MM-Fit: Multimodal Deep Learning for Automatic Exercise Logging across Sensing Devices

DAVID STRÖMBÄCK, University of Edinburgh, UK
SANGXIA HUANG, R&D Center Lund Laboratory, Sony Europe, Sweden
VALENTIN RADU, University of Edinburgh, University of Sheffield, UK

Fitness tracking devices have risen in popularity in recent years, but limitations in terms of their accuracy and failure to track many common exercises presents a need for improved fitness tracking solutions. This work proposes a multimodal deep learning approach to leverage multiple data sources for robust and accurate activity segmentation, exercise recognition and repetition counting. For this, we introduce the **MM-Fit** dataset; a substantial collection of inertial sensor data from smartphones, smartwatches and earbuds worn by participants while performing full-body workouts, and time-synchronised multi-viewpoint RGB-D video, with 2D and 3D pose estimates. We establish a strong baseline for activity segmentation and exercise recognition on the MM-Fit dataset, and demonstrate the effectiveness of our CNN-based architecture at extracting modality-specific spatial temporal features from inertial sensor and skeleton sequence data. We compare the performance of unimodal and multimodal models for activity recognition across a number of sensing devices and modalities. Furthermore, we demonstrate the effectiveness of multimodal deep learning at learning cross-modal representations for activity recognition, which achieves 96% accuracy across all sensing modalities on unseen subjects in the MM-Fit dataset; 94% using data from the smartwatch only; 85% from the smartphone only; and 82% on data from the earbud device. We strengthen single-device performance by using the zeroing-out training strategy, which phases out the other sensing modalities. Finally, we implement and evaluate a strong repetition counting baseline on our MM-Fit dataset. Collectively, these tasks contribute to recognising, segmenting and timing exercise and non-exercise activities for automatic exercise logging.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Knowledge representation and reasoning**; **Neural networks**; **Learning latent representations**.

Additional Key Words and Phrases: multimodal learning, deep learning, exercise recognition, activity recognition, repetition counting, wearable, earbud, smartwatch, smartphone

ACM Reference Format:
David Strömbäck, Sangxia Huang, and Valentin Radu. 2020. MM-Fit: Multimodal Deep Learning for Automatic Exercise Logging across Sensing Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 168 (December 2020), 22 pages. https://doi.org/10.1145/3432701

1 INTRODUCTION

The popularity of fitness tracking devices has risen in recent years [34], enabling users to monitor their health and fitness levels through their smart devices. Current fitness trackers, however, are predominantly focused on tracking continuous high-movement aerobic activities, such as walking, running, and swimming. A small subset

Authors' addresses: David Strömbäck, University of Edinburgh, Edinburgh, UK, k.d.m.stromback@gmail.com; Sangxia Huang, R&D Center Lund Laboratory, Sony Europe, Lund, Sweden, sangxia.huang@sony.com; Valentin Radu, University of Edinburgh, University of Sheffield, UK, v.radu@shef.ac.uk.

2474-9567/2020/12-ART168 $15.00
https://doi.org/10.1145/3432701

of fitness trackers also support tracking body-weight and resistance exercises, but they are often restricted to only tracking exercises in which the fitness tracker is heavily involved in the exercise movements, thus often failing to track many common exercises. Furthermore, the reliability of current fitness tracking technology is raising more questions [5], with significant discrepancies being observed across different fitness tracking brands and models. Their main purpose is to keep the users engaged and in control of their fitness progress. Automatic exercise logging will bring benefits in tracking user exercises in any settings – working out from at home [17], or as part of physical rehabilitation programs.

In this work, we propose a multimodal deep learning solution for robust and accurate automatic exercise logging, which will contribute to advance the current fitness tracking technology. Our system consists of two main stages; an activity segmentation and exercise recognition stage, followed by a repetition counting stage. This work also makes contributions towards research into the use of multimodal deep learning for human activity recognition (HAR) across multiple devices, and for the relatively unexplored task of exercise repetition counting.

There are a number of factors that make HAR and repetition counting challenging. The high variation in how different people perform the same activity, and even the variation in how an individual performs an activity on different occasions, sets high demands on HAR and repetition counting systems. There are also modality specific challenges that arise for HAR and repetition counting tasks. Learning from sensor data, in particular from sensor data generated across multiple heterogeneous sensors, poses a number of challenges. The model and the brand of the sensor device, the positioning of the sensor, and the sampling frequency of the sensor, are a number of factors that can cause significant variations in sensor readings. Another limitation of sensor data for HAR and repetition counting is that signals are weaker when sensing device is not directly attached to the part of the body involved in the exercise, which makes some activities harder to distinguish. Finally, sensor drift is a problem that affects all sensors over time, and can lead to unreliable and deteriorating sensor reading accuracy.

To tackle these challenges, we build our solution on multimodal deep learning methods. Combining inertial sensor data from user-worn smart devices, assisted by available 3D skeleton sequence data extracted from RGB video, we hypothesise that these modalities will complement each other, and result in more robust and accurate HAR and repetition counting models than in isolation. The inertial sensors used in this work are accelerometer and gyroscope sensors (recording the acceleration and angular velocity, respectively) from smart wearable devices. The acceleration and angular velocity of body-worn devices change with the body movements, and are thus informative for inferring what activity a person is performing. The third modality we study in this work is 3D pose estimate data. Advances in 2D and 3D pose estimation have made it possible to acquire highly accurate and robust pose estimates in real-time [7]. 3D pose sequence information is highly discriminative for the task of activity recognition, and allows for a much more compact representation than raw image or depth data, resulting in more lightweight models. With the widespread availability of cheap cameras and inertial sensors, and the growing trend of ubiquitous computing, we are seeing an increase in the number of data modalities that are available across a wide range of settings. This contributes to the timeliness of this research, as we believe there are many settings where this is feasible (at the gym and at home). We use deep learning methods [29], due to their demonstrated ability to learn generalised hierarchical representations, which are important for dealing with the high intra-class variability found in HAR datasets. In addition, deep learning methods can be trained end-to-end directly on the raw data, avoiding the need for designing handcrafted features of shallower learning methods.

The contributions of this work are three-fold:

- We introduce the MM-Fit dataset, a large collection of time synchronised multi-view, multi-location on the participant body (inertial sensing) and ambient video sensor (RGB-D) streams, capturing the motion of full-body workouts. The MM-Fit dataset[1] is the first publicly available multimodal dataset for exercise recognition and repetition counting across multiple devices.

[1] https://mmfit.github.io

- We propose a multimodal deep learning framework that uses autoencoders to leverage unlabelled data. We fuse inertial sensor data and 3D pose estimate data for more accurate exercise recognition.
- We demonstrate that the performance of a single-device HAR model is boosted when training in the presence of other sensing perspectives – available only at training time, and eliminated at run-time – by using the zeroing-out training strategy.

2 RELATED WORKS

2.1 Human Activity Recognition

Human activity recognition approaches can be divided into two main categories, sensor-based methods [6, 9, 28, 51], relying on data from inertial measurement units (IMUs), such as accelerometers and gyroscopes, and video-based methods [1, 21, 39], operating on visual input data, such as colour and depth. The latter category also includes skeletal-based methods which take as input 2D or 3D human pose estimates, typically extracted from colour or depth data. Previous works have predominantly focused on unimodal learning, however, there are many real-world applications, such as exercise recognition, where it is feasible to leverage information from multiple modalities. Performance on HAR tasks has improved significantly in recent years, as research focus has shifted from using handcrafted features to using modern deep learning solutions [21]. For conciseness, we focus on the more recent and effective deep-learning approaches, and limit the scope to sensor-based and skeletal-based methods, as these are the modalities investigated in this work.

