Dumbbell Use
Upper-Body LME exercises	Bicep Curls	BC	A pair
	Frontal Raise (arms up)	FR	A pair
	Lateral Raise (arms up)	LR	A pair
	Triceps Extension (right arm)	TER	Single
	Pec Dec	PD	Single
	Trunk Twist	TT	Single
Lower-Body LME exercises	Standing Bicycle Crunches	SBC	— No —
	Squats	SQ	Single
	Leg Lateral Raise (right leg)	LLR	— No —
	Lunges (alternate sides)	L	A pair

3.1. Sensor Calibration

Sensor calibration is a method of improving the sensor unit's performance to get a very precise and accurate measurement. The Shimmer3 (**Figure 1(a)**) inertial measurement unit (IMU) is a light-weight wearable sensor unit from Shimmer-Sensing¹. Each IMU comprises of a 3 MHz MSP430 CPU, two 3D accelerometers, a 3D magnetometer and a 3D gyroscope. A calibrated Shimmer3 IMU, when firmly attached on the limb, is capable of collecting precise and accurate data. Each Shimmer3 has a microSD to store the data locally or can stream the data over Bluetooth.

Shimmer3 inertial measurement units were used in the exercise data collection process and they were calibrated using Shimmer's 9DoF calibration application². The IMUs were used with a sampling frequency of 512 Hz along with a calibration range of $\pm 16g$ for the 3D low noise accelerometer and a $\pm 2000dps$ for the 3D gyroscope. An elastic wristband was used to firmly place the Shimmer3 IMU on the right-wrist of each participant during the data collection process. The sensor orientation and pictorial representation of the unit attachment on the right-wrist are shown in **Figure 1(a)** and **Figure 1(b)** respectively.

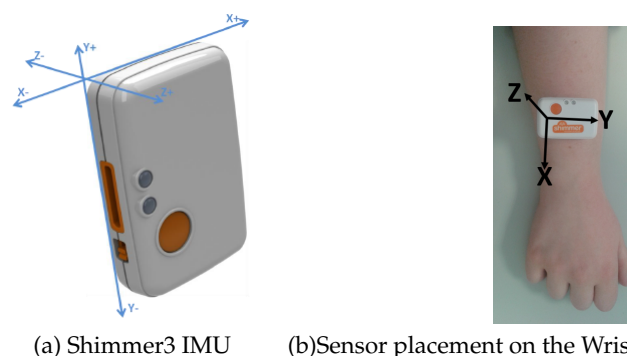


Figure 1. Illustration of (a) Shimmer3 IMU, (b) Sensor placement and sensor orientation on the right-wrist.

3.2. LME Exercise set and Experimental Protocol

Ten local muscular endurance (LME) exercises, as listed in Table 1, prescribed in cardiovascular disease rehabilitation were selected for data collection. The ten exercises comprise of six upper-body exercises: bicep curls (BC), frontal raise (FR), lateral raise (LR), triceps extension-right arm (TER), pec dec (PD), and trunk twist (TT); along with four lower-body exercises: squats (SQ), lunges - alternating sides (L), leg lateral raise (LLR), and standing bicycle crunches (SBC). The representative postures for

¹ <http://www.shimmersensing.com/products/shimmer3>

² <https://www.shimmersensing.com/products/shimmer-9dof-calibration>

the execution of six upper-body LME exercises are shown in Figure 2 and that of four lower-body LME exercises are shown in Figure 3.

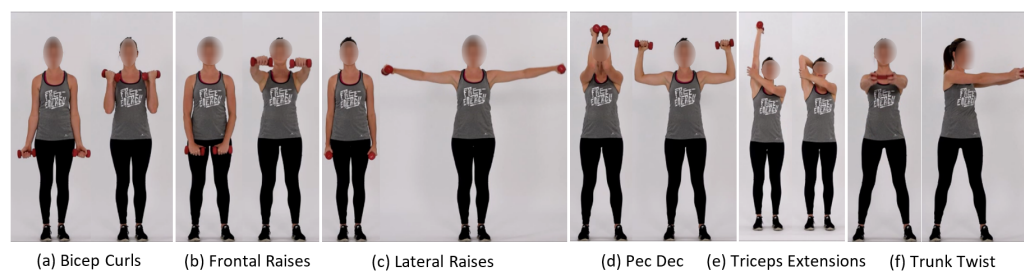


Figure 2. Upper-body LME exercises

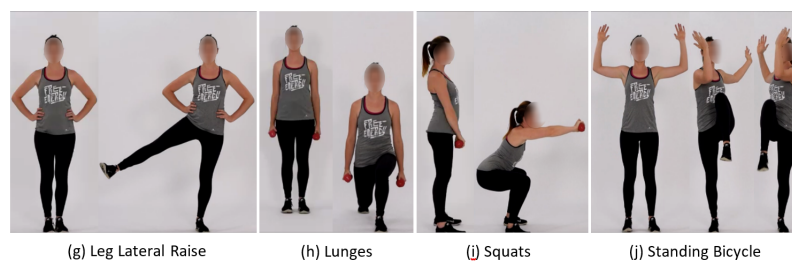


Figure 3. Lower-body LME exercises

A pair of 1 kg dumbbells were used by each participant while performing BC, FR, LR, and PD exercises. A single dumbbell of 1kg were used during TER, TT, L, and SQ. Exercises LLR and SBC were performed without dumbbells. The data from these exercises correspond to ten different classes of exercise. The ten exercises that were used in CVD rehabilitation were either employed a single joint movement effect (BC, FR, LR, PD, TER, and LLR) or employed multiple joint movements (TT, L, SQ, and SBC). Some of these exercises have significantly similar arm movements and hence it was considered of interest to investigate how the models were able to distinguish between these exercises. It was also of interest to see how robust the models were in terms of their capacity to distinguish between the exercise actions in comparison to limb movements that were commonly observed between the exercises. The common limb movements selected for inclusion were side bending, sit-to-stand and stand-to-sit, lean down to lift water bottle or dumbbell kept on the floor, arm-stretching front-straight, lifting folded arm up-ward, and body stretching up-ward with calf raising for relaxation. These observed common actions have significant similarity in terms of limb movement with that of the exercises. The data corresponding to these common actions together describe an eleventh class of movement.

3.3. Participants

A total of seventy-six volunteers (47 males, 29 females, age group range: 20 - 54 yrs, median age: 27 years) participated in the data collection process. No participants had any musculoskeletal injury in the recent past which would affect the exercise performance and all were healthy. Having prior knowledge of exercise was not a criterion in volunteer recruitment. The study protocols used in data collection were approved by the university research ethics committee [REC Reference: DCUREC/2018/101].

3.4. Data Capture Process

The exercise protocol was explained to the participants on their arrival to the laboratory. Each participant underwent a few minutes of warm-up with arm-stretching, leg-stretching and basic body-bending exercises. We developed an exclusive MATLAB-GUI module [Figure 4a] to collect the data from the participants wearing IMUs via Bluetooth streaming. The “Exercise Data Capture Assist Module” was designed to select a particular exercise, to play demo videos to illustrate how to perform

the exercise, to initialize and disconnect Shimmer IMUs remotely, to start recording exercise data, to stop recording exercise data and a storage path location to store the streamed data. The streamed data were stored automatically with participant_ID and the exercise type in the filename, completely anonymizing the details of the participants. The Shimmer-MATLAB Instrument driver interface was used to connect and collect data from multiple shimmer units, therefore the designed module was capable of recording from multiple participants at any given time.

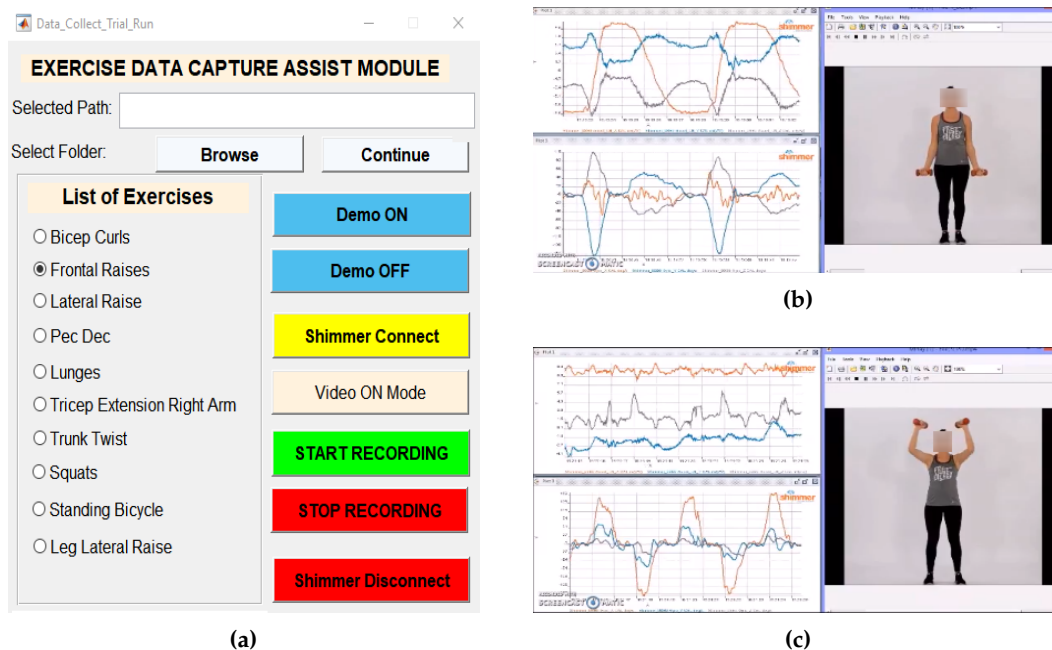


Figure 4. MATLAB GUI for exercise data capture process. (a) Interactive MATH Lab GUI (b) Bicep Curls Data Streaming (c) Pec-Dec Data Streaming

All consenting participants performed the ten exercises in two sets and the common movements as described in section 3.2. During the constrained set of exercising, the participants performed the LME exercises while observing demonstrative videos on the screen and following the limb movement actions relatively synchronous with the demonstrator in the video. Participants were told to pay particular attention to the following: the initial limb resting position, how to grip the dumbbells (in case the exercise requires the use of dumbbells), the limb movement plane and the speed of limb movement during demo video. The constrained setup facilitated minimal variations in the collected data in terms of planar variations and speed and thus ensuring participants perform exercises at a similar tempo of movement. The participants were asked to perform each exercise for 30 seconds which resulted in approximately 7 to 8 repetitions. After each exercise, participants were given sufficient time to rest before moving on to the next exercise.

During the unconstrained set of exercising, a timer was used and displayed on the screen. Participants performed the exercises by recalling what they had learned during the constrained performance and were free to execute them for 30 seconds. The data collected during the unconstrained set corresponds to a variable range of variations from that of exercise data collected from the constrained set of execution. The variations observed were in terms of the plane of limb movement, speed, and the rest position of the limb; these variations were used to mimic macro variations that would typically during home-based exercising.

In addition to the constrained set and the unconstrained set of data collection, participants were instructed to perform the common movements as stated in section 3.2. Inclusion of these non-exercise movements were essential that the built models can distinguish the actions corresponding to the exercises movements from that of non-exercise movements. Participants were asked to perform each

of these actions repeatedly for about 30 seconds. The 5 second instances from each of these actions represents almost one full action and collectively constitutes the eleventh class.

All IMUs used in the process of data capture were calibrated as stated in section 3.1. The IMU was securely placed on the right-wrist, as shown in Figure 1(b), with the help of an elastic band and demo videos of all the exercises were shown to each participant. Data collected from both the constrained set and the unconstrained set were class labelled and stored in ten different exercise folders. An eleventh class labelled as “others” was created to store the data from all of the common movements.

Among 76 participants, 75 people participated both in the constrained set and unconstrained set of data collection. However, one participant performed only the constrained set. Only a few participants had not performed all the exercises. However, overall the collected data set was well balanced and Table 2 indicates the participation summary for each exercise under the constrained set and the unconstrained set of data capture. The data set was then segregated and stored into three different sets: the training set, the validation set and the test set, and were used in all model building. The data from 46 participants were used in the training set and the data from 15 participants were used in both the validation set and the test set. The entire data set is termed the INSIGHT-LME dataset.

Table 2. Data capture participation summary, exercise-wise

Exercise Type	Exercise Acronym	Number of Participants	
		Constrained Set	Unconstrained Set
Upper-Body LME exercises	BC	76	75
	FR	76	75
	LR	76	74
	TER	76	75
	PD	75	74
	TT	76	75
Lower-Body LME exercises	SBC	75	74
	SQ	73	73
	LLR	75	74
	L	73	75
Others	OTH	76	75

4. Methodology: Overall Framework

Figure 5 represents an overall framework with three major processing blocks. The comparative study aims to find the best possible method from the different AI models for each task in automatic exercise recognition and repetition counting. The first block represents the INSIGHT-LME data set processing and data preparation in terms of filtering, segmentation, 6D vector generations and/or 2D image creation. Data preparation requirements were different for each specific method used in both comparative studies and hence data processing specifics pertaining to individual method are discussed along with each model below.

The second block represents the comparative study for the exercise recognition task. The exercise recognition task was treated as a multi-class classification task. We compared traditional approaches in the exercise recognition, which were supervised machine learning approaches, with two recent deep CNN based approaches. The first deep CNN model used the AlexNet architecture and the second deep CNN model was developed from scratch to use the time-series array information. In supervised machine learning (ML) models, different models were constructed using the four supervised algorithms such as SVM, RF, kNN and MLP. The eight models from these four ML algorithms were studied with and without the dimensionality reduction measures using principal component analysis (PCA). The best model from the supervised ML set was then compared with two deep CNN models to find the best possible method for the exercise recognition task.

The third block represents the comparative study for the repetition counting task. The repetition counting task was treated as a binary classification task followed by a counter to count the repetitions.

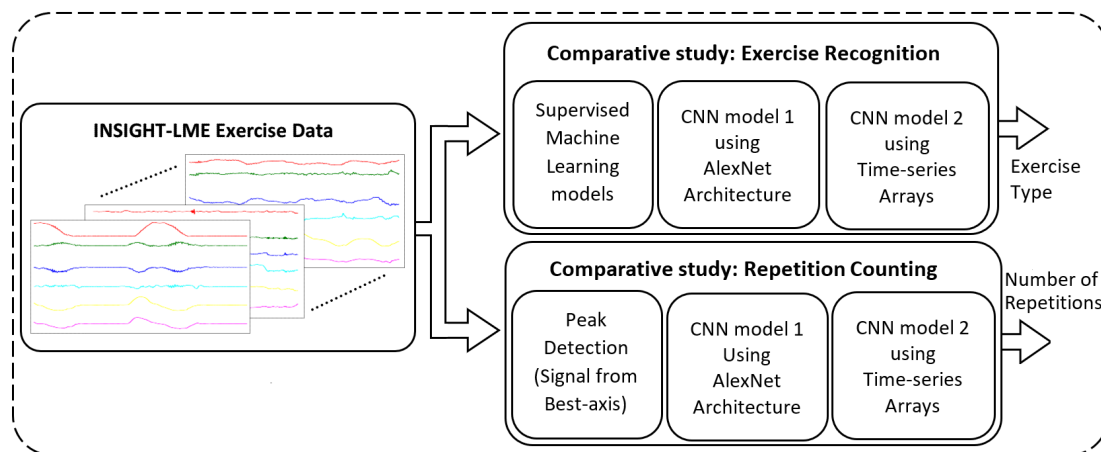


Figure 5. Framework for the comparative study of models. Task 1: Exercise recognition, Task 2: Repetition counting

Again three different methods were used in repetition counting and the performances were compared to find the single best method for repetition counting. We compared traditional signal processing models based on peak detection with two deep CNN approaches. Deep CNN models were similar to the exercise recognition task, the first deep CNN model used the same AlexNet architecture as previously, whilst the second deep CNN model was developed from scratch to work with the time-series array information.

4.1. Exercise Recognition with Supervised Machine Learning Models

Figure 6 illustrates the end-to-end pipeline framework adopted for supervised machine learning model-based (ML) exercise recognition. As discussed in section 4, a total of eight supervised machine learning models were studied using this framework to classify the eleven activity classes, in which, ten classes were corresponding to the ten LME exercises and the eleventh class “others” for the common movements observed during exercising. The eight supervised machine learning models were constructed using four algorithms, support vector machine (SVM), random forest (RF), k-nearest neighbors (kNN), and multi-layer perceptron (MLP), either with or without dimensionality reduction using principal component analysis (PCA).

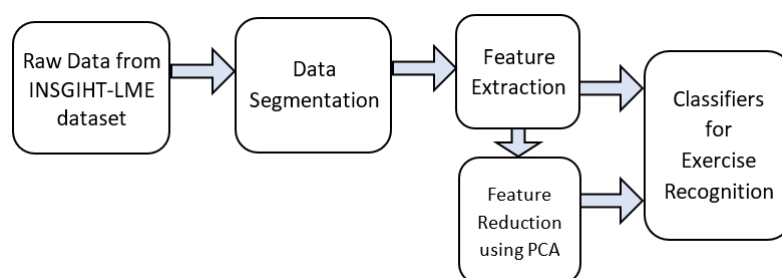


Figure 6. End-to-end pipeline framework for the machine learning models

4.1.1. Data Segmentation

Raw data from the INSIGHT-LME dataset (section 3.4), corresponding to ten LME exercises prescribed for CVD rehabilitation and the common movements, were subjected to data segmentation. 25 seconds of 3D accelerometer and 3D gyroscope data of each exercise, excluding an initial few seconds of recording, were segmented from each participant retaining class-label information. The process of segmentation was carried out on all the three sets: training set, validation set and test set from the

INSIGHT-LME dataset. The segmented 3D accelerometer and 3D gyroscope signal corresponding to the bicep curl exercise are shown in Figure 7. 3D sensor plots for all ten LME exercises are given in Appendix.A. The 25seconds of 6D segmented data consists of approximately five or six repetitions of an exercise with each repetition duration lasting approximately 4 seconds. The segmented data with retained class-label information was used in feature extraction in the next stage.

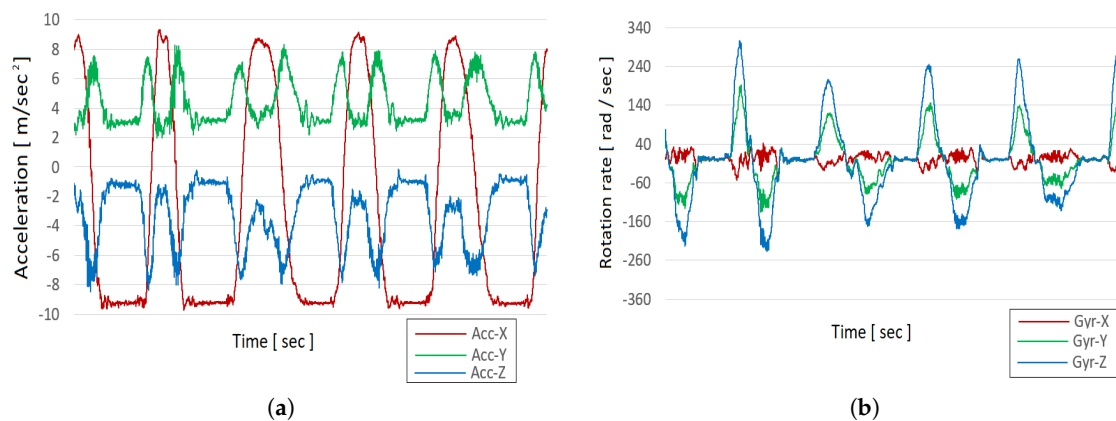


Figure 7. 25 seconds segmented plots of Bicep Curls exercise (a) 3D Accelerometer signals, (b) 3D Gyroscope signals

4.1.2. Feature Extraction

Time and frequency features were extracted from the 6D segmented data using an overlapping sliding window method. Three sliding window-lengths of 1 second, 2 seconds, and 4 seconds were used along with an overlap of 50% in all cases to find an optimum window length selection in the classifier design. The maximum window-length selection was restricted to 4 seconds because the length of one complete repetition of an exercise was approx. 4 seconds.

From each sliding window, statistical time-domain features, such as, mean, standard deviation, root mean square value, minimum and maximum values were computed from each axis of 3D accelerometer and 3D gyroscope data. Auto-correlation values were computed from the pairs of XY, XZ and YZ axes of both accelerometer and gyroscope sensor data. Frequency-domain features, such as, Fourier coefficients and energy were computed from each axis of the 3D accelerometer and 3D gyroscope data.

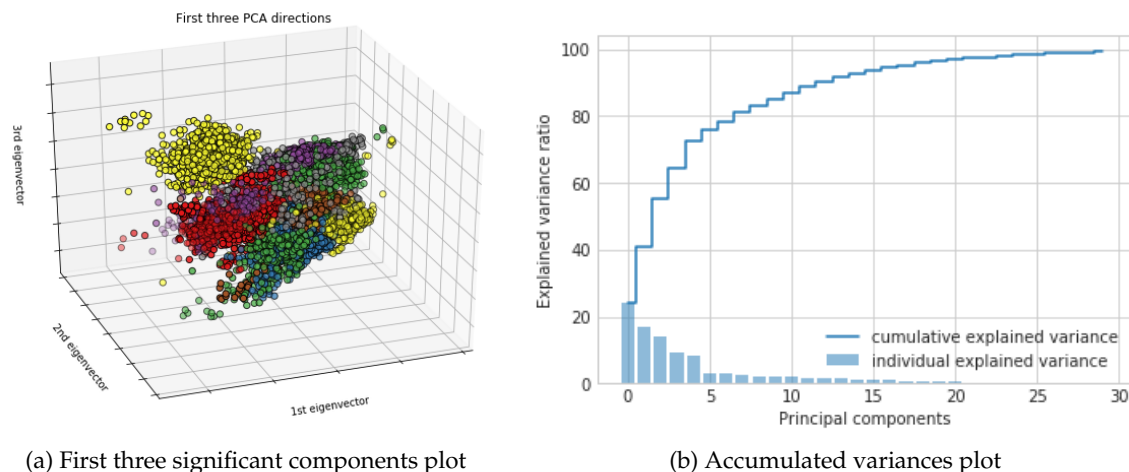
A vector of 48 features, 24 time-frequency features were thus computed for each sliding window and repeated for every slide. Class label information was retained for the feature vectors corresponding to each exercise class and the “others” class. A combined feature set, referred to as “training feature set”, was formed by combining feature vectors from all the exercise classes and the “others” class from the training set. The training feature set is computed for each sliding windows of the 1 sec, 2 sec, and 4 sec window-length on the training set of the INSIGHT-LME data set. Similarly, the “validation feature set” and the “test feature set”, are computed on each of the sliding windows of 1 sec, 2 sec, and 4 sec input data from the validation set and the test set of the INSIGHT-LME data set, respectively.

Feature sets computed over each sliding window length were then used for training, validation and testing of the four supervised machine learning models using ML algorithms (SVM, RF, kNN, and MLP) forming a total of 12 classifiers.

4.1.3. Feature Reduction using PCA

To study the effect of dimensionality reduction, principal component analysis (PCA) was used on the feature sets computed from section 4.1.2 to reduce the overall feature dimensionality of the input vectors to the ML models. Significant principal components, which were having an accumulated variance greater than 99%, were retained. New feature sets corresponding to the training feature set,

validation feature set and test feature set were computed using PCA for each of the 1sec, 2sec, and 4sec window-length cases. New feature sets with dimensionality reduction using PCA were then used in the training, validation and testing of additional machine learning models using ML algorithms (SVM, RF, kNN, and MLP) for each window-length case, forming an additional 12 classifiers.



(a) First three significant components plot

(b) Accumulated variances plot

Figure 8. PCA plots for the training feature set for a 4second sliding window (a) 3D plot of the first three significant components, (b) Accumulated variances plot

4.1.4. Classifiers for Exercise Recognition

Exercise recognition from the single wrist-worn inertial sensor data for a set of exercises prescribed for cardio-vascular disease rehabilitation is a classic classification task using machine learning or deep learning methods. A total of twenty-four classifiers were constructed from the feature vectors as explained in section 4.1.2 and 4.1.3 and were analysed for exercise recognition. Each classifier model was constructed using the training set feature vectors, with 10 fold cross-validation using the grid-search method to ensure the models to have optimum hyper-parameters (for SVM models, kernel options between rbf and linear, and model parameters C and gamma values; for kNN models to find the best k-value or number of nearest neighbors; for RF models the number $n_{\text{estimator}}$ or the number of trees to be used in the forest; for MLP the step value α).

All models were first evaluated using the validation set feature vectors to evaluate the following: firstly, the optimum sliding window-length among all possible selected windowing methods was determined based on the validation accuracy measure. Secondly, to see the effect of dimensionality reduction on ML model performance. Finally, to select the single best-supervised ML model to recognize the exercises.

4.1.5. Experimental Results of Exercise Recognition with Supervised Machine Learning Models

A total of twenty-four classifiers were constructed using three sliding windowing methods with four supervised machine learning algorithms with and without dimensionality reduction using PCA. Among these models, SVM models were constructed using One-Vs-Rest multi-class classifier, and were designed have optimum hyper-parameters using a grid search method with 10-fold cross-validation. The values, $C = 100$, $\gamma = 0.01$ and rbf kernel were found to be the optimum hyper-parameters for all the 6-SVM classifiers. For all the 6 kNN models, $k = 1$ found to be the optimum value and for all the 6 RF models, $n_{\text{estimator}} = 10$ found to be the optimum value. Similarly, for all the 6 MLP classifiers the step value, $\alpha = 1$, was optimum over a range of $1.0E-5$ to $1E+3$ on a logarithmic scale.

Selection of suitable sliding window-length was done based on the validation results using the validation feature set. While the training score indicates the self classifying ability of the model, the validation score helps in accessing the suitability of any model deployment on the unseen data. The

training and validation scores for all the twenty-four classifiers segregated with the corresponding window-length are shown in Table 3 and a pictorial comparison of the validation score measure is shown in Figure 9. The training and validation score values from Table 3 indicate that the models with the window-length of 2sec or 4sec were better self-trained and has better validation measures in comparison to the models that were built using 1sec window-length.

Table 3. Classifier performance comparison over varied window-lengths

Window length	Classifiers	without PCA		with PCA	
		Training Score	Validation Score	Training Score	Validation Score
1 Sec	SVM	0.9735	0.8559	0.9674	0.8525
	MLP	0.9232	0.8190	0.9041	0.8041
	kNN	0.9390	0.8248	0.9307	0.8227
	RF	0.9925	0.8165	0.9898	0.8179
2 Sec	SVM	0.9907	0.8906	0.9875	0.8816
	MLP	0.9690	0.8615	0.9568	0.8475
	kNN	0.9715	0.8571	0.9613	0.8520
	RF	0.9956	0.8607	0.9850	0.8439
4 Sec	SVM	0.9974	0.9171	0.9965	0.9089
	MLP	0.9961	0.8709	0.9939	0.8709
	kNN	0.9944	0.8848	0.9845	0.8828
	RF	0.9995	0.8905	0.9994	0.8670

A few more observations can be drawn from the validation score results from Table 3 and Figure 9. Dimensionality reduction from 48-D time-frequency vector to 30-D transformed vector by having an accumulated variance greater than 99% using PCA had no significant effect on the performance of the models. Performance in terms of validation score for ML models with PCA were almost comparable but not improved in comparison to ML models without PCA. Also, the performance of the supervised ML models built using 4sec window-length were better when compared with the models built with a window-length of 2sec in terms of validation score. Therefore, the eight supervised ML models constructed using 4sec sliding window-length were retained for further comparison using the test set feature data to find the possibility of having a single best classifier for exercise recognition.