2.1.1 Inertial Sensor-Based HAR. There are a range of sensing modalities that have been used for HAR, with accelerometers and gyroscopes being the most common ones [51]. These sensors output multivariate time-series data, which are often segmented using the sliding-window approach, before being passed on to the feature extraction and classification stages. Traditional pattern recognition techniques, such as Hidden Markov Models (HMMs), decision trees, and Support Vector Machines (SVMs), have demonstrated strong results on various activity recognition tasks in controlled environments, however, their reliance on handcrafted features and ability to only learn shallow representations, restricts their performance and generalisability [28]. Deep learning approaches, on the other hand, can learn high-level features directly from the raw sensor data as part of an end-to-end system. Given sufficient amount of data, traditional machine learning and signal processing techniques are in large part outperformed by modern deep learning approaches [15]. Convolutional Neural Networks (CNNs) [30] are particularly well suited for working with sensor data, as they efficiently capture local dependencies in the data, and can learn scale-invariant features, important for dealing with activities performed at different rates of frequency [18, 54, 55]. Hammerla et al. carried out a comparison of different deep learning approaches for HAR, investigating Recurrent Neural Networks (RNNs), deep fully-connected networks, and CNNs on three public datasets. They found that for tasks where long-term dependencies play an important role, RNNs tend to perform best, whereas if detecting local patterns is of higher importance, as is the case for exercise recognition, CNNs are preferable [19]. These deep learning approaches require large training sets. Active learning has been used to reduce the amount of data that needs to be labelled without sacrificing accuracy [22]. Autoencoders and Restricted Boltzmann-Machines (RBMs) have also been applied to sensor data for HAR with unlabelled data [2, 40]. Radu et al. demonstrate their superior HAR performance compared to traditional shallow-learning approaches, by using a multimodal-RBM network to learn a shared representation for accelerometer and gyroscope data [40]. Here we explore a similar approach but across multiple devices and more complex modalities.

2.1.2 Skeletal-Based HAR. With the release of the Microsoft Kinect depth sensor and body tracking SDK in 2010, and the more recent development of real-time and accurate RGB-based human pose estimation techniques [7, 35, 42], the research area of skeletal-based HAR has emerged. Skeletal sequence data consists of body joint trajectories, and can, similarly to inertial sensor data, be viewed as multivariate time-series data. Early works

focus on constructing handcrafted features to obtain discriminative skeleton representations [14, 50, 52], with more recent approaches, again, focusing on deep learning.

A range of deep learning approaches have been explored for skeletal-based HAR, in particular, recurrent neural networks (RNNs) [13], graph neural networks (GNN) [33, 53], and CNNs [12, 25, 32]. RNNs are able to efficiently leverage time-series information, while GNNs are able to incorporate the spatial constraints and relationships of human joints in their models. Finally, CNNs have widely been demonstrated to perform well in learning representations at different abstraction levels, and at learning spatial temporal features. Ke et al. propose a 3D skeleton representation form using cylindrical coordinates to encode the relative joint positions with respect to selected reference joints [25]. They demonstrate the efficiency of this representation in combination with deep CNNs to learn spatial temporal features for HAR. Motivated by their success, we use their proposed skeleton representation form in this work. Wang et al., explore the integration of video and inertial sensors from a wrist worn device, using CNNs to extract features independently on each modality, and performing classification using a RNN[23].

2.1.3 Exercise Recognition. The growing popularity of smartwatches, earbuds and fitness-trackers has stimulated research interest in HAR for workout settings. Given that the vast majority of existing public datasets for exercise recognition, only contain inertial sensor data [36, 49], the majority of previous exercise recognition works are sensor-based. Chang et al. presented one of the earliest works that tackled exercise recognition [20]. They propose a Naive Bayes Classifier and a HMM approach using two triaxial accelerometers, to achieve 90% accuracy on a dataset with nine exercise classes. Morris et al. present an automated exercise logging system, consisting of three main stages; segmenting exercise and non-exercise segments, recognising exercises, and counting repetitions [36]. In contrast to our approach, Morris et al. use handcrafted features in combination with an SVM to classify exercises. They evaluate their system on the RecoFit dataset, a large-scale dataset consisting of accelerometer and gyroscope data collected from an arm-worn sensor. A number of more recent approaches have also explored deep learning based approaches for exercise recognition, in particular using CNNs [46, 48]. A video motion-based method, GymCam, proposed in [26], detects and counts exercise repetitions by identifying repetitive motions in videos with handcrafted optical flow features and a fully-connected classifier. They test their approach in an unconstrained and challenging environment, obtaining a 93.6% exercise recognition accuracy, and an average repetition count within 1.7 of the true count per exercise set. Other systems have considered the home WiFi radio as sensing modality for exercise recognition [17].

2.2 Multimodal Deep Learning

Multimodal deep learning is concerned with how to fuse different modalities, such that the learning task can best leverage the different data perspectives. Ngiam et al. propose a cross-modal RBM learning approach to learn better unimodal features by training on multiple modalities [38]. They demonstrate the efficiency of their approach by training on video and audio data, and testing on just a single modality, for the task of audio-visual speech classification. Alternative solutions for missing modalities have been explored, such as with adversarial autoencoders, which show good performance in generating the missing modality to be used in the recognition stage [44]. Radu et al. explore a number of fusing strategies for multimodal deep learning methods [41], comparing early feature concatenation and modality-specific architectures, on activity and context recognition tasks using inertial sensor data. Modality-specific architectures first learn unimodal features before learning a shared cross-modal representation. We develop a similar, but more complex multimodal neural network architecture in this work. [41] demonstrate that their proposed modality-specific CNN architecture outperforms the other explored approaches on three out of four tasks, and achieves an average accuracy that is 5% higher across all four tasks, compared to the early feature concatenation architecture. To specialise the activity recognition process to each

user, adversarial networks have been used [3], which rely on Siamese networks to decrease the variants between the representations of different subjects.

2.3 Repetition Counting

We focus on sensor-based repetition counting as this is the approach taken in this work, however, there are also a number of works for video-based repetition counting [31, 43]. The majority of sensor-based repetition counting works focus on counting exercise repetitions [10, 20, 37], and rely on signal-processing techniques. Chang et al. propose two approaches for repetition counting on triaxial accelerometer data; a peak counting algorithm, and a method using the Viterbi algorithm with a Hidden Markov Model (HMM). On a dataset of nine workout exercises they achieve a miscount rate of 5%. Choi et al. adopt a simple signal-processing technique to track repetitions using accelerometer sensors attached to gym exercise machines [10]. Their predictions are within one repetition of the ground-truth, 95% of the time. A recent deep learning approach proposed by Soro et al., leverage repetition-level annotations to regress the number of repetitions in a given input segment window. This is done using a CNN applied on inertial sensor data collected from a wrist-worn and an ankle-worn sensor [46]. Their model's predictions are within one of the true repetition count, 91% of the time, on a dataset with 10 CrossFit exercises. A limitation with their approach is that it requires having a separate repetition counting model for each exercise class, and having access to repetition level annotations.

3 MM-FIT DATASET

To enable research on multimodal learning for activity segmentation, exercise recognition and repetition counting, we have collected and annotated a multimodal dataset of participants performing full-body workouts. The MM-Fit dataset is made publicly available[2], with the hope that it will stimulate further research in these areas and be a valuable resource to the community. This section outlines the data collection and processing of the MM-Fit dataset.

3.1 Data Collection Overview

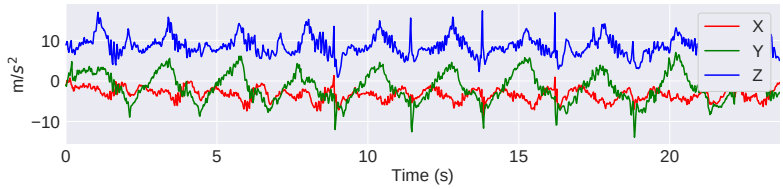
Participants performed full-body workout exercises in front of two depth cameras while wearing five differently-positioned smart devices collecting inertial sensor data. We restrict the task to single-person exercise recognition, so each workout session involves just one participant. This scenario is specific to home workouts. The workout sessions consist of three sets of ten exercises, with ten repetitions for each set. The following set of well-known resistance training exercises were chosen; squats, lunges (with dumbbells), bicep curls (alternating arms), sit-ups, push-ups, sitting overhead dumbbell triceps extensions, standing dumbbell rows, jumping jacks, sitting dumbbell shoulder press, and dumbbell lateral shoulder raises. The exercises were demonstrated to the participants before their workout session to ensure familiarity with each exercise, but no corrections to the participant's form were made during the workout. For the exercises involving dumbbells, the participants had access to two dumbbells with adjustable weight of up to 7.5kg each, and were free to choose how much weight to use. In between sets participants could decide how to rest and for how long, with the sensors continuing to collect data.