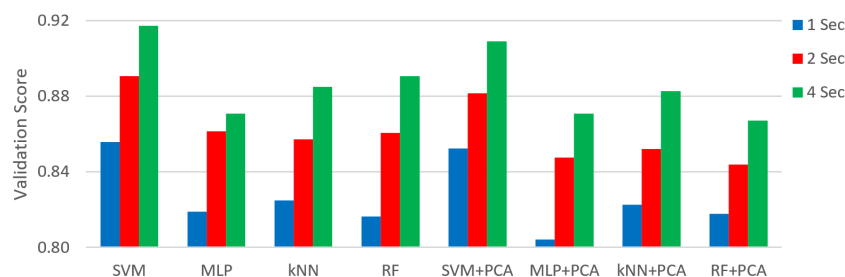


Figure 9. Comparison of validation score

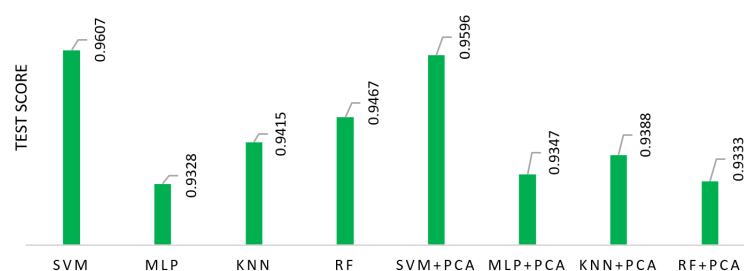


Figure 10. Test score comparison of all the eight 4 second window-length based classifier models

Figure 10 shows the test score comparison among all eight supervised ML classifiers with 4sec window-length. All the models were tested with the same test set data. The SVM model without PCA

was found to be the single best performing model with a test score of 96.07%. The SVM model with PCA was found to be the second-best model with a test score of 95.96%.

The best ML model, SVM without PCA, was further evaluated to find the performance of individual exercises and the statistical parameters such as precision, recall and F1-score measures for each exercise were tabulated in Table 4. The precision with which each exercise recognized varied from a low of 91.42% (FR) to as high as 100% (BC and TER). Recall rate was found to be 93.33% (LR and SBC) to 100% (TER) and measured F1-score measures was from 92.52% (FR) to 100% (TER).

Table 4. Performance evaluation measures of SVM Classifier on individual exercises

	Exercise Type	Acronym	Precision	Recall	F1-score
Upper-Body LME exercises	Bicep Curls	BC	1.0000	0.9970	0.9985
	Frontal Raise	FR	0.9142	0.9364	0.9252
	Lateral Raise	LR	0.9194	0.9333	0.9263
	Triceps Extension	TER	1.0000	1.0000	1.0000
	Pec Dec	PD	0.9599	0.9424	0.9511
	Trunk Twist	TT	0.9910	0.9970	0.9940
Lower-Body LME Exercises	Standing Bicycle Crunches	SBC	0.9419	0.9333	0.9376
	Squats	SQ	0.9907	0.9727	0.9817
	Leg Lateral Raise	LLR	0.9760	0.9849	0.9804
	Lunges	L	0.9296	0.9606	0.9449
Common Movements	Others	OTH	0.9481	0.9139	0.9307

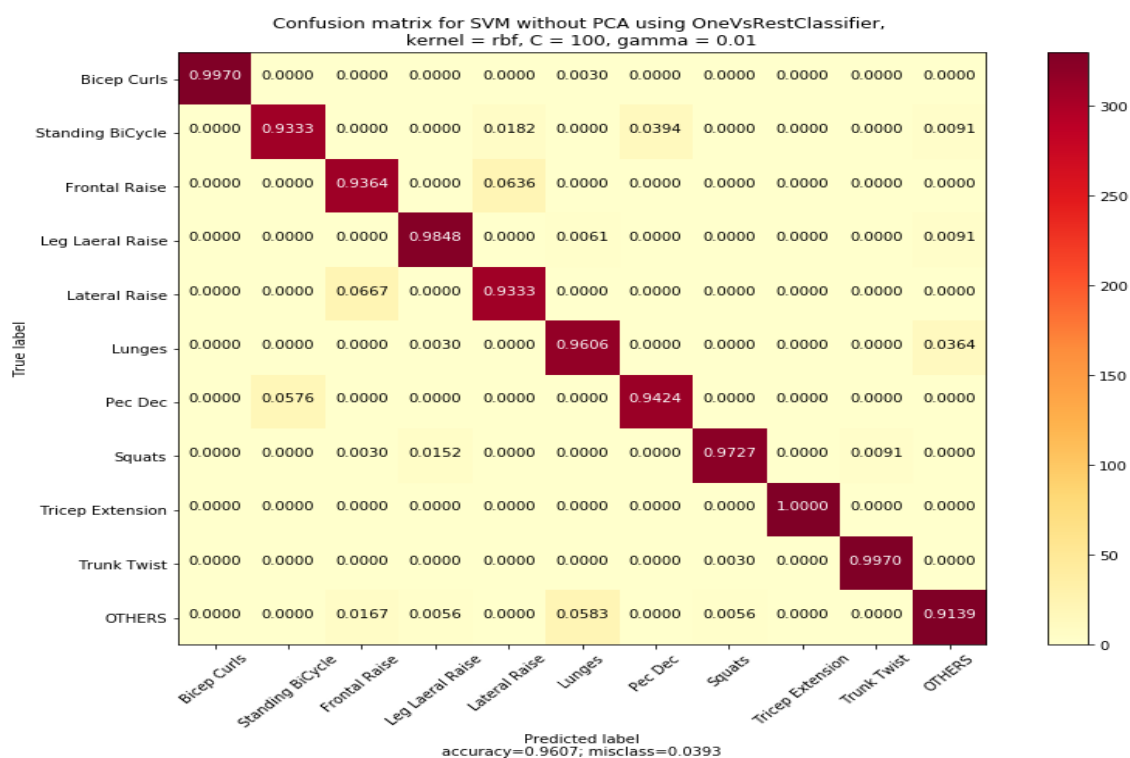


Figure 11. Normalized confusion matrix for the SVM model

These results are significant in that they indicate that with a single wrist-worn inertial sensor it is possible to recognize the LME exercises used in CVD rehabilitation process with a very good overall score of 96.07%. The significance of this finding is very important as the set of LME exercises used in this study are not only single joint upper-body exercises but also has exercises with multi-joint lower-body exercises. Lower-body exercises like standing bicycle crunches (SBC), squats (SQ), leg lateral raise (LLR), and lunges (L) were also accurately detected with a single wrist-worn device with

very high F1-scores: 93.76%, 98.17%, 98.04%, and 94.49% respectively. Additional studies into exercise recognition using CNN based deep learning models studied are discussed in the subsequent sections.

The model's normalized confusion matrix plot representing the confusions among the exercises are plotted and shown in the Figure 11. Confusions among the exercises having similar wrist-movement actions were evident from the confusion matrix plot and are discussed here. The first observed confusion was between two upper-body LMEs, the Frontal raises (FR) and the Lateral raises (LR), and a 6.36% of the FR exercises were confused with that of the LR while a 6.67% of the LR exercises were confused with that of FR. In both FR and LR exercises, raising the hands straight was commonly observed with significant movements on the plane of the accelerometer x-axis direction. However, the wrist-movement actions were different for FR from that of LR only during the movement from the initial resting position. The second observed confusion was between the exercises Pec Dec (PD) and the Standing bicycle crunches (SBC). A 3.94% confusion were observed in SBC from PD, whereas a 5.76% of PD were getting confused with SBC. The wrist rotary movements in the plane of the gyroscope y-axis direction were similar for these SBC and PD exercises. The third observation was for the lower-body LME exercise Lunges were getting confused with the common movements (others) and a 3.64% confusion was observed. However, the common movements (others) were confused with Lunges with a 5.8% confusion.

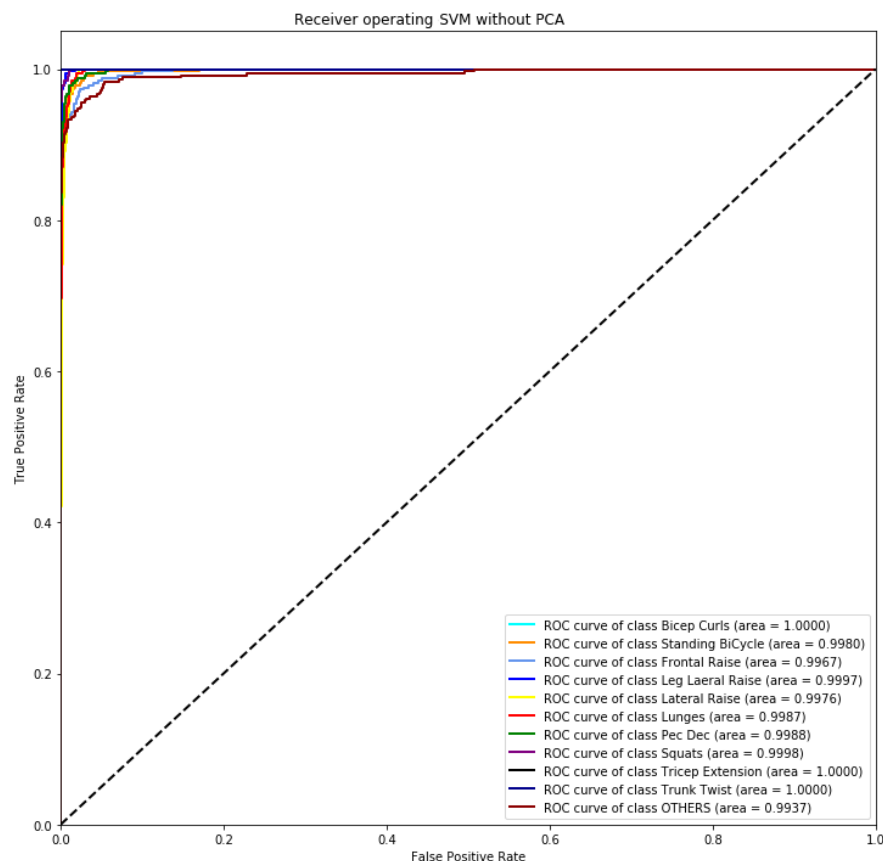


Figure 12. Receiver operating characteristic of the SVM model

Performance measurement or the capability of the classifier models were represented using the area under the curve plot or also known as the receiver operating characteristic (AUC-ROC) curve plots. The AUC-ROC curves are plotted with the true positive rate (TPR) on the y-axis against the false positive rate (FPR) the x-axis. Figure 12 represents the AUC-ROC plot for the SVM classifier without PCA and a minimum AUC value of 99.67% for FR to a maximum of 100% for BC, TT and TER.

4.2. Exercise Recognition with CNN_Model1 using AlexNet Architecture

The second method used in the comparative study of the exercise recognition task (Figure 5) was a deep convolutional neural network (CNN) model using the AlexNet architecture (Figure 13) [45]. The AlexNet model is an eight layer model with five convolutional layers, three fully-connected maximum pooling layers and a rectified linear unit (ReLU) as activation function in each layer. Batch normalization is used before passing the ReLU output to the next layer. A 0.4 dropout is applied in the fully connected layers to prevent overfitting of the data. This eight layered architecture generates a trainable feature map which is capable of classifying a maximum of 1000 different classes.

Figure 14 represents an end-to-end pipeline structure of an optimized CNN model, using the AlexNet architecture, for the LME exercise recognition task. The LME exercise recognition task was an 11-class classification task and hence we used a final output layer, a fully connected dense layer, with a softmax activation function for the classification of 11 classes. An optimum CNN model with best learning rate, optimizer function and loss function was trained using the training data and was further validated using the validation data before testing the model with the test data from the INSIGHT-LME dataset (Section 3.4).

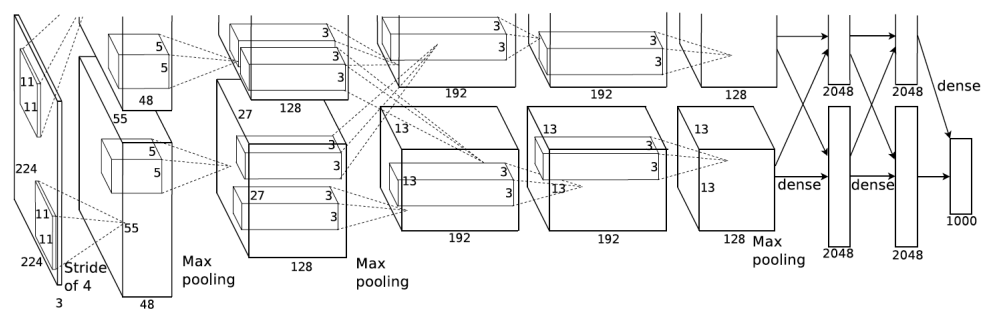


Figure 13. Illustration of AlexNet architecture. Image credits to Krizhevsky et al., the original authors of the AlexNet paper.

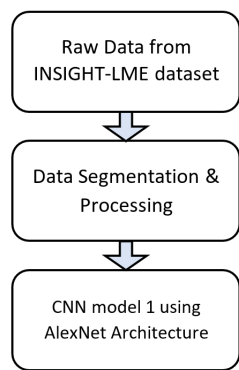


Figure 14. End-to-end pipeline for exercise recognition with AlexNet

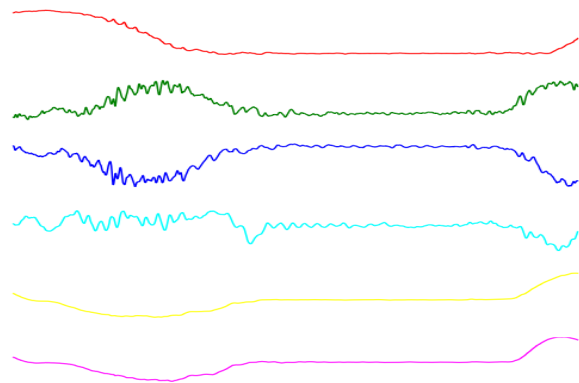


Figure 15. A sample 2D (576 x 576) image of accelerometer and gyroscope signal plot for Bicep Curls

4.2.1. Data Segmentation and Processing

The CNN model with the AlexNet architecture (CNN_Model1) requires an input data in the form of 2D images of size 227x227. Data segmentation and processing methods were used to convert the 6D time-series data from the input INSIGHT-LME dataset to 2D images. To compare the results of CNN_Model1 with the ML models discussed in Section 4.1.5, a 4-second windowing method with an overlap of 1-second was used to segment the 6D (3D accelerometer and 3D gyroscope) time-series data

and an image of size 576 x 576 with plots of all 6 axes were plotted. A 4-second, 6D time-series data segment of Bicep Curls was 2D plotted into a 576 x 576 image and is shown in Figure 15. Top three plots: red, green and blue, represents the plots corresponding to the 3D accelerometer x-axis, y-axis and z-axis; and the bottom three plots: cyan, yellow and magenta, corresponding to the plots of 3D gyroscope x-axis, y-axis and Z-axis.

An image dataset was generated, by data segmentation and processing method, from the entire time-series raw data of the INSIGHT-LME dataset using the 4-second windowing method with a 1-second overlap. The image dataset comprises of 11-classes of image data, among which, ten classes were from the ten LME exercises and the eleventh class from the common movements observed during the exercises. The training set was formed with a total of 43306 images from eleven class of data from 46 participants. Similarly, the validation set was formed with 13827 images from 15 participants and the test set was formed with 14340 images from 15 participants. Downsampling of images to 227 x 227 image was further achieved by data augmentation method in the input layer during the model implementation.

4.2.2. CNN_Model1 for the Exercise Recognition Task

An optimum model, termed CNN_Model1 here, was developed using python sequential modelling along with the Keras API, a high-end API for TensorFlow. The model constructed here was an optimum model with the best possible optimizer function, good learning rate to achieve better accuracy and with a very good loss function. The model was constructed with the choice of optimizer function among stochastic gradient descent (SGD), Adam, and RMSprop and the model was trained with varied learning rates ranging from 1e-03 to 1e-6 values. Also, the model was trained with loss functions such as categorical cross-entropy (CCE) and Kullback-Leibler divergence (KLD). The best model parameters were selected with an iterative evaluation using a varied number of epochs.

Data augmentations, like resizing of input dataset images and shuffling of input images were achieved using `flow_from_directory` method in `ImageDataGenerator` class from Keras image processing. Since the input images correspond to time-series data, augmentation operations such as shearing, flipping, and rotation tasks were not performed. CNN models were constructed using the training image dataset and validated using the validation image dataset while monitoring the validation loss. A model with a minimum validation loss was saved for each combination of network parameters. The model parameters such as training accuracy, validation accuracy, training loss and validation loss against the number of iterations were obtained and were plotted. A best model having the highest validation accuracy was selected and tested with the test image dataset and the resulting evaluation parameters such as test accuracy and loss measures were recorded. The best model, CNN_Model1, was then compared with the best model selected from supervised machine learning model. A complete list of the architecture parameters can be found in Table A1 in Appendix A.

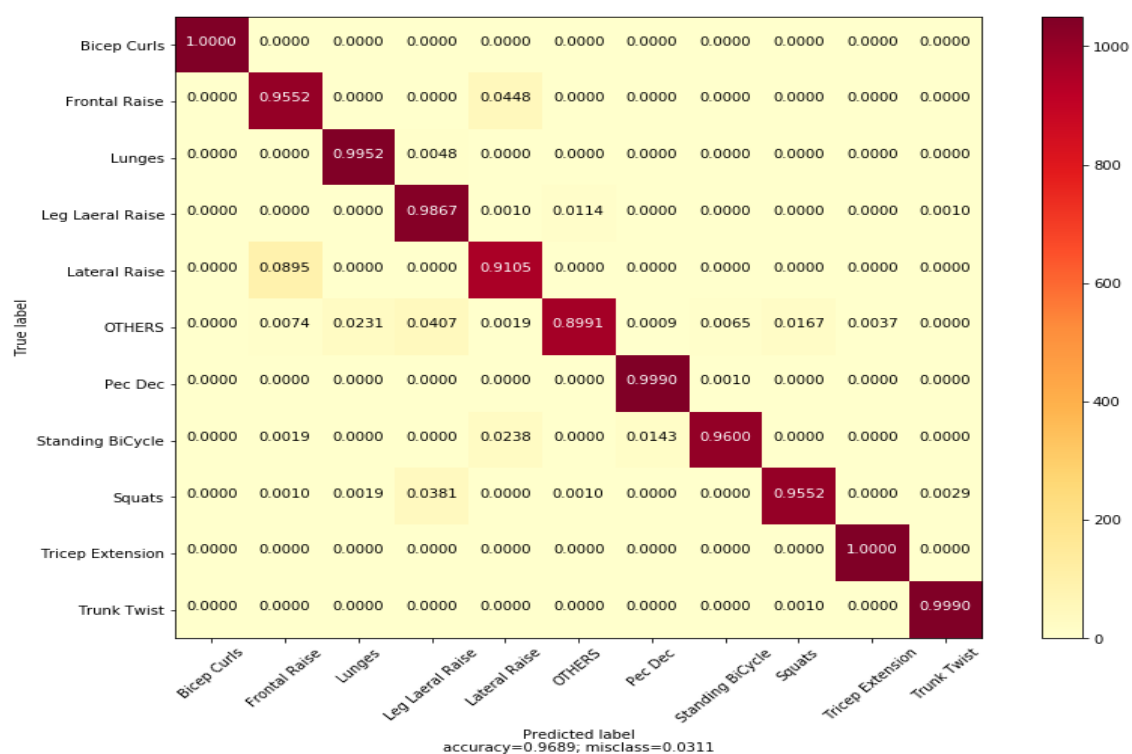
4.2.3. Experimental Results of CNN_Model1

The CNN_Model1 having Adam optimizer, a learning rate 1e-4 with KLD loss function was the best model with a training score of 99.96% and a validation score of 94.01%. The model was further evaluated using the test set image dataset and an overall F1-score of 96.895% was recorded and was almost 1% better in comparison with the SVM model, the best performing supervised ML model (Section 4.1.5). The performance of CNN_Model1 for the individual exercises were evaluated and the statistical parameters measures like precision, recall and F1-score for each exercise were tabulated in Table 5. These test score measures of the individual exercise recognition of the CNN model with AlexNet architecture were comparably better with that of the SVM model (Table 4).

Figure 16 represents the normalized confusion matrix for the CNN_Model1. The values on the main diagonal representing recall or sensitivity of the model to the individual exercises were showing significant improvement for all the exercises, in comparison with the SVM model, except for the case of Lateral Raise (LR) exercise. An overall performance comparison of the SVM model and CNN_Model1

Table 5. Performance evaluation measures of CNN_Model1

	Exercise Type	Acronym	Precision	Recall	F1-score
Upper-Body LME exercises	Bicep Curls	BC	1.0000	1.0000	1.0000
	Frontal Raise	FR	0.9052	0.9552	0.9296
	Lateral Raise	LR	0.9273	0.9105	0.9188
	Triceps Extension	TER	0.9962	1.0000	0.9981
	Pec Dec	PD	0.9850	0.9990	0.9920
	Trunk Twist	TT	0.9962	0.9990	0.9976
Lower-Body LME Exercises	Standing Bicycle Crunches	SBC	0.9921	0.9600	0.9758
	Squats	SQ	0.9814	0.9552	0.9681
	Leg Lateral Raise	LLR	0.9209	0.9867	0.9526
	Lunges	L	0.9748	0.9952	0.9849
Common Movements	Others	OTH	0.9868	0.8991	0.9409

**Figure 16.** Normalized confusion matrix for CNN model with AlexNet architecture

for upper-body and lower-body exercises is shown in Figure 17. The CNN_Model1 outperformed the SVM model in both the upper-body LME exercises and the lower-body LME exercises.

4.3. Exercise Recognition with CNN_Model2 using Time-Series Arrays

The third method, termed CNN_Model2, used in the comparative study of the exercise recognition task (Figure 5) was the model that we designed and built from scratch using the deep CNN concept. The motivation to build CNN_Model2 was to process the raw time-series data from the wearable sensor through deep network without converting to images. Though the results observed from CNN_Model1 were very good for the exercise recognition task, it was interesting to test a time-series data classification model for exercise recognition. Figure 18 represents the pipeline used in the development of a CNN model from scratch which uses wearable sensor data in the form of six 1-D time-series vector arrays. The CNN_Model2 was constructed using the time-series array data from the INSIGHT_LME data set (Section 3.4). Data segmentation and processing, model construction and the obtained results are discussed in the following sections.

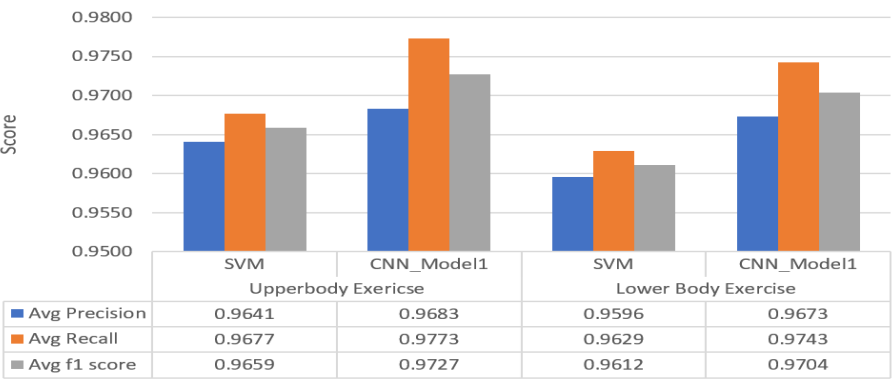


Figure 17. Statistical parameter comparison for CNN_Model1 and SVM models

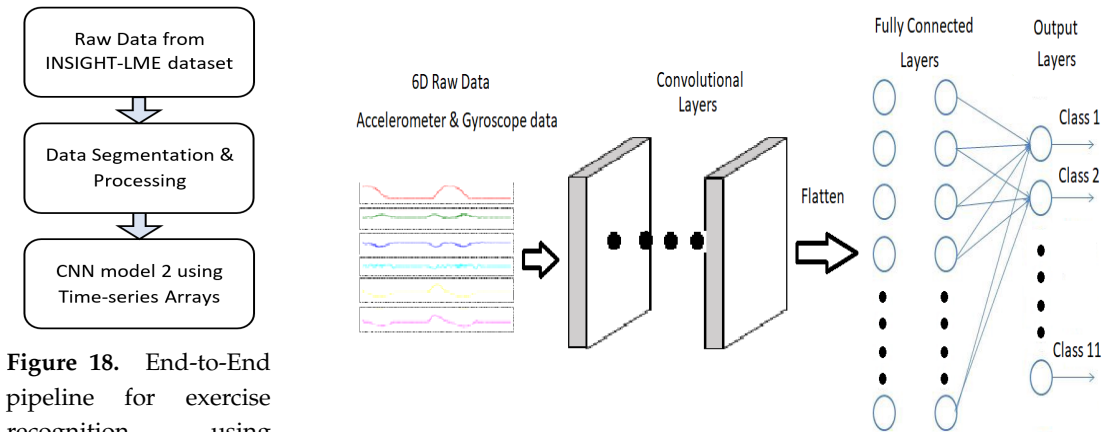


Figure 18. End-to-End pipeline for exercise recognition using CNN_Model2

Figure 19. CNN_Model2 Architecture for exercise recognition

4.3.1. Data Segmentation and Preprocessing

Data segmentation was achieved using the sliding window method with a window length of 4seconds having an overlap of 0.5second. A 6D time-series data array of size (2048 x 1 x 6) was formed from every segmented frame. A 6D time-series array comprises six 1D time-series arrays corresponding to data from each axis of 3D accelerometer and 3D gyroscope. 6D time-series data set for the training, validation and testing were prepared from the training set, validation set and test set of the INSIGHT_LME data set and the class label information were retained in each data set. The data set was then used by CNN_Model2 for the exercise recognition task.

4.3.2. CNN_Model2 Architecture for Exercise Recognition

Figure 19 illustrates the base architecture of the CNN_Model2 classifier for the recognition task. The deep network consists of seven 2D convolutional layers in addition to an input layer, two fully connected layers and a dropout layer. In each convolution layer, the convolution operation was followed by batch normalization, activation and max-pooling operations. The output of the seventh convolution layer was flattened and applied to fully connected layers with a ReLU activation function. A dropout of 0.25 was used, to prevent overfitting, before connecting to an output layer. A fully-connected output layer with a softmax activation function was used which was capable of classifying output into 11 classes. The number of filters used in seven convolution layers were 16, 16, 32, 32, 64, 64 and 96 respectively. The selection of the number of convolutional layers and the number of filters in each layer of the CNN_Model2 architecture were arrived after the initial few trials with different configurations. Table A2 of Appendix A lists the complete architecture parameters for the CNN_Model2.

The CNN_Model2 was constructed using Keras API with the TensorFlow back end with the choice of optimizer function among SGD, Adam and RMSprop. Similarly, a best learning rate was selected by training the model over a range of $1e-03$ to $1e-10$ with a decay of $1e-01$. The models were optimized using the loss functions categorical cross-entropy and Kullback-Leibler divergence to have lower losses. Early stopping while the model building by monitoring validation loss and learning rate scheduler using the ReduceOnPlateau function from Keras was effectively used. Data augmentations like shearing, resizing, flipping, rotation were not performed on the time-series data. Models were trained and validated using 6D array data set from training dataset and validation data set (Section 4.3.1). A model with a minimum validation loss and with the best validation accuracy was selected as CNN_Model2 and this model was further tested using the test dataset.