Participants were asked to wear two smartwatches, one on each wrist, an earbud in their left ear, and two smartphones, one in their left trouser pocket (Huawei P20) and the other in their right trouser pocket (Samsung S7). The Huawei P20 was only used to collect data in six of the workout sessions, and we have therefore decided to exclude this device from our data analysis here, but we still include this device in the larger dataset. The smartwatches and earbud are fixed in orientation when worn, and the smartphones were ensured to be placed in the pocket with the camera facing downwards and outwards. Inertial sensor data was collected from all five devices using the inbuilt triaxial accelerometer and gyroscope, and also the magnetometer in the smartphones. Heart rate data was also collected from the smartwatches using the optical heart rate sensor. Figure 1 displays an

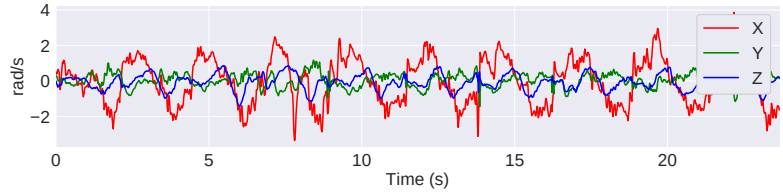
[2] https://mmfit.github.io

Table 1. An overview of the devices used in the data collection, and the collected modalities.

Device	Modality	Frequency (Hz)	Resolution
Orbbec Astra Pro	RGB	30	1080x720
	Depth	30	640x480
Mobvoi TicWatch Pro	Accelerometer	100	-
	Gyroscope	100	-
	Heart beats per minute	1	-
eSense	Accelerometer	90	-
	Gyroscope	90	-
Samsung S7	Accelerometer	210	-
	Gyroscope	210	-
	Magnetometer	100	-
Huawei P20	Accelerometer	500	-
	Gyroscope	500	-
	Magnetometer	65	-



(a) Accelerometer.



(b) Gyroscope.

Fig. 1. Sensor readings from a left-worn smartwatch triaxial accelerometer and gyroscope recorded during a set of squats.

example accelerometer and gyroscope sensor segment. Custom applications stream and store the sensor data from each device. All the devices used in the data collection, their sensing modalities and sampling frequencies are presented in Table 1.

Colour images and depth maps of the workouts were recorded at 30 frames per second from two viewpoints using the Orbbec Astra Pro RGB-D camera. The Astra Pro depth sensor uses structured light to capture depth information, and has a range of 0.6-8m. We place the camera such that the participant is within a 2-6m range from the camera. The depth maps capture the planar distance to the camera of the scene. For each workout session the camera was set up in approximately the same position. The colour and depth data was recorded at the maximum resolution, 1280×720 for colour, and 640×480 for depth. Example RGB frame and depth map outputs are given in Figure 2.

We provide an additional vision-based modality, in the form of 3D pose estimates. We extract 3D pose estimates from single-view RGB frames using the highly-accurate 2D pose estimation system OpenPose [7], and the 2D

(a) Camera 1 RGB image.

(b) Camera 1 depth map.

Fig. 2. Example colour and depth camera output.

Fig. 3. 2D and 3D pose estimates obtained using OpenPose [7] and Martinez et al. [35], respectively.

to 3D regression model proposed by Martinez et al. [35]. OpenPose is an open source multi-person 2D pose estimation system that uses a CNN-based bottom-up approach to find body parts in RGB images. To lift the 2D pose estimates to 3D, Martinez et al. propose a simple but effective deep fully-connected network, which at the time of publication in 2017, outperformed the state-of-the-art 3D pose estimation on the Human3.6M dataset by 30% [35]. Examples of 2D and 3D pose estimates on two RGB frames from the MM-Fit dataset are shown in Figure 3. The resulting pose estimates are highly accurate, robust to fast movements, and can handle simple cases of occlusion, as demonstrated in Figure 3.

In total, 21 workout sessions were recorded, totalling 809 minutes of data. The workout lengths range from 27 to 67 minutes, with an average duration of 39 minutes. A total of 616 sets, and 6160 repetitions were collected, and exercises constitute 26%, or 207 minutes of the data, with the remaining portion corresponding to non-exercise activities. Ten subjects participated in the data collection; two participants carried out six workout sessions each, one participated in two sessions, and the remaining participants carried out one workout session each.

3.2 Data Annotation

To use the MM-Fit dataset for multimodal learning, the data from each of the six devices used in the data collection were time-synchronised. This was done by asking each participant to perform an abrupt synchronisation jump from a stand-still position at the beginning of each workout. The synchronisation jump was used to determine

the offset between each device system clock and a chosen reference clock. With the offset of system clocks to the reference clock, samples from each device could be time aligned.

Each workout has been manually annotated with the video frame boundaries of where each exercise set begins and ends, and the number of repetitions in each set. The workouts were annotated by the same annotator by inspecting the videos of the workout sessions. Participants were instructed to perform 10 repetitions in each set, however, in limited cases, due to miscounting, fewer or more than 10 repetitions were performed. There is some ambiguity involved when annotating the start and end frames of exercises, and the number of repetitions. When determining when a participant starts an exercise, is it when they are in the exercise start position, or is it only once they start the exercise motion? Is the end of a set of jumping jacks when the feet come together for the last repetition, or is it when the arms stop swinging? Does sitting up at the end of a set of sit-ups count as a repetition? These ambiguous cases do not arise very frequently, but to minimise the impact on the data quality, we ensured to handle these cases in a consistent manner by the same human annotator. Given the discussed ambiguities, and the difficulties in identifying exactly when a workout begins and ends, we estimate that the precision of the majority of our start and end annotations are within 2-3 frames (60-100ms) of the ground-truth, which is more than sufficient for the task of activity recognition and repetition counting.

4 METHODOLOGY

4.1 Activity Segmentation and Exercise Recognition

In this section we outline our proposed multimodal approach for activity segmentation and exercise recognition. Our approach can be split into three main training stages; learning modality-specific representations using unimodal autoencoders, learning a shared cross-modal representation using a multimodal autoencoder, and finally, training a classifier for segmentation and exercise recognition using the shared cross-modal representation.

4.1.1 Unimodal Autoencoders. Motivated by the demonstrated strong performance of modality-specific architectures in previous multimodal deep learning works [38, 41], we employ the same late fusing strategy in this work. We first train separate autoencoder networks for each device and modality to extract modality-specific representations, and to enable us to pretrain our network on external datasets. This approach is particularly useful in settings where labelled data is limited, as it is the case with sensor data from wearable devices.

We use a stacked convolutional autoencoder with a bottleneck to force the network to learn a compact modality-specific representation. CNNs are efficient at learning scale-invariant features and detecting temporal patterns, which make them well-suited for our task. The autoencoders are constructed in a symmetric fashion, with the encoder consisting of convolutional and max-pooling layers, and the decoder consisting of deconvolutional and max-unpooling layers. The rectified linear unit (ReLU) activation function [16] is applied after every convolutional layer, and after the first two deconvolutional layers. The network configuration for the accelerometer and gyroscope modalities, and the skeleton modality are given in Table 2 and 3, respectively.

4.1.2 Inertial Sensor Autoencoders. This section outlines how unimodal representations are learned from the accelerometer and gyroscope data collected from the smartwatches, smartphones and earbud.