4.3.3. Experimental Results of CNN_Model2

CNN_Model2 having Adam optimizer with a learning rate $1e-7$ with KLD loss function was found to be the best model recorded with an overall training score of 96.89% and a validation score of 88.97%. The model recorded an overall test accuracy of 95.61% with an overall F1-score measure of 96% for the test dataset. The CNN_Model2 performance was very much comparable with both SVM model from supervised ML model (Section 4.1) and CNN_Model1 (Section 4.2).

Table 6. Performance evaluation measures of CNN_Model2

Exercise Type		Acronym	Precision	Recall	F1-score
Upper-Body LME exercises	Bicep Curls	BC	0.9952	0.9713	0.9831
	Frontal Raise	FR	0.8917	0.9574	0.9234
	Lateral Raise	LR	0.9389	0.9178	0.9283
	Triceps Extension	TER	0.9985	1.0000	0.9992
	Pec Dec	PD	0.9953	0.9837	0.9895
	Trunk Twist	TT	0.9721	0.9977	0.9847
Lower-Body LME Exercises	Standing Bicycle	SBC	0.9834	0.9651	0.9742
	Squats	SQ	0.9874	0.9698	0.9785
	Leg Lateral Raise	LLR	0.9771	0.9907	0.9838
	Lunges	L	0.8917	0.9574	0.9234
Common Movements	Others	OTH	0.8975	0.8389	0.8672

The performance of CNN_Model2, in terms of statistical parameter measurements such as precision, recall and F1-score, for individual exercise were measured and tabulated in Table 6. A comparative representation of all three models with relative statistical measures for the upper-body and lower-body LME exercises were shown in Figure 20. These statistical evaluation measures obtained from the CNN_Model2 were compared with the results obtained from the SVM model (the best performed supervised ML model) and CNN_Model1. The overall performance of the CNN_Model2 was found to be better than the SVM model performance. However, the overall performance of CNN_Model1 was found to be better compared to CNN_Model2.

4.4. Summary of Comparative Study of Models for Exercise Recognition

The following conclusions can be drawn from the model comparison. Among the supervised ML models, models that were constructed with the dimensionality reduction using PCA were observed with inferior performance compared to the models without PCA. Among the four ML models without PCA, the SVM model performed the best. The deep CNN model, the CNN_Model1, using the AlexNet architecture found to be the single best model when compared with the SVM and CNN_Model2, and outperformed both SVM and CNN_Model2 in terms of overall test accuracy, precision, recall and F1-score values. The second deep CNN model, the CNN_Model2, was found to be the second best and outperformed the SVM model in terms of overall performance over test accuracy, precision, recall and F1-score measures. The models were compared for their performances on the upper-body

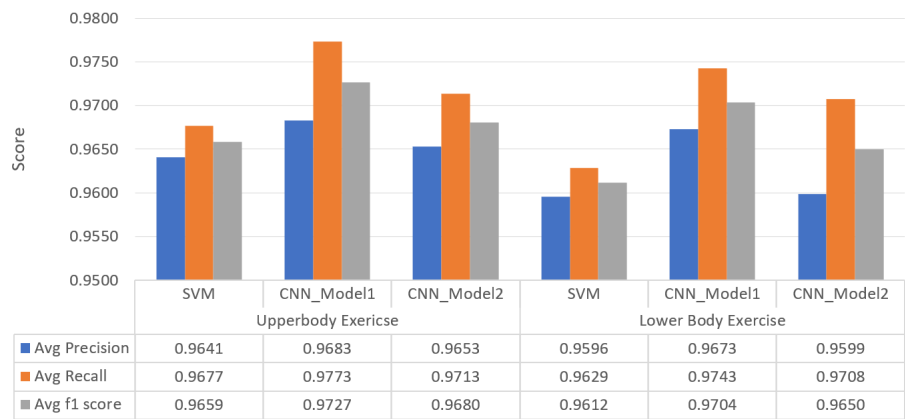


Figure 20. F1-score comparison of SVM, CNN_Model1 and CNN_Model1 for Upper-Body and Lower-Body LME exercises

LME exercise recognition and lower-body LME exercise recognition and observed that CNN_Model1 outperforms the other two models

4.5. Exercise Repetition Counting with Peak Detection Method

The first method, among three, investigated for exercise repetition counting was a signal processing method based on peak detection. The concept of the peak detection method lies in the identification of the peaks corresponding to maximum or minimum signal strength of any periodic time-series data. Figure 21 represents the end-to-end pipeline used for peak detection and counting repetitions using peak information. Raw data from the INSIGHT-LME data set corresponds to 3D accelerometer and 3D gyroscope recordings for limb movement having for each of the exercises. Each exercise type exhibits different signal patterns on the different sensor axes and the signal strengths on any given axes is proportional to the plane of limb movement. The periodicity of the signal observed on any significant axis of the sensor was used in the peak detection after completion of the exercise recognition task. Hence, 10 peak detectors were used, one for each exercise. The raw data from all the participants from the INSIGHT-LME dataset was used here to count the number of repetitions for each of the exercises. Data processing, filtering, peak detection and counting are discussed in the following section.

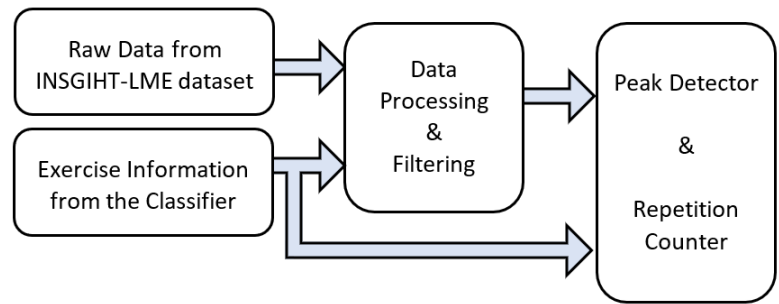


Figure 21. Pipeline for repetition counting using a peak detector

4.5.1. Data Processing and Filtering

6D time-series data from INSIGHT-LME dataset were the information obtained from each participant while exercising . The signal pattern variations in all the three axes of the accelerometer and the gyroscope represents the significant translatory motion and rotary motion respectively. While exercising, repetitions are reflected in the periodicity in the signal patterns on these axes of the sensors. The signal amplitude on each axis represents the significance of limb movement in any particular

direction. However, these signals were affected with the inherent noise introduced by the sensor. To understand and retrieve these signal variations and to calculate repetitions, the raw data was first processed and filtered.

The first step is to identify a dominant sensor axis for individual exercise and use this signal in peak detection. The dominant sensor axis in the plane of limb movement was evaluated using the mean square values of acceleration measurements from all the three axes of the accelerometer and the mean square values of the rotation rate from all the three axes of the gyroscope.

Table 7. The sensor and the dominant axis information for Individual LME Exercises

Exercise Type		Acronym	Sensor used & Dominant Axis
Upper-Body LME Exercises	Bicep Curls	BC	Accelerometer: X – Axis
	Frontal Raises	FR	Accelerometer: X – Axis
	Lateral Raises	LR	Accelerometer: X – Axis
	Triceps Extension Right	TER	Accelerometer: X – Axis
	Pec Dec	PD	Gyroscope: X – Axis
	Trunk Twist	TT	Gyroscope: Y – Axis
Lower-Body LME Exercises	Standing Bicycle Crunch	SBC	Gyroscope: X – Axis
	Squats	SQ	Accelerometer: X – Axis
	Leg Lateral Raise	LLR	Accelerometer: Y – Axis
	Lunges	L	Accelerometer: X – Axis

Table 7 summarizes the sensor and the dominant axis information used in repetition counting for repetition counting for individual LME exercises from the mean square value evaluation. For each exercise, the observed plane of movement of the right-wrist of the participant exercising were matched with the calculated dominant sensor axis using the mean square method. Signal plots of 3D accelerometer and 3D gyroscope for all the exercises are shown in Figure A1 of Appendix B and Figure A2 of Appendix C respectively.

Dominant axis signals were smoothed to remove the possible noise using a low pass Savitzky-Golay filter. The Savitzky-Golay filter removes high-frequency noise and has the advantage of preserving the original shape and features of the time-series signal. A window of 1023 samples and a filter order 4 was used.

4.5.2. Peak Detection and Repetition Counting

The peak detector detects both positive peak and negative peak values from the input time-series signal. The peak detector uses a threshold value based on the exercise type and from the input dominant signal and then using this threshold value it calculates two threshold points, an upper threshold point and a lower threshold point. Depending upon initial signal slope direction, a first max value or a first min value was determined. With an initial max/min value was determined, the detector next finds the next min/max value using the lower/upper threshold point and the search continues for the next max/min with the help of upper/lower threshold point. A pair of max-min value was used as an increment counter in the repetition counting. Figure 22 represents the filtered dominant accelerometer x-axis signal plotted with both positive and negative peaks using peak detector. A total of ten different peak detectors were used, one for each individual exercise.

4.5.3. Experimental Results of repetition counting using peak detectors

All the input data signals from the INSIGHT-LME dataset were used in testing to evaluate the overall performance of the peak detectors. The number of error counts, i.e. the difference between the actual number of repetition counts and the number of detected counts, was recorded in each case. Table 8 shows the results of repetition counting for individual LME exercise in terms of the number of errors with that of the actual count using peak detection method.

The table also indicates the total number of subjects that were used in testing each exercise. The **Error Count**, or 'e | X |', was used to show the number of participants having ' | X | ' number of error

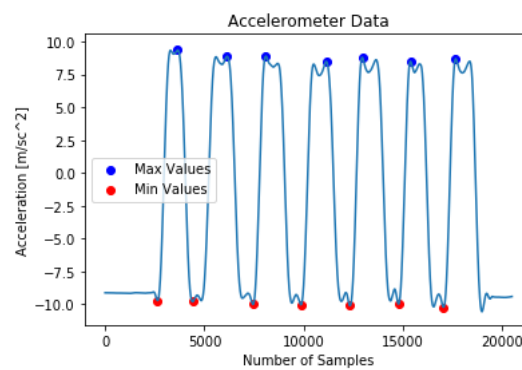


Figure 22. An example of repetition counting for Bicep Curls on the filtered dominant signal from the x-axis of the accelerometer sensor

count/s, where ' $|X|$ ' was used to for absolute error count in terms of 0, 1, 2 or more than 2 errors. The peak detector method used for the repetition counting performed better for upper-body exercises like, BC, FR, LR and TER in comparison to the repetition counting of the lower-body exercises. Figure 23 indicates the performance and the percentage of error in the repetition counts for each exercise. For example, from Table 8, for Bicep Curls an upper-body LME exercise, repetition counting without any error were reported for 144 instances among 151 subject trials. However, for 7 subject trials ± 1 error count was reported.

Table 8. Number of error counts in the repetition using Peak Detector Method

Exercise Type	Exercise	Acronym	Total Subjects	Error Count			
				$e 0 $	$e 1 $	$e 2 $	$e> 2 $
Upper-Body LME Exercises	Bicep Curls	BC	151	144	7	0	0
	Frontal Raises	FR	151	140	11	0	0
	Lateral Raises	LR	150	141	9	0	0
	Triceps Extension Right	TER	152	143	9	0	0
	Pec Dec	PD	149	120	8	3	18
	Trunk Twist	TT	151	128	14	5	4
Lower-Body LME Exercises	Standing Bicycle Crunch	SBC	149	132	8	4	5
	Squats	SQ	146	63	11	6	66
	Leg Lateral Raise	LLR	149	73	10	18	48
	Lunges	L	147	11	9	13	114

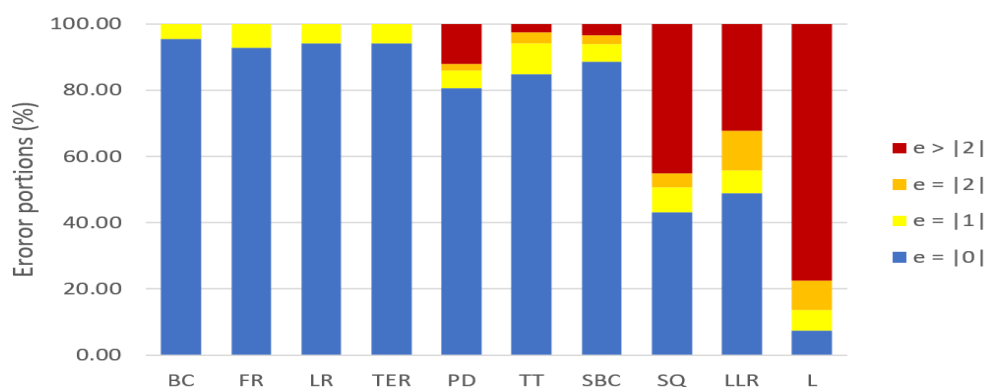


Figure 23. Number of errors of the repetition counting using a peak detector

4.6. Exercise Repetition Counting with CNN_Model1 using AlexNet Architecture

The second approach investigated for repetition counting was a deep CNN model, using the same CNN architecture as previously i.e. CNN_Model1. We compare a single deep CNN model for the

repetition counting task of all the exercises as opposed to the the use of multiple CNN models as used in [26]. Figure 24 illustrates the pipeline used for the repetition counting task using the CNN_Model1 as a binary classifier along with an additional repetition counter block.

Inspired by the signal processing approach to repetition counting, CNN_Model1 uses the peak information from the signals. However, the CNN_Model1 uses a binary classifier for the repetition counting instead of 11 class classifier as in the case of the exercise recognition task (Figure 14). The output of the binary classifier using the CNN_Model1 was given to a repetition counter which counts the total repetitions for any given exercise.