We treat the triaxial accelerometer and gyroscope data from the smartwatches and earbud as 1D images with three channels, corresponding to the X, Y, and Z axis. For the smartphone data we use the magnitude of the accelerometer and gyroscope readings, $\sqrt{X^2 + Y^2 + Z^2}$, instead of the raw triaxial signal, and hence the smartphone data only has one channel. Accelerometer and gyroscope readings are dependant on the sensor's orientation. The smartwatches and the earbud are worn such that the orientation of the devices are relatively fixed on the body across all workouts, however, the orientation of the smartphones can vary significantly throughout and across workouts. This transformation is performed to make our smartphone models invariant to the orientation of the smartphone. The magnitude of the accelerometer and gyroscope corresponds to the speed of acceleration, and

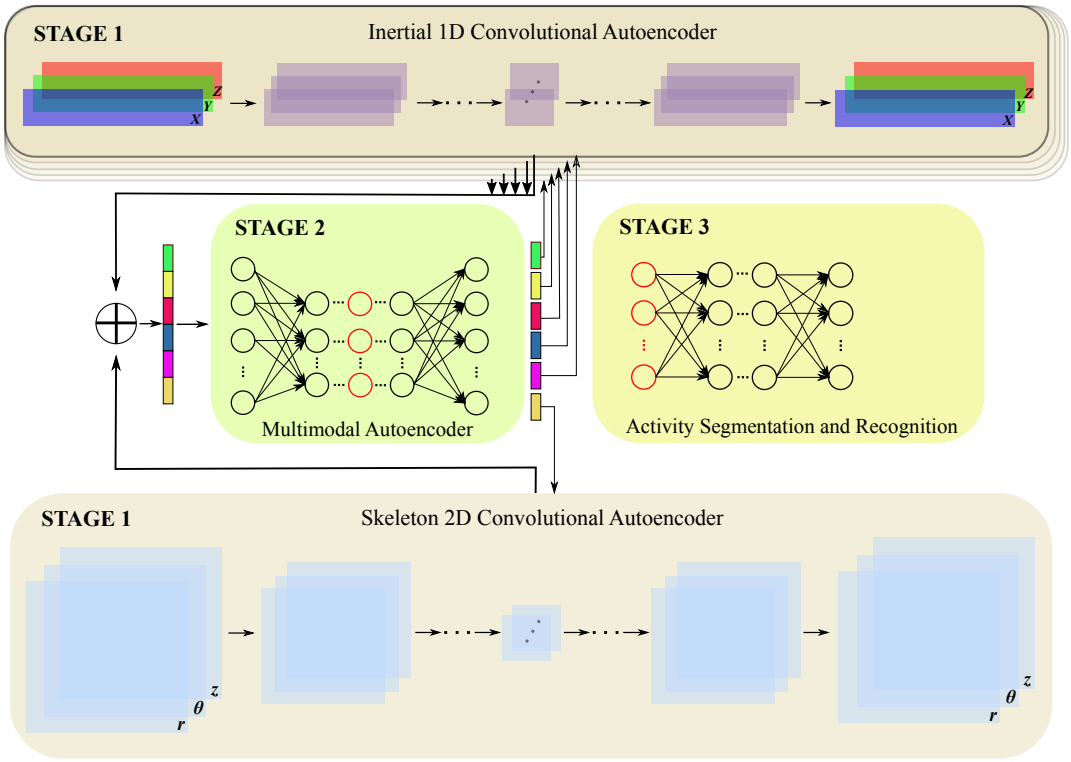


Fig. 4. An overview of the proposed activity segmentation and exercise recognition approach. Stage 1: Learn modality-specific representations using a separate autoencoder for each device and modality. The layers of the inertial 1D convolutional autoencoder block are used to illustrate that separate autoencoders are used for each device and modality. Stage 2: Flatten and concatenate the modality-specific representations outputted by the encoder component of each unimodal autoencoder. Learn a shared cross-modal representation using a fully-connected multimodal autoencoder that attempts to reconstruct the original inputs from the shared representation. The output vector of the multimodal autoencoder is split along the concatenation indices, and fed to the decoder component of the corresponding unimodal autoencoder, to reconstruct the original input, and backpropagate the reconstruction loss. Stage 3: A fully-connected classifier is attached to the learnt shared cross-modal representation. The entire network is trained for the task of activity segmentation and exercise recognition, with the pretrained unimodal and multimodal autoencoder weights being fine-tuned.

speed of rotation, irrespective of the direction. This transformation results in a loss of directional information, but we gain invariance to the orientation of the sensing device. The resulting 1D images for each modality and device are convolved and max-pooled with 1D filters along the temporal dimension. The accelerometer and gyroscope data are processed in separate networks to extract modality-specific representations. We do a network architecture search by using a cross-validation framework to evaluate different network configurations, including the number of convolutional and deconvolutional layers, the kernel size, the kernel stride, and using depth (grouped) or regular convolutions. The final network configurations for all three devices and modalities are given in Table 2.

Table 2. Stacked convolutional autoencoder network architecture for smartwatch and earbud data. The configuration for the smartphone autoencoders only differ in that the number of channels are divided by a factor of three, since the smartphone data only has one input channel. The first two convolutional layers in the accelerometer autoencoder are grouped (G) convolutions, with the number of groups equal to three. The ReLU activation function is applied after every convolutional layer, and after the first two deconvolutional layers. All the layers use a kernel stride of two.

Layer	**Kernel dims (HxW)**		**Output dims (C@HxW)**
	Acc	**Gyr**	
input	-	-	3@1x250
conv1	1x11 (G)	1x3	9@1x125
conv2	1x11 (G)	1x3	15@1x63
conv3	1x11	1x3	24@1x32
maxpool	1x2	1x2	24@1x16
unpool	1x2	1x2	24@1x32
deconv1	1x11	1x3	15@1x63
deconv2	1x11	1x3	9@1x125
deconv3	1x11	1x3	3@1x250

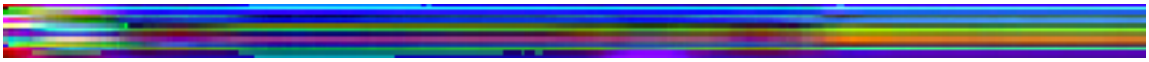
Table 3. Stacked convolutional autoencoder network architecture for skeleton data. The first two convolutional layers in the accelerometer autoencoder are grouped (G) convolutions, with the number of groups equal to three. The ReLU activation function is applied after every convolutional layer, and after the first two deconvolutional layers. All the layers use a kernel stride of two in the height dimension, and one in the width dimension.

Layer	**Kernel dims (HxW)**	**Output dims (C@HxW)**
input	-	3@150x16
conv1	11x11 (G)	9@75x16
conv2	11x11 (G)	15@38x16
conv3	11x11	24@19x16
maxpool	2x2	24@9x8
unpool	2x2	24@19x16
deconv1	11x11	15@38x16
deconv2	11x11	9@75x16
deconv3	11x11	3@150x16

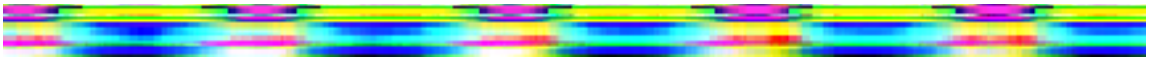
4.1.3 Skeleton Autoencoder. To incorporate visual information into our model, we use 3D pose information as one of our input modalities. Our 3D skeleton joint model consists of 17 joints, for which we have the Cartesian coordinate position for each video frame. The coordinate system used to describe the pose estimate is relative to the person, with the origin coordinate being at the centre hip joint. We use a skeleton representation proposed by Ke et el. [25], which involves transforming the 3D Cartesian coordinates, (x, y, z), to cylindrical coordinates, (r, θ, z), as follows:

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ \theta &= \tan^{-1}\left(\frac{x}{y}\right) \\ z &= z \end{aligned} \tag{1}$$

Cylindrical coordinates is demonstrated to improve performance over Cartesian coordinates [25]. This improvement is attributed to the fact that human motions use pivotal movements, and are therefore best described by cylindrical coordinates. There is however a limitation with using cylindrical coordinates, that is not discussed in the original paper [25], namely the angle discontinuity at -180° and 180° of the azimuth component of the cylindrical coordinate. This causes large jumps in the azimuth signal for potentially very small pivotal movements which can complicate learning. After the transformation, a reference joint, R, is chosen, and each joint position is represented in terms of its relative position to the reference joint. We chose the centre hip joint as the reference joint since it is a stable and stationary joint, which results in the final skeleton representation being less prone to noise. Finally the relative cylindrical joint coordinates are aligned into a 3D image. The channels in the image correspond to the radial, azimuth, and vertical components, the width corresponds to the spatial dimension, and the height to the temporal dimension. This results in a $N \times 16 \times 3$ image, where N is the sequence length, and 16



(a) Non-exercise segment.



(b) Squats segment.

Fig. 5. Visualisation of two 10 second segments of the skeleton representation during a non exercise activity, and a set of squats. The segment has been standardised, and then scaled channel-wise between 0 and 255. The radial coordinate, r, corresponds to the red channel, the azimuth coordinate, θ, to the blue channel, and the vertical coordinate, z, to the green channel.

is the number of joints, discarding the reference centre hip joint. A visualisation of the skeleton representation is given in Figure 5, where the repetitive nature of a set of squat repetitions is captured in the form of a repeating visual pattern.

The skeleton representation enables us to use a stacked convolutional autoencoder model to learn modality-specific features that capture joint interactions across time. The skeleton autoencoder we propose is very similar to the autoencoders we use for the inertial sensor streams by using 2D convolution and pooling operations instead of 1D operations. The model configuration is selected by cross-validating a number of different configurations in a hyperparameter grid search. Table 3 contains the final network configuration for the skeleton autoencoder.

Our approach leverages the large number of 3D pose estimates in the MM-Fit dataset to train our model just on skeleton representations. We hypothesise that training our network exclusively on skeleton data will improve our model as the skeleton representation images differ significantly from natural images, even the low-level features. This is illustrated by the example skeleton representation shown in Figure 5.

4.2 Multimodal Autoencoder

Once the modality-specific representations have been learnt for each modality and device, the features from the embedding layer are flattened and concatenated, before being fed into the next stage; the multimodal autoencoder. The aim of this stage is to learn cross-modal representations that are discriminative for the exercise recognition task. This is again done through the use of an autoencoder network, but this time using a stacked fully-connected autoencoder. By using fully connected layers, the network is easily able to intermix features from all modalities. As can be seen in Figure 4, the architecture structure is symmetric, with the decoder component of the multimodal autoencoder reconstructing a vector of the same size as the concatenated input vector. The output vector is then split at the concatenation indices, such that each segment can be reshaped into the same size as its corresponding modality-specific embedding layer. The reshaped vector segment is then passed on to the decoder module of the corresponding unimodal autoencoder, which attempts to reconstruct the original input for that modality. The best network configuration is outlined in Table 4.

4.3 Multimodal Classification

The final stage of our approach is the classification stage, in which the learned cross-modal representations are fed into a classifier to determine the predicted label for each input segment. Unlike [36], we treat activity segmentation and exercise recognition as one task, instead of two separate tasks, which simplifies our exercise logging pipeline. This is done by adding a non-exercise class to the set of exercise classes. The classifier consists of three fully-connected layers which are attached to the embedding layer of the multimodal autoencoder. The first two fully connected layers consist of 100 hidden units, and use ReLU as the activation function. The entire

Table 4. The network configuration for the stacked fully-connected multimodal autoencoder. The inputs are all the modalities from all the devices, where $12200 = 6 \times (3@1 \times 250) + 2 \times (1@1 \times 250) + (3@150 \times 16)$ is the total size of the inputs across all devices and modalities, and $4288 = 6 \times (24@1 \times 16) + 2 \times (8@1 \times 16) + (24@9 \times 8)$, is the size of the concatenated embedding features from each modality and device.

Layer	**Output Units**
input	12200
mod_specific_encoders	4288
flatten & concat. embeddings	4288
enc_fc1	1000
enc_fc2	1000
enc_fc3	1000
dec_fc1	1000
dec_fc2	1000
dec_fc3	4288
split	4288
mod_specific_decoders	12200

network is trained end-to-end, fine-tuning the autoencoder weights to specialise the learnt features for the task of exercise recognition. The cross-entropy loss is backpropagated through the network to update the whole model parameters.

The signal processing method proposed for RecoFit [36] is inferior to our DL based exercise recognition because it relies on one shot observations and does not learn from more data available from multiple subjects. Secondly, the RecoFit solution is designed to operate on only one modality, whereas our multimodal DL approach takes multiple modalities from multiple devices as input.

4.3.1 Multimodal Zeroing-Out Training Strategy. To learn more robust representations for segmentation and exercise recognition when data from only a single device is available at test time, we investigate whether multimodal training can be used to strengthen the unimodal representations. We do this during training by gradually zeroing out the input modalities that are only available at training time, but still requiring the multimodal autoencoder to reconstruct all the original inputs. By forcing the network to reconstruct modalities for which it only sees the input for occasionally, the network will learn features that encode information about the relationships between different modalities. Thus this supplements and strengthens the unimodal features of the device. In particular, we believe that the less discriminative modalities (from the earbud and smartphone), can benefit from the data perspectives provided by the other stronger modalities during training.

4.4 Repetition Counting

To count the number of repetitions we implemented a signal processing based approach originally designed for repetition counting on accelerometer and gyroscope data collected from the upper-arm [36].

This approach relies on identifying periodic peaks and auto-correlations in the data, which are strong indicators of a repetition. It is assumed that the signal is already segmented. This is preceded by the activity segmentation and exercise recognition stage.

The multidimensional data (3 axes for accelerometer and gyroscope data, or 16 relative joint positions in cylindrical coordinates for skeleton data) is first standardised and then smoothed using a third-degree polynomial Savitzky–Golay filter [45], along each feature dimension. The processed data is then projected onto its first principal component direction to obtain a 1D signal.

The next stage is to identify peaks, or local maxima, in the processed 1D signal. This is done by simply comparing neighbouring values in the signal. We sort the list of candidate peaks based on their amplitude, and in

descending order keep the peaks that have no other higher peak within a minimum allowed threshold distance. This minimum threshold is chosen independently for each exercise based on an estimate of the minimum amount of time required to execute one repetition of the exercise.

Next we exclude unlikely peaks using auto-correlation of the signal. The auto-correlation is computed for a window centred at each peak for lags between the minimum and maximum expected duration of a repetition. The largest autocorrelation value within this range of lags is chosen to be the period, P, at that candidate peak. Any peaks with smaller amplitudes, and within a distance of $0.75 * P$ from the current peak, are removed.

The final filtering is to remove all peaks that have an amplitude lower than half of the 40th percentile of the remaining peak amplitudes. The number of remaining peaks is our predicted repetition count.

5 EVALUATION

This section outlines and discusses the evaluation of our proposed exercise logging system on the MM-Fit dataset. The system is evaluated for exercise recognition and repetition counting. We also present and analyse the multimodal zeroing-out training strategy.

5.1 Activity Segmentation and Exercise Recognition

We evaluate our proposed multimodal deep learning approach on the MM-Fit dataset, and compare its performance to unimodal models, and single-device models. Unimodal models are models that only take input from one modality and device, for example, accelerometer data from the left smartwatch. Single-device models are models that take as input all the modalities generated by a single device, for example, accelerometer and gyroscope data from the left smartwatch. We evaluate the effect of pretrained unimodal autoencodersversus training from scratch. Our analysis considers two cases, the first is a random split of the entire dataset between training and test sets, and the second is a leave one participant out split, where the test set contains only samples from participants who are never included in the training set.

5.1.1 Pretraining Unimodal Autoencoders. This section outlines the experimental setup used to pretrain the unimodal autoencoders. We trained a total of seven autoencoders, corresponding to the following devices and modalities; smartwatch accelerometer and gyroscope, smartphone accelerometer and gyroscope, earbud accelerometer and gyroscope, and skeleton data. The following three datasets were used for pretraining, RecoFit [36], the Heterogeneity Human Activity Recognition (HHAR) dataset [47], and Human3.6M [8, 24].

The Microsoft RecoFit dataset [36] was used to pretrain the smartwatch accelerometer and gyroscope autoencoders. The RecoFit dataset consists of accelerometer and gyroscope data collected at 50Hz from an arm-worn sensor of individuals working out in a gym. To pretrain the smartphone and earbud accelerometer and gyroscope autoencoders we use the smartphone data from the HHAR dataset. The HHAR dataset consists of triaxial accelerometer and gyroscope data collected from 12 smart device models at different frequencies. Finally, the skeleton autoencoder was pretrained on ground truth motion capture data from the Human3.6M dataset. The Human3.6M dataset contains 3D pose information collected from a highly accurate motion capture system at 50Hz.