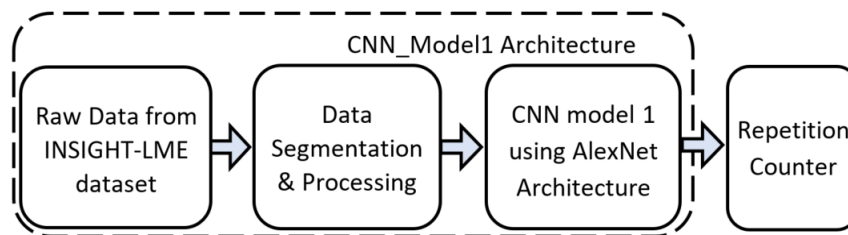


Figure 24. Pipeline for repetition counting using CNN_Model1

4.6.1. Data Segmentation & Processing

We used the image dataset created with a 4-second sliding window from section 4.2.1 and created new target label information from the dominant axis information from section 4.5.1. New labels for “Peak” and “NoPeak” were appended to the each exercise class image data using dominant signal information. Binary target class label information was generated using a grid of 50% width of the image and if the peak of the dominant axis signal plot in the image lies on the left half of the vertical-axis of the grid then the image was labelled with “Peak” (“1”) otherwise the image was labelled with “NoPeak” (“0”).

Figure 25 explains the labelling process by observing dominant axis signal plot information for the Bicep Curls exercise. in Figure 25, two 2D image plots corresponding to two different 4-second segments of 6D sensor signals of Bicep curls were labelled using the dominant axis information. Exercise data of 25 seconds corresponding to one exercise, when segmented using a 4-second window having an overlap of 0.5 seconds, resulted in a sequence of 43 images. The top most plot refers to the accelerometer x-axis which was the dominant axis for the Bicep Curls exercise. The training image data set with binary class label information was formed to have all images corresponding to all the exercise data from the training set of INSIGHT-LME dataset. Similarly, the validation and the test image data-sets are generated for the binary classification.

4.6.2. CNN_Model1 as a Repetition Counter

Models were trained with the training dataset of the newer image dataset with binary class label information and validated with the validation results. Models were built to have optimum parameters with variation in learning rate and selection of optimizer as discussed in section 4.2.2. We used a binary cross-entropy loss function while training all models and the best model was selected based on the validation score evaluation.

Repetition counting was done by testing a sequence of 43 images corresponding to a 25 second exercise data. The predicted result, from the model, on each image of the sequence was recorded and used in the repetition counter. A repetition counter counts the total number of transitions from “Peak” to “NoPeak” (“1” to “0”) and from “NoPeak” to “Peak” (“0” to “1”). The total repetition count corresponds to half the number of total transitions and Figure 26 illustrates the counting from the prediction labels .

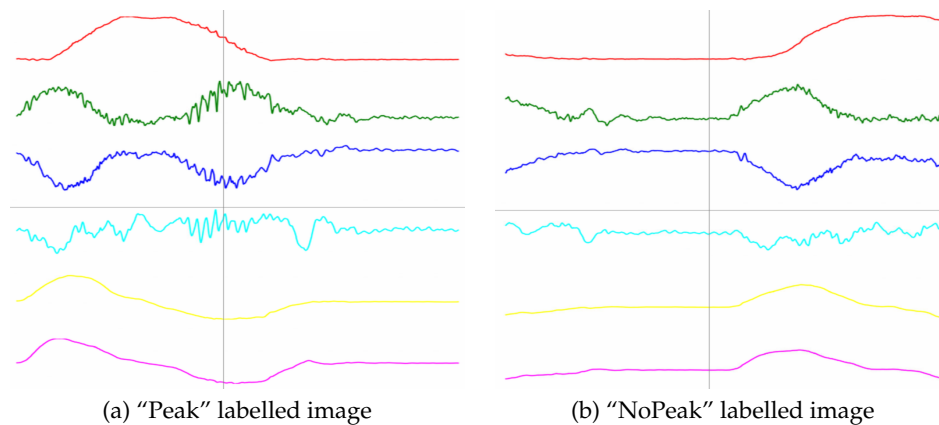


Figure 25. Example image with repetition class information for repetition detection on 4-second segmented images of Bicep Curls (a) “Peak” labelled image of Bicep Curls, (b) “NoPeak” labelled image of Bicep Curls

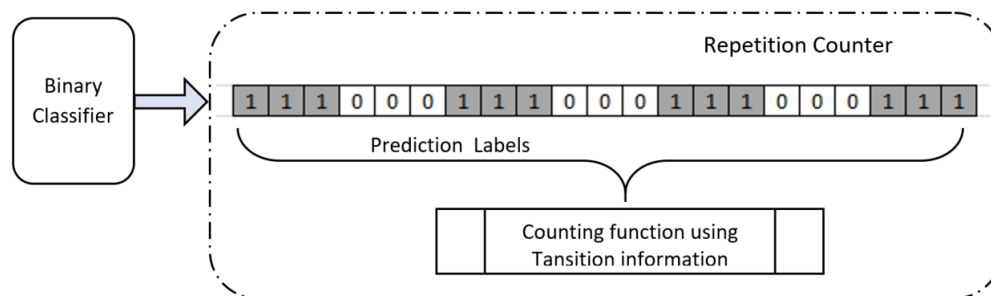


Figure 26. Repetition Counter

4.6.3. Experimental Results of repetition counting using CNN_Model1

The optimization of parameters were selected based on lowest validation loss measures and the optimum CNN_Model1 for the repetition counting task was with Adam optimizer and with a learning rate of $1e-5$. The model was further tested with the test dataset images. The test data set corresponds to the data from fifteen participants and each exercise was performed twice by each participant resulted with a total of 30 exercises data for each exercise.

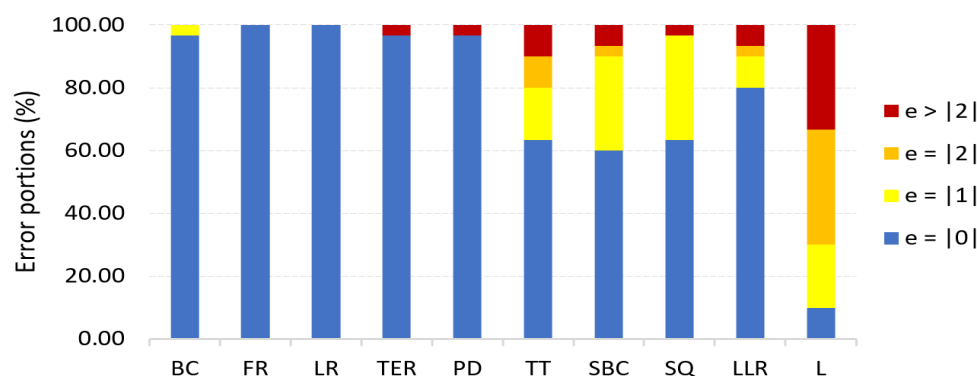
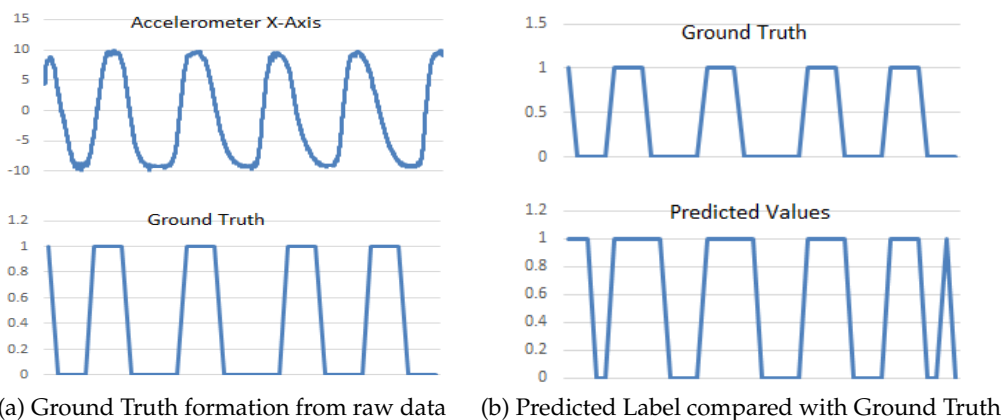
Table 9 shows the result of the repetition counting for individual LME exercises in terms of the number of errors with that of the actual count. The overall performance of the model in the repetition counting using a single AlexNet architecture based CNN model was very accurate for most of the upper-body LME exercises. However, for the lower-body exercises the repetition count performance for LLR was 80% and was better compared to the performance with other lower-body exercises. For the lunges, the model performance was poorest in the repetition counting.

Figure 27 shows the percentage-wise result of the model in terms of the repetition count errors for the individual exercises. The performance of the model for the upper-body LMEs like: FR and LR, it was 100%. For other upper-body LMEs like: BC, TER, PD correct counting was 96.67%. In the case of lower-body LME exercises, for LLR the correct counting was 80.0% . For other exercises the performances of the model with zero error count was poor. However, the overall count performance of CNN_Model1 was better for most of the exercises, when compared to the repetition counting using the signal processing model (Figure 23).

A significant portion of the error counting was related to the single count error. Ground truth and predicted count patterns were studied to understand the reasoning for most of the single count errors. Figure 28(a) and Figure 28(b) depicts 25 seconds of Bicep curls exercise data representation with the ground truth compared with that of predicted output from the model. Most of the single count error from the predicted signal were due to the binary classifier misclassifying the transition state peaks,

Table 9. Number of error counts in the repetition using CNN_Model1

Exercise Type	Exercise	Acronym	Total Subjects	Error Count			
				e 0	e 1	e 2	e> 2
Upper-Body LME Exercises	Bicep Curls	BC	30	29	1	0	0
	Frontal Raises	FR	30	30	0	0	0
	Lateral Raises	LR	30	30	0	0	0
	Triceps Extension Right	TER	30	29	0	0	1
	Pec Dec	PD	30	29	0	0	1
	Trunk Twist	TT	30	19	5	3	3
Lower-Body LME Exercises	Standing Bicycle Crunch	SBC	30	18	9	1	2
	Squats	SQ	30	19	10	0	1
	Leg Lateral Raise	LLR	30	24	3	1	2
	Lunges	L	30	3	6	11	10

**Figure 27.** Number of errors of the repetition counting using CNN_Model1**Figure 28.** Illustration for single count error tolerance. (a) Ground truth formation from the dominant signal information and (b) Comparison of predicted labels with the ground truth

at the beginning or at the end of 25-second signal segment. Similar observations of the single error counts were observed with different exercises.

4.7. Exercise Repetition Counting with CNN_Model2 using Time-Series Arrays

The third model used in the repetition counting task was again a deep CNN model using the CNN_Model2 architecture which uses 6D time-series array data from the raw signal for binary classification. Figure 29 represents the pipeline using the CNN_Model2 (Figure 19 architecture in the repetition counting task. This method used was again uses a single CNN model for the repetition counting instead of the multiple CNN models used by [26].

The CNN_Model2 used here includes a binary classifier for repetition counting instead of an 11 class classifier model used in the case of the exercise recognition task (section 4.3. The data processing

method uses the 6D time-series array data set from section 4.3.1 along with the new binary labelling information generated from section 4.6.1. The data processing and segmentation associated is further discussed in data segmentation and processing section below. The output of the CNN_Model2 was given to a repetition counter to count the total repetition counts for any given exercise.

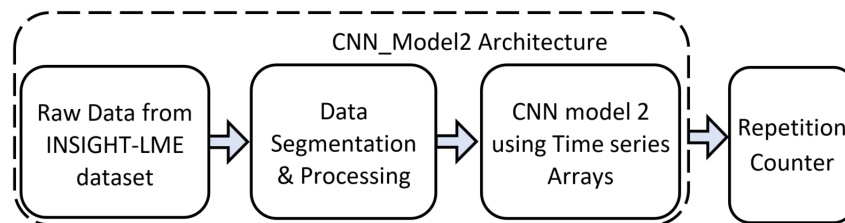


Figure 29. Pipeline for repetition counting using CNN_Model2

4.7.1. Data Segmentation & Processing

Additional binary target class label information from section 4.6.1 was appended on the 6D time-series array dataset generated with a 4-second sliding window from section 4.3.1. 25 seconds of sensor data, corresponding to the individual exercises, when segmented using a 4-second windowing method with 0.5 second overlap results into a total of 43 6D time-series array segments. Newly labeled data sets for the training, the validation and the testing data sets from the INSIGHT-LME dataset were used in the training, validation and testing the binary classifier.

4.7.2. CNN_Model2 as a binary classifier and Repetition Counter

The binary classifier models with CNN_Model2 architecture were trained with the training dataset of the newer 6D time-series array dataset with the binary label information and were validated with the validation results. The models were built to have optimum parameters with variation in learning rate and selection of optimizers among SGD, Adam and RMSprop along with the binary cross-entropy loss function. Testing the model with exercise data corresponding to a single participant was about testing the sequence of all 43 6D time-series array segments. The predicted results from the model, for these sequence of arrays from the test dataset was used in the repetition counting by a repetition counter. Similar to the section 4.6.2, the repetition counting was done by calculating the number of transition from “Peak” to “NoPeak” (“1” to “0”) and from “NoPeak” to “Peak” (“0” to “1”). The total number of repetition counts were equal to half the number of total transitions. The total repetition count corresponds to half the number of total transitions and Figure 26 illustrates the counting from the prediction labels .

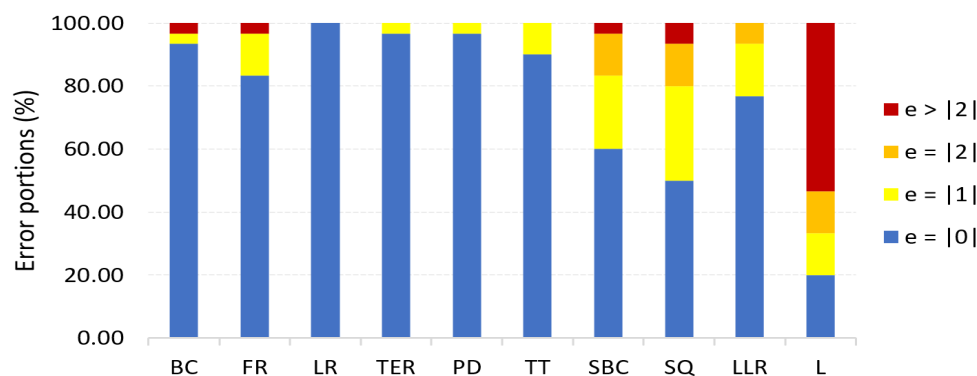
4.7.3. Experimental Results of repetition counting using CNN_Model2

The optimum model was selected based on the validation score and was with an Adam optimizer, and a learning rate of 1e-06. The optimum model was further tested with the test data set. The test data set consists of total 30 exercise data from each exercise type corresponding to the fifteen participants performing each exercise twice.