As a pre-processing step we first downsample all inertial sensor readings to 50Hz, and the 3D pose data to 30Hz. Then the magnitude of the smartphone accelerometer and gyroscope data was computed, and the 3D pose data was transformed to the skeleton representation form outlined in section 4.1.3. Each dataset was then randomly split into a train, test, and validation set, using a 70-15-15 split.

To select the network configuration for our stacked convolutional autoencoder we evaluate different network configurations. The hyperparameter search was carried out separately for each modality, however, we only used the RecoFit dataset to choose the network configuration for the accelerometer and gyroscope autoencoders. The assumption is that a similar network architecture would work well across all three devices for the same type of

modality. The following network configurations were explored; the number of convolutional and deconvolutional layers, the kernel size, the kernel stride, and using depth (grouped) or regular convolutions. The configurations with the lowest reconstruction loss on the external dataset's test set were evaluated on the MM-Fit dataset for exercise recognition, with the configuration. The highest accuracy one is selected as our final configuration. We use the results from the autoencoder model hyperparameter search to guide the final model selection for each modality, with the intuition that models with a low reconstruction loss are able to preserve salient features of activity data in their embedding layer. The final network configurations are given in Table 2 and 3.

We use 5 second window segments as input to the autoencoders. Instances are randomly sampled from the training set, using a batch size of 128. Each model was trained for 100 epochs, using an Adam optimiser [27] with a learning rate of 0.001, and the following exponential decay rates for the moment estimates, β_1 equal to 0.9 and β_2 equal to 0.999. The final weights of the seven autoencoder models are used for exercise recognition on the MM-Fit dataset.

5.1.2 Unimodal, Single-Device and Multimodal Models. We detail the approach taken to train the exercise recognition models that leverage the learnt modality-specific representations (pretrained unimodal encoders).

We split the MM-Fit dataset into a train, validation, and two test sets, first containing other samples from participants seen in the training set, and second with previously unseen participants. The dataset is divided by assigning each workout session to one of the splits. For reproducibility the workout IDs for each split: train (1, 2, 3, 4, 6, 7, 8, 16, 17, 18), validation (14, 15, 19), seen subject test set (9, 10, 11), cross-subject test set (0, 5, 12, 13, 20). We use only one RGB-D camera from the MM-Fit dataset.

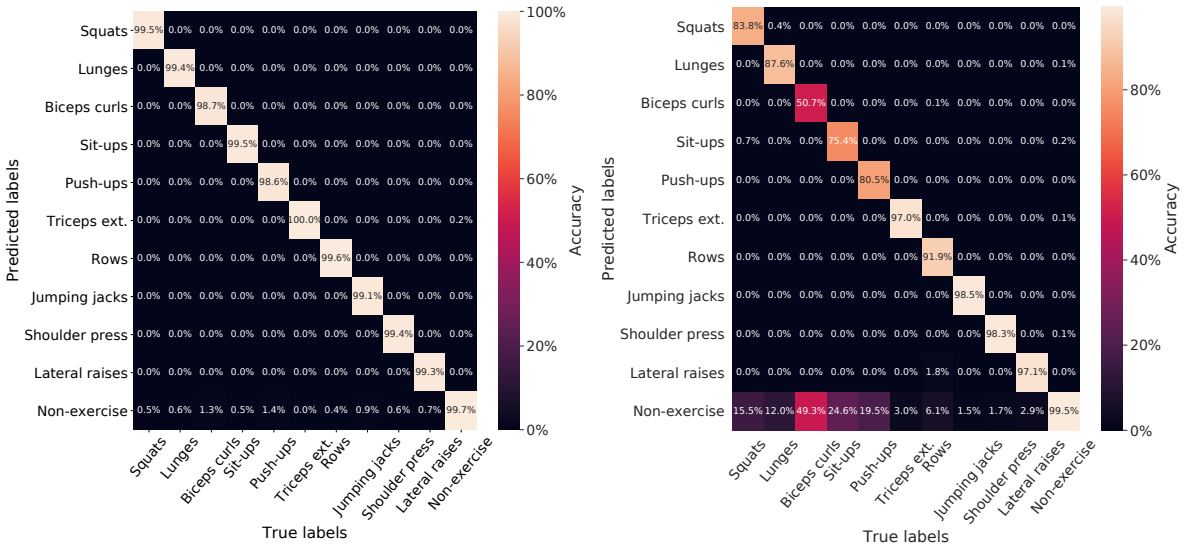
All the models take a 5 second window segment as input, which corresponds to 250 sensor samples at 50Hz, or 150 skeleton samples at 30Hz. There is often a trade-off between accuracy and recognition speed [4]. Five second windows are large enough to capture the repetitive nature of repetitions, whilst also ensuring that the recognition speed (2.5 seconds in overlapping windows) is acceptable for an exercise logging system. During training, instances were randomly sampled from all workouts in the training set, using a batch size of 128. At test time we generated batches using sequential strided sampling, with a stride of 0.2 seconds. This is how the system would be used in a real-world application. The above experimental setup resulted in 702703 training instances, 25969 validation instances, 40544 test instances, and 58328 cross-subject test set instances. Window segments are labelled according to the majority class in the segment, which is common-practice for activity recognition.

We determine the number of fully-connected layers, hidden units, and amount of dropout to use for the unimodal, single-device, and multimodal models through hyperparameter search. The configuration with the highest validation accuracy is chosen for each model. The best configuration is 3 layers with 100 hidden units, and no dropout for all three model types. Another hyperparameter search finds the configuration of the multimodal autoencoder. The final configuration consists of three encoder and three decoder fully-connected layers, with 1000 hidden units, and 1000 units in the embedding layer. The unimodal models are evaluated for exercise recognition by flattening the embedding representation from the modality-specific autoencoder and attaching the three-layered fully-connected classifier. The same approach is taken for single-device models, with the only difference being that the embedding layer from two modalities are flattened and concatenated before attaching the fully-connected classifier. Finally, the multimodal model is evaluated for exercise recognition by attaching the fully-connected classifier to the embedding layer of the multimodal autoencoder. In all three model types, the autoencoder weights are fine-tuned using a lower learning rate than for the fully-connected layers.

All experiments are trained using an Adam optimiser [27] with a learning rate of 0.001, and the following exponential decay rates for the moment estimates, β_1 equal to 0.9 and β_2 equal to 0.999. For fine-tuning the pretrained weights, an order of magnitude lower learning rate of 0.0001 is used. A learning rate scheduler is used to decrease the learning rate by an order of magnitude if the validation metric does not improve for 10 epochs. The models is trained until convergence.

Table 5. Multimodal, unimodal and single-device exercise recognition accuracies on the MM-Fit test sets. We evaluate whether pretraining (PT) boosts performance, and how well each model generalises to unseen test subjects (UTS).

Device	**Modality**	**Accuracy** (w/o PT)	**Accuracy** (w PT)	**Accuracy** (w/o PT, UTS)	**Accuracy** (w PT, UTS)
All		-	**99.60%**	-	**96.37%**
Camera	Skeleton	98.02%	98.78%	**94.59%**	96.01%
Watch left	Acc	98.39%	98.55%	90.74%	90.79%
	Gyr	97.31%	97.54%	89.95%	90.30%
	Acc & Gyr	**98.87%**	98.82%	91.89%	91.74%
Watch right	Acc	98.22%	98.51%	92.72%	91.50%
	Gyr	97.02%	97.80%	89.28%	89.34%
	Acc & Gyr	98.80%	98.76%	93.81%	91.95%
Phone right	Acc	91.34%	89.63%	82.72%	82.22%
	Gyr	86.25%	86.33%	78.26%	79.01%
	Acc & Gyr	92.50%	92.09%	84.75%	83.11%
Earbud	Acc	88.61%	88.73%	79.76%	80.06%
	Gyr	86.25%	88.27%	81.12%	80.55%
	Acc & Gyr	90.55%	91.43%	81.82%	79.90%



(a) Seen subjects.

(b) Unseen subjects.

Fig. 6. Confusion matrix for the multimodal model on the MM-Fit test set, for seen and unseen subjects.

The exercise recognition accuracies of our models over the MM-Fit dataset are presented in Table 5, along with confusion matrices for the multimodal model in Figure 6. The highest accuracies on the two test sets, seen and unseen subjects, are obtained by the multimodal model with 99.60% and 96.37%, respectively. The skeleton model is the best performing unimodal model, with 98.78% accuracy on the seen test set, and 96.01% on the unseen test set. The strong performance of the skeleton model is expected, since the visual information encoded in the

skeleton sequence data is naturally very discriminative for the exercise recognition task. Incorporating sensor information into the skeleton model through multimodal learning has marginal benefit.

The next-best performing models are the left and right multimodal smartwatch accelerometer and gyroscope models, with test accuracies of 98.87% and 98.80%, respectively. The strong performance of the smartwatch models can be explained by the fact that the smartwatches are involved in the movements of all ten exercises studied in the MM-Fit dataset. The earbud and right smartphone models perform substantially worse than the other devices' models, which is expected since the earbud and smartphone are positioned such that they are not heavily involved in many of the exercises. The smartphone and earbud show poor classification performance on exercises with limited head and core movement, such as biceps curls, triceps extensions, shoulder press and lateral raises, and better performance on, for example, squats and jumping jacks, as observed from the confusion matrix.

Among the unimodal models we observe that the unimodal accelerometer models consistently outperform the corresponding unimodal gyroscope model, suggesting that acceleration is more discriminative than angular velocity for exercise recognition. Fusing accelerometer and gyroscope data, as is done in the single-device models, further improves exercise recognition accuracy for all devices, demonstrating the benefit of leveraging multiple data perspectives through multimodal learning.

To evaluate the generalisation ability of our models we tested each model on unseen test subjects (subjects that were not present in the training data). The results show that our models perform 5-10% worse on unseen subjects, but still generalising well.

The confusion matrices reveal that the most common misclassification case is exercise segments being predicted as non-exercise segments, in particular on the unseen subject test set. Many of these misclassifications occur at the ambiguous boundary regions at the start and end of exercise sets, where the window segment contains both exercise and non-exercise samples. This problem could be alleviated through temporal smoothing of predictions, for example, only predicting a new activity class if three consecutive predictions are in agreement.

Surprisingly, the effect of model pretraining was not significant for improving model performance on our exercise recognition task, unlike what is currently understood in the transfer learning community [11]. This is because our lightweight models (with few parameters) can train from scratch on our sufficiently large dataset without overfitting. However, in settings where only a small dataset is available for the target task, but where there are many similar unlabelled instances available, pretraining could still help to improve performance.

5.1.3 Multimodal Zeroing-Out Training Strategy. To learn more robust representations for segmentation and exercise recognition when data from only a single device is available at test time, we investigate whether multimodal training can be used to strengthen the unimodal representations. We refer to the device that is used at test time as the target device.

We initialise the multimodal autoencoder with the pretrained weights of a multimodal autoencoder trained on all available modalities. For the first two epochs, we randomly select a device (excluding the target device) to zero out for each batch, and for every following two epochs, we increase the number of devices to zero out. This process continues until only the target device is not being zeroed-out, at which point we let the network train for another five epochs. The model is evaluated on the validation set after every epoch, by zeroing out all the modalities except for the modalities belonging to the target device, and recording the reconstruction loss in reconstructing all the original inputs. The model with the lowest validation reconstruction loss is selected to be evaluated for exercise recognition. To evaluate the learnt representation on the MM-Fit dataset for exercise recognition, three fully connected layers with 100 hidden units are attached to the embedding layer of the multimodal autoencoder. The model is then trained for exercise recognition, zeroing out all modalities, except for the modalities belonging to the target device. The model with the highest validation accuracy is selected and

Table 6. Single-device test accuracies for pretrained models trained using zeroing out strategy, for both seen and previously unseen test subjects (UTS). Results should be compared with the corresponding single-device pretrained model's accuracies in Table 5.

Device	**Accuracy**	**Accuracy** (UTS)
Camera left	97.34%	92.79%
Smartwatch left	**99.20%**	**93.59%**
Smartwatch right	98.80%	92.43%
Smartphone right	92.50%	84.20%
Earbud	91.41%	78.68%

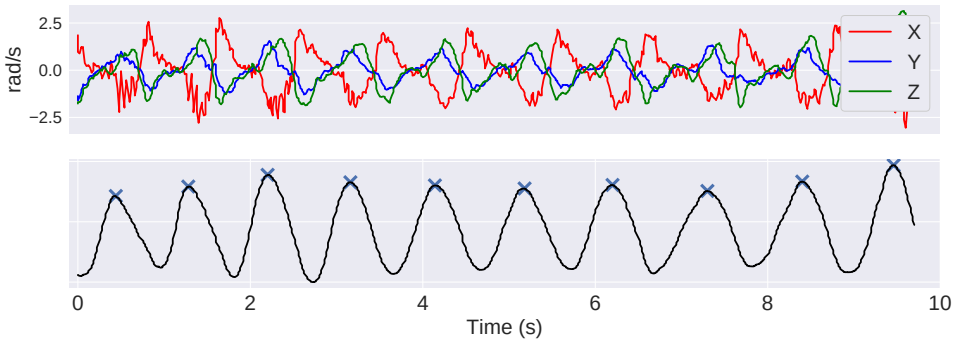


Fig. 7. Repetition counting for a set of ten squat repetitions using gyroscope readings from a smartwatch worn on the left wrist. The original gyroscope data is shown in the top plot, and the processed 1D signal is shown in the bottom plot, with the predicted repetitions marked out.

evaluated on the test sets. The experimental setup is identical to the one outlined in section 5.1.2 in terms of learning rate, optimiser, and batch size.

We trained five models, each with a different target device; left camera, left smartwatch, right smartwatch, right smartphone, and earbud. The test set accuracies for each of the five models is presented in Table 6. The multimodal zeroing-out training strategy results in better accuracy for the two smartwatches on the seen test set, and for the right smartwatch on the unseen test set compared to the corresponding single-device models. However, for the remaining devices, the test set accuracy is lower when using the zeroing-out training strategy. These results indicate that the zeroing-out training strategy is not very effective in learning more robust representations to leverage external data perspectives.

5.2 Repetition Counting

We implement and evaluate the baseline approach for repetition counting on the MM-Fit dataset. Figure 7 presents an example output of the repetition counting of peaks. Since the repetition counting does not require any training, we can evaluate this on all the workout sessions. This brings the total number of exercise sets to evaluate on to 616.

The Mean Absolute repetition counting Error (MAE) per set obtained by the peak counting and auto-correlation method, for each modality and device, are given in Table 7. Table 8 presents the percentage of predictions within different ranges of the ground truth. The left smartwatch gyroscope modality obtains the lowest MAE, 0.34, closely followed by the right smartwatch gyroscope, 0.35, and the left and right smartwatch accelerometers, 0.41

Table 7. The mean absolute repetition counting error per set for each modality and device across all exercises; squats (Sq), push-ups (Pu), shoulder press (Sp), lunges (Lu), dumbbell rows (Ro), sit-ups (Su), triceps extensions (Te), biceps curls (Bc), lateral raises (Lr), and jumping jacks (Jj).

		Exercises										**Total**
Device	**Mod.**	Sq	Pu	Sp	Lu	Ro	Su	Te	Bc	Lr	Jj	
Cam L.	Skel	**0.05**	0.52	0.63	**0.03**	0.23	**0.17**	0.33	4.41	0.09	**0.26**	0.67
Watch L.	Acc	0.25	0.6	**0.18**	**0.03**	**0.22**	0.31	0.30	1.73	**0.05**	0.42	0.41
	Gyr	0.19	**0.38**	0.65	0.60	0.39	0.26	**0.19**	**0.41**	0.09	**0.26**	**0.34**
Watch R.	Acc	0.30	0.62	0.25	0.10	**0.22**	0.31	0.27	1.78	0.09	0.49	0.44
	Gyr	0.25	0.40	0.50	0.56	0.39	0.26	0.27	0.42	0.12	0.35	0.35
Phone R.	Acc	1.39	0.43	1.78	0.31	1.23	2.43	2.64	3.02	2.07	1.14	1.64
	Gyr	3.31	0.77	1.30	1.16	0.67	2.72	1.80	1.36	1.50	1.46	1.61
Earbud	Acc	0.50	1.00	1.52	0.42	1.80	1.03	0.83	1.98	1.48	2.75	1.31
	Gyr	1.11	1.22	1.18	0.85	1.59	0.75	0.81	1.98	1.36	2.46	1.31

Table 8. The percentage of exercise sets for which the predicted repetition count is exact, within 1, or within 2 of the ground-truth repetition count, for each modality and device. The percentages in brackets for the skeleton modality, are the results obtained when ignoring biceps curls.