Table 10 shows the results of repetition counting for individual LME exercise in terms of the number of errors with that of the actual count. Overall performance of repetition counter using a single CNN_model2 of 6D time-series array data is good for upper-body LME exercises. However, for lower-body exercises the repetition count performances are average one. For lunges, a lower-body exercise, the repetition counting is the poorest. Figure 30 represents the plot of error portions of error counts for each exercise. we could achieve 100% correct counting only in the case of LR exercise trials. However, the count performances were better in comparison with the signal processing based peak detection method. However, the repetition counting was very poor in the case of Lunges.

Table 10. Number of error counts in the repetition using CNN_Model2

Exercise Type	Exercise	Acronym	Total Subjects	Error Count			
				e 0	e 1	e 2	e> 2
Upper-Body LME Exercises	Bicep Curls	BC	30	28	1	0	1
	Frontal Raises	FR	30	25	4	0	1
	Lateral Raises	LR	30	30	0	0	0
	Triceps Extension Right	TER	30	29	1	0	0
	Pec Dec	PD	30	29	1	0	0
	Trunk Twist	TT	30	27	3	0	0
Lower-Body LME Exercises	Standing Bicycle Crunch	SBC	30	18	7	4	1
	Squats	SQ	30	15	9	4	2
	Leg Lateral Raise	LLR	30	23	5	2	0
	Lunges	L	30	6	4	4	16

**Figure 30.** Number of errors of the repetition counting using CNN_Model2

A significant amount of error count for upper-body LME exercises was with one count error. As discussed in section 4.6.3 the single error count cases were mainly observed with the signal transitions at the beginning or the end of the 25 seconds signal samples. With a tolerance of one count error (i.e. blue + yellow, Figure 30) performance of the model were almost comparable to that of CNN_Model1. In comparison to the CNN_Model1, CNN_Model2 performance was inferior but comparable.

4.8. Summary of Comparative study of models for Exercise Repetition Counting

We studied three different methods for the exercise repetition counting task: the peak detector, CNN_Model1 and CNN_Model2. The peak detector method, was a signal processing approach where we designed ten different peak detectors based on the dominant sensor-axis signal information, one for each exercise. The method was a dependent model on the exercise recognition task and therefore was used only after completion of the exercise recognition task as a follow-up block processing method. The fact that the requirement of ten different peak detectors one for each exercise recognition and a follow-up sequential block processing makes it a weak contender though it has an overall good repetition counting output.

The second model, the CNN_Model1 was built based on AlexNet architecture, and was a single deep CNN model used in the repetition counting in comparison with any state of the art CNN models proposed in repetition counting for a set of exercises. The CNN_Model1 found to have extremely good performance for seven out of ten exercises and also the overall a single best deep CNN model studied here, capable of counting repetitions with all the exercises. Overall performance of CNN_Model1 was extremely better for the upper-body LMEs.

The third model, CNN_Model2, which processes raw time-series 6D signals was a single deep CNN model capable of counting repetition of all the exercises. The CNN_Model2 was also a second single deep CNN model that we are proposing as the single model which performs the repetition counting task of all exercises from a set of exercises. This model performance was the overall second

best in the repetition counting task and performed better for the upper-body LMEs similar to the CNN_Model1. However, the CNN_Model1 was the single best model that could be used for the repetition counting task among all three models.

5. Conclusions and Future Work Scope

Exercise detection and repetition counting task using a single wearable device with the state of the art algorithms is a key technological demand in the area of e-Health, e-Rehabilitation methods. In this paper, we presented the exercise recognition and repetition counting of LME exercises used in cardiovascular disease rehabilitation using different methods based on machine learning and deep learning. No public dataset was available with a single sensor wearable device specifically for the exercises used in cardiovascular disease rehabilitation which can be used on eHealth, mHealth platforms. We created the INSIGHT-LME dataset using a single wrist-worn wearable devices under the supervision of health experts from the sports clinic and with the guidance from clinical staff. The new dataset will encourage further research in the field of application using single wrist-worn inertial sensor in exercise-based rehabilitation.

Our study was to find a single best machine learning or deep learning method, individually in the exercise recognition and repetition counting. In the first comparative study, we studied the exercise recognition task, using the supervised machine learning models using SVM, RF, kNN and MLP with and without dimensionality reduction using PCA. SVM model was found to be the single best performing supervised ML model with an overall accuracy measure of 96.07%. In addition to the supervised models, two deep CNN models were studied, the CNN_Model1 based on the AlexNet architecture and the CNN_Model2 a custom-built to handle time-series array data and these models are compared with the SVM model performance. The CNN_Model1 found to be the single best performing model for the exercise recognition task with an overall f1-accuracy score measure of 96.89%.

In the second comparative study, we studied repetition counting task using again three methods: repetition counting using a signal processing method, CNN_Model1 and CNN_Model2 methods. Signal processing method found to be the best in terms of accurate counting of repetition counts. However, this method can be used as a sequential processing method and can only be used after exercise recognition and requires 10 different algorithms for each type of exercise. CNN_Model1 using AlexNet architecture is the second-best repetition counting method having accurate counting for most of upper-body LME exercises. With a tolerance of ± 1 count error, the performance of CNN_Model1 is accurate in almost all exercises except for Lunges a lower-body LME exercise among 90% or repetition sets.

In future, this study can be used to enhance the model's capability to provide qualitative feedback based on the variations observed from the exercise data. We have studied the tasks of exercise recognition and repetition counting in offline mode with a windowing method using the smallest overlap of 0.5 seconds, which can lead further studies to the more real-time recognition and counting using wearables and on-device computations with limited memory availability. Two approaches studied here can be further studied for the hardware complexities and time complexity analysis if needed to be implemented on miniaturized wearable devices.

Author Contributions: “conceptualization, G.P., N.C. and K.M.; methodology, G.P., N.C. and K.M.; software, G.P.; validation, G.P., N.C. and K.M.; formal analysis, G.P.; investigation, G.P.; resources, G.P., N.C. and K.M.; data curation, G.P.; writing—original draft preparation, G.P.; writing—review and editing, G.P., N.C. and K.M.; visualization, G.P.; supervision, N.C. and K.M.; project administration, G.P., N.C. and K.M.; funding acquisition, N.C. and K.M.” ,

Funding: This work is supported by Science Foundation Ireland (SFI) under the Insight Centre award, Grant Number SFI/12/RC/2289, and AC-QUIS BI, an industrial partner of Insight Centre for Data Analytics, Dublin City University, Ireland.

Acknowledgments: We would like to acknowledge Dr Kevin McGuinness, Dr Sussanne Little, Dr Joseph Antony, Dr Amin Ahmadi, Dr Naresh Yarlapati, and Mr Venkatesh Gurram Munirathnam for their suggestions and inputs on certain aspects of the work carried out.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Model architecture for CNN_Model1 and CNN_Model2

CNN Model architecture used in the exercise recognition task for CNN_Model1 is given in Table A1 and for CNN_Model2 is given in Table A2 represents the number of layers and the parameters used. The same model with only output layer variation is used in the repetition counting task

Table A1. All architecture parameters for CNN_Model1. CL: Convolution Layer, DL: Dense Layer

Layer	Value	Parameters
Input Layer	227x227x3	0
Convolution Filters CL1	96	34944
Kernel Size CL1	(11, 11)	-
Strides CL1	(4, 4)	-
Pooling PL1	(3, 3)	0
Strides PL1	(2, 2)	-
Convolution Filters CL2	256	614656
Kernel Size CL2	(5, 5)	-
Strides CL2	(1, 1)	-
Pooling PL2	(3, 3)	0
Strides PL2	(2, 2)	-
Convolution Filters CL3	384	885120
Kernel Size CL3	(3, 3)	-
Strides CL3	(1, 1)	-
Convolution Filters CL4	384	1327488
Kernel Size CL4	(3, 3)	-
Strides CL4	(1, 1)	-
Convolution Filters CL5	256	884992
Kernel Size CL5	(3, 3)	-
Strides CL5	(1, 1)	-
Pooling PL3	(2, 2)	0
Strides PL3	(2, 2)	-
Dense DL1	4096	4198400
Dropout DL1	0.4	0
Dense DL2	4096	16781312
Dropout DL2	0.4	0
Dense DL3	1000	4097000
Dropout DL3	0.4	0
Batch Normalization	Yes	384+1024+1536+1536+
CL1, CL2, CL3, CL4, CL5, DL1, DL2, DL3		1024+16384+16384+4000
Activation function	ReLU	0
CL1, CL2, CL3, CL4, CL5, DL1, DL2, DL3		
Activation function DL2	softmax	0
Total Parameters	:	28,877,195
Trainable Parameters	:	28,856,059
Non-trainable Parameters	:	21,136

Appendix B. 3D Accelerometer Raw Data Signal Plots for all exercises

3D Accelerometer raw data signal plots for all the 10 LME exercises corresponding to the data from the wrist-worn sensor of one participant is shown in Figure A1.

Appendix C. 3D Gyroscope Raw Data Signal Plots for all exercise

3D Gyroscope raw data signal plots for all the 10 LME exercises corresponding to the data from the wrist-worn sensor of one participant is shown in Figure A2.

Table A2. All architecture parameters for CNN_Model2. CL: Convolution Layer and DL: Dense Layer

Layer	Value	Parameters
Input Layer	2048 x 1 x 6	0
Convolution Filters CL1	16	304
Kernel Size CL1	(3, 1)	-
Strides CL1	(1, 1)	-
Convolution Filters CL2	16	784
Kernel Size CL2	(3, 1)	-
Strides CL2	(1, 1)	-
Convolution Filters CL3	32	1568
Kernel Size CL3	(3, 1)	-
Strides CL3	(1, 1)	-
Convolution Filters CL4	32	3104
Kernel Size CL4	(3, 1)	-
Strides CL4	(1, 1)	-
Convolution Filters CL5	64	6208
Kernel Size CL5	(3, 1)	-
Strides CL5	(1, 1)	-
Convolution Filters CL6	64	12352
Kernel Size CL6	(3, 1)	-
Strides CL6	(1, 1)	-
Convolution Filters CL7	96	18528
Kernel Size CL7	(3, 1)	-
Strides CL7	(1, 1)	-
Batch Normalization CL1, CL2, CL3, CL4, CL5, CL6, CL7	Yes	64+64+128+128 +256+256+384
Activation function CL1, CL2, CL3, CL4, CL5, CL6, CL7	ReLU	0
Dense Layer DL1	128	25165952
Dropout DL1	0.25	0
Dense Layer DL2	11	1419
Activation function DL2	softmax	0
Total Parameters	:	25,211,499
Trainable Parameters	:	25,210,859
Non-trainable Parameters	:	640

References

1. Cardiovascular diseases (CVDs). <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-cvds>.
2. de la Cuerda, R.C.; Diego, I.M.A.; Martín, J.J.A.; Sánchez, A.M.; Page, J.C.M. Cardiac rehabilitation programs and health-related quality of life. State of the art. *Revista Española de Cardiología (English Edition)* **2012**, *65*, 72–79.
3. Pescatello, L.S.; Riebe, D.; Thompson, P.D. *ACSM's guidelines for exercise testing and prescription*; Lippincott Williams & Wilkins, 2014.
4. Dalal, H.M.; Zawada, A.; Jolly, K.; Moxham, T.; Taylor, R.S. Home based versus centre based cardiac rehabilitation: Cochrane systematic review and meta-analysis. *Bmj* **2010**, *340*, b5631.
5. Buys, R.; Claes, J.; Walsh, D.; Cornelis, N.; Moran, K.; Budts, W.; Woods, C.; Cornelissen, V.A. Cardiac patients show high interest in technology enabled cardiovascular rehabilitation. *BMC medical informatics and decision making* **2016**, *16*, 95.
6. Jackson, L.; Leclerc, J.; Erskine, Y.; Linden, W. Getting the most out of cardiac rehabilitation: a review of referral and adherence predictors. *Heart* **2005**, *91*, 10–14.
7. Guiraud, T.; Nigam, A.; Juneau, M.; Meyer, P.; Gayda, M.; Bosquet, L. Acute responses to high-intensity intermittent exercise in CHD patients. *Med Sci Sports Exerc* **2011**, *43*, 211–7.
8. Morris, D.; Saponas, T.S.; Guillory, A.; Kelner, I. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2014, pp. 3225–3234.

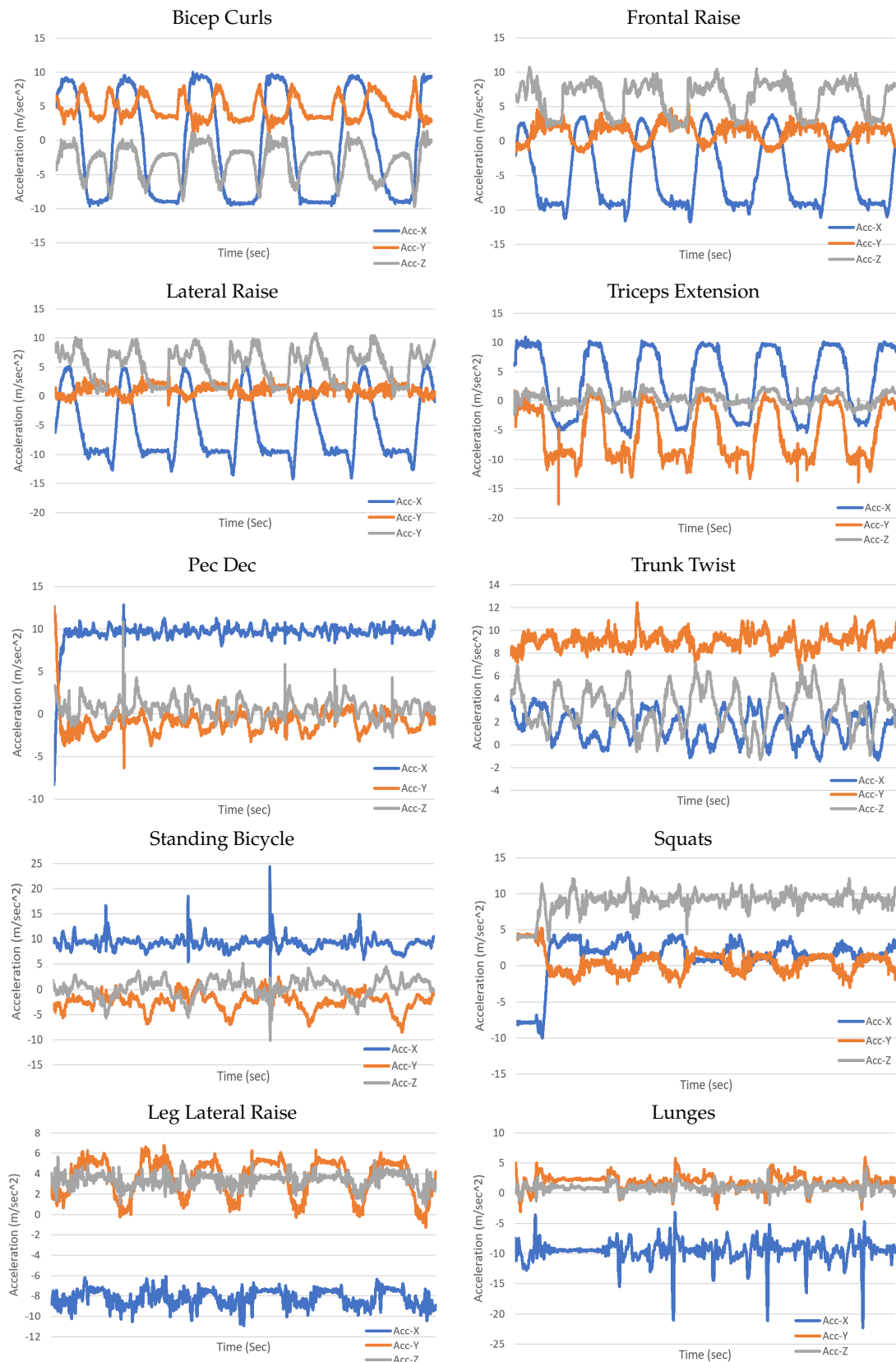


Figure A1. 25 seconds 3D accelerometer signal plots for all exercises

9. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials* **2012**, *15*, 1192–1209.

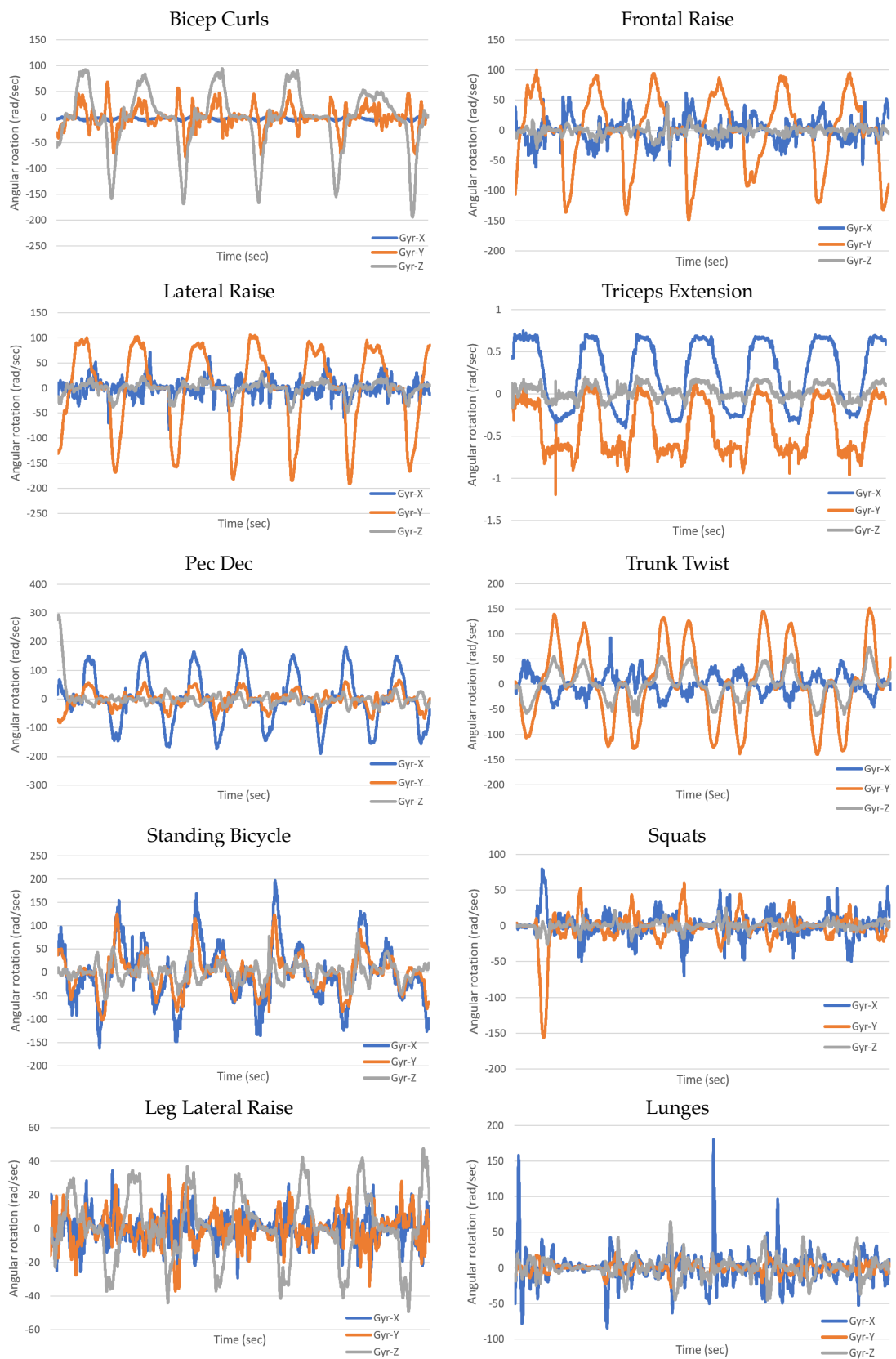


Figure A2. 25 seconds 3D gyroscope signal plots for all exercises

10. Nabian, M. A comparative study on machine learning classification models for activity recognition. *Journal of Information Technology & Software Engineering* **2017**, *7*, 4–8.

11. Ahmadi, A.; Mitchell, E.; Destelle, F.; Gowing, M.; O'Connor, N.E.; Richter, C.; Moran, K. Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors. 2014 11th International Conference on Wearable and Implantable Body Sensor Networks. IEEE, 2014, pp. 98–103.
12. Köping, L.; Shirahama, K.; Grzegorzec, M. A general framework for sensor-based human activity recognition. *Computers in biology and medicine* **2018**, *95*, 248–260.
13. Mannini, A.; Intille, S.S.; Rosenberger, M.; Sabatini, A.M.; Haskell, W. Activity recognition using a single accelerometer placed at the wrist or ankle. *Medicine and science in sports and exercise* **2013**, *45*, 2193.
14. Mannini, A.; Sabatini, A.M.; Intille, S.S. Accelerometry-based recognition of the placement sites of a wearable sensor. *Pervasive and mobile computing* **2015**, *21*, 62–74.
15. Szttyler, T.; Stuckenschmidt, H.; Petrich, W. Position-aware activity recognition with wearable devices. *Pervasive and mobile computing* **2017**, *38*, 281–295.
16. O'Reilly, M.A.; Whelan, D.F.; Ward, T.E.; Delahunt, E.; Caulfield, B. Classification of lunge biomechanics with multiple and individual inertial measurement units. *Sports biomechanics* **2017**, *16*, 342–360.
17. Ebert, A.; Beck, M.T.; Mattausch, A.; Belzner, L.; Linnhoff-Popien, C. Qualitative assessment of recurrent human motion. 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017, pp. 306–310.
18. Koskimäki, H.; Siirtola, P. Recognizing gym exercises using acceleration data from wearable sensors. 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, 2014, pp. 321–328.
19. Huang, B.; Giggins, O.; Kechadi, T.; Caulfield, B. The limb movement analysis of rehabilitation exercises using wearable inertial sensors. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2016, pp. 4686–4689.
20. Weiss, G.M.; Timko, J.L.; Gallagher, C.M.; Yoneda, K.; Schreiber, A.J. Smartwatch-based activity recognition: A machine learning approach. 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI). IEEE, 2016, pp. 426–429.
21. Sarcevic, P.; Pletl, S.; Kincses, Z. Comparison of time-and frequency-domain features for movement classification using data from wrist-worn sensors. 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 2017, pp. 000261–000266.
22. Chernbumroong, S.; Atkins, A.S.; Yu, H. Activity classification using a single wrist-worn accelerometer. 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings. IEEE, 2011, pp. 1–6.
23. Gupta, P.; Dallas, T. Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Transactions on Biomedical Engineering* **2014**, *61*, 1780–1786.
24. Catal, C.; Tufekci, S.; Pirmit, E.; Kocabag, G. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing* **2015**, *37*, 1018–1022.
25. Zhu, C.; Sheng, W. Recognizing human daily activity using a single inertial sensor. 2010 8th World Congress on Intelligent Control and Automation. IEEE, 2010, pp. 282–287.
26. Soro, A.; Brunner, G.; Tanner, S.; Wattenhofer, R. Recognition and repetition counting for complex physical exercises with deep learning. *Sensors* **2019**, *19*, 714.
27. Whelan, D.; O'Reilly, M.; Ward, T.; Delahunt, E.; Caulfield, B. Evaluating performance of the single leg squat exercise with a single inertial measurement unit. Proceedings of the 3rd 2015 Workshop on ICTs for improving Patients Rehabilitation Research Techniques, 2015, pp. 144–147.
28. Mortazavi, B.J.; Pourhomayoun, M.; Alsheikh, G.; Alshurafa, N.; Lee, S.I.; Sarrafzadeh, M. Determining the single best axis for exercise repetition recognition and counting on smartwatches. 2014 11th International Conference on Wearable and Implantable Body Sensor Networks. IEEE, 2014, pp. 33–38.
29. Bao, L.; Intille, S.S. Activity recognition from user-annotated acceleration data. International conference on pervasive computing. Springer, 2004, pp. 1–17.
30. Baños, O.; Damas, M.; Pomares, H.; Rojas, I.; Tóth, M.A.; Amft, O. A benchmark dataset to evaluate sensor displacement in activity recognition. Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 2012, pp. 1026–1035.
31. Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)* **2014**, *46*, 1–33.
32. Attal, F.; Mohammed, S.; Dedabrishvili, M.; Chamroukhi, F.; Oukhellou, L.; Amirat, Y. Physical human activity recognition using wearable sensors. *Sensors* **2015**, *15*, 31314–31338.

33. De, D.; Bharti, P.; Das, S.K.; Chellappan, S. Multimodal wearable sensing for fine-grained activity recognition in healthcare. *IEEE Internet Computing* **2015**, *19*, 26–35.
34. Whelan, D.; O'Reilly, M.; Ward, T.; Delahunt, E.; Caulfield, B. Evaluating performance of the lunge exercise with multiple and individual inertial measurement units. *Pervasive Health 2016: 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*, Cancun, Mexico, 16–19 May 2016. ACM, 2016.
35. O'Reilly, M.; Duffin, J.; Ward, T.; Caulfield, B. Mobile app to streamline the development of wearable sensor-based exercise biofeedback systems: system development and evaluation. *JMIR rehabilitation and assistive technologies* **2017**, *4*, e9.
36. Ding, H.; Han, J.; Shangguan, L.; Xi, W.; Jiang, Z.; Yang, Z.; Zhou, Z.; Yang, P.; Zhao, J. A platform for free-weight exercise monitoring with passive tags. *IEEE Transactions on Mobile Computing* **2017**, *16*, 3279–3293.
37. Pernek, I.; Kurillo, G.; Stiglic, G.; Bajcsy, R. Recognizing the intensity of strength training exercises with wearable sensors. *Journal of biomedical informatics* **2015**, *58*, 145–155.
38. Crema, C.; Depari, A.; Flammini, A.; Sisinni, E.; Haslwanter, T.; Salzmann, S. IMU-based solution for automatic detection and classification of exercises in the fitness scenario. *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2017, pp. 1–6.
39. Hatami, N.; Gavet, Y.; Debayle, J. Classification of time-series images using deep convolutional neural networks. *Tenth International Conference on Machine Vision (ICMV 2017)*. International Society for Optics and Photonics, 2018, Vol. 10696, p. 106960Y.
40. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* **2016**.
41. Yang, J.; Nguyen, M.N.; San, P.P.; Li, X.L.; Krishnaswamy, S. Deep convolutional neural networks on multichannel time series for human activity recognition. *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
42. Mohammad, Y.; Matsumoto, K.; Hoashi, K. Deep feature learning and selection for activity recognition. *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 930–939.
43. Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzec, M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors* **2018**, *18*, 679.
44. Veiga, J.J.D.; O'Reilly, M.; Whelan, D.; Caulfield, B.; Ward, T.E. Feature-free activity classification of inertial sensor data with machine vision techniques: method, development, and evaluation. *JMIR mHealth and uHealth* **2017**, *5*, e115.
45. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, pp. 1097–1105.