Device	**Modality**	**Exact**	**Within 1**	**Within 2**
Camera left	Skeleton	69.81% (76.84%)	89.77% (98.20%)	90.91% (99.46%)
Smartwatch left	Acc	72.73%	94.16%	96.59%
	Gyr	**73.38%**	96.27%	98.21%
Smartwatch right	Acc	73.05%	93.83%	95.62%
	Gyr	71.92%	**96.59%**	**98.70%**
Smartphone right	Acc	41.40%	65.10%	75.81%
	Gyr	30.68%	66.40%	82.95%
Earbud	Acc	39.45%	68.18%	80.52%
	Gyr	37.66%	68.34%	81.33%

and 0.44, respectively. The predictions made using phone and earbud data are significantly worse compared to those made using the watch and skeleton data. This is expected since the phone and earbud do not capture the central movements of many of the exercises due to their position (for example biceps curls, rows, and triceps extensions). Surprisingly, the skeleton modality obtains a MAE which is over two times greater than the best performing modality, the left smartwatch gyroscope. This is surprising since intuitively the skeleton sequence data should be the most informative irrelevant of which body parts are involved in the exercise. Analysing the results further by looking into the performance at the exercise level (Table 7), reveals that the skeleton modality struggles to count biceps curls, but performs well on all the other exercises. The reason for the skeleton modality's poor performance on counting biceps curl repetitions is explained by Figure 8. The processed signal obtained using the implemented approach defines one bicep curl repetition as two alternating curls (one for each arm). This highlights an ambiguous case of how a repetition should be defined; is one curl a repetition, or are two alternating curls one repetition? Ignoring the biceps curl exercise results in a MAE of 0.26 for the skeleton modality, which is on level with the performance of the smartwatch models.

The results we obtain are in line with the results reported by Morris et al. for repetition counting on the RecoFit dataset [36]. They obtain a MAE of 0.26 across 160 sets for 26 different exercises. Their predictions were obtained using data from a right-arm worn accelerometer sensor. They obtain exact predictions in 50% of the sets, within-1 in 97% of the sets, and within-2 in 99% of the sets, which can be compared to our right watch accelerometer, exact

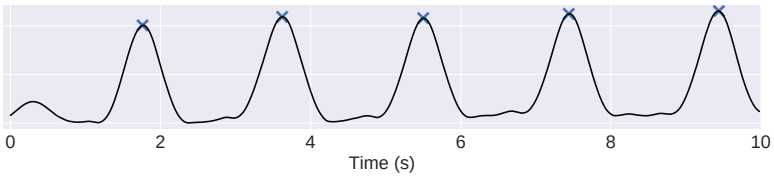


Fig. 8. Processing the skeleton modality as signal for repetition counting on a set of biceps curls, with the predicted repetitions marked out (using the method of [36]). This miss-counts by ignoring the alternating hand in the exercise set.

72%, within-1 97%, and within-2 99%. We find gyroscope data to be more useful for repetition counting than the accelerometer, although by a small margin.

5.3 Discussion

The repetition counting solution is efficient, but it has a few limitations. One limitation is that a minimum and maximum repetition duration threshold needs to be manually chosen for each exercise. The performance is sensitive to how accurate these thresholds are, and also entails that the approach is not invariant to the rate at which repetitions are performed. Another limitation is that the method is heavily reliant on accurate segmentation and classification in order to obtain good results. Finally, the method relies on simple heuristics based signal processing techniques, and does not learn a high-level representation of what constitutes a repetition. This problem is exemplified by its behaviour in counting bicep curl repetitions on smartwatch data. Intuitively one would expect the repetition counting method to estimate five repetitions for a set of ten alternating biceps curls since the smartwatch registers the curls from one of the arms. However, the method often predicts ten or close to ten repetitions in these scenarios. This is because the assumptions supplied to the model about how long a biceps curl repetition should last causes the model to interpret the small peaks in between the actual repetitions as corresponding to repetitions. These peaks could be just noise but the method can not distinguish between actual repetition peaks and random peaks. Adaptive methods to learn variations in repetitions are required for more confident repetition counting.

6 CONCLUSIONS

In this work, we developed the components of an automatic exercise logging systems, comprising of activity segmentation, exercise recognition and repetition counting. These are built on multimodal deep learning methods to extract relevant information from across multiple devices and multiple sensing modalities.

We evaluate our technical solution on our newly collected and annotated MM-Fitdataset – a substantial dataset of 21 full-body workout sessions with 809 minutes across 10 participants, comprised of RGB-D video, inertial sensing, and pose estimate data. Our proposed multimodal architecture achieves 96% accuracy across all modalities, 94% for the smartwatch, 85% for the smartphone and 82% for the earbud device on the MM-Fitdataset. The benefit of our multimodal learning approach is also seen in the performance boost observed across all sensing devices when fusing their accelerometer and gyroscope modalities, and for the support provided by additional perspectives available during training time. Finally, we implement a strong repetition counting baseline. We find that the best performing modality for repetition counting is the smartwatch gyroscope, with a mean absolute prediction error per set of 0.34 on our MM-Fit dataset.

ACKNOWLEDGMENTS

This project has received funding from Vinnova, Sweden's innovation agency, under grant agreement No. 2018-04643. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the funding body.

REFERENCES

[1] Jake K. Aggarwal and Lu Xia. 2014. Human activity recognition from 3D data: A review. *Pattern Recognition Letters* 48 (2014), 70–80.
[2] B Almaslukh. 2017. An Effective Deep Autoencoder Approach for Online Smartphone-Based Human Activity Recognition. *International Journal of Computer Science and Network Security* 17 (04 2017).
[3] Lei Bai, Lina Yao, Xianzhi Wang, Salil S Kanhere, Bin Guo, and Zhiwen Yu. 2020. Adversarial Multi-view Networks for Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–22.
[4] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. 2014. Window Size Impact in Human Activity Recognition. *Sensors* 14, 4 (2014), 6474–6499.
[5] C. G. Bender, J. C. Hoffstot, B. T. Combs, S. Hooshangi, and J. Cappos. 2017. Measuring the fitness of fitness trackers. In *2017 IEEE Sensors Applications Symposium (SAS)*. 1–6.
[6] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* 46 (2014), 33:1–33:33.
[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*.
[8] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. 2011. Latent Structured Models for Human Pose Estimation. In *International Conference on Computer Vision*.
[9] Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. 2012. Sensor-Based Activity Recognition. *Trans. Sys. Man Cyber Part C* 42, 6 (Nov. 2012), 790–808.
[10] K. S. Choi, Y. S. Joo, and S. Kim. 2013. Automatic exercise counter for outdoor exercise equipment. In *2013 IEEE International Conference on Consumer Electronics (ICCE)*. 436–437.
[11] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems* 36, 3 (2013), 537–556.
[12] Yong Du, Yun Fu, and Liang Wang. 2015. Skeleton based action recognition with convolutional neural network. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)* (2015), 579–583.
[13] Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 1110–1118.
[14] Georgios Dimitrios Evangelidis, Gurkirt Singh, and Radu Horaud. 2014. Skeletal Quads: Human Action Recognition Using Joint Quadruples. *2014 22nd International Conference on Pattern Recognition* (2014), 4513–4518.
[15] Hristijan Gjoreski, Jani Bizjak, Martin Gjoreski, and Matjaz Gams. 2016. Comparing Deep and Classical Machine Learning Methods for Human Activity Recognition using Wrist Accelerometer.
[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research)*, Geoffrey Gordon, David Dunson, and Miroslav Dudík (Eds.), Vol. 15. PMLR, Fort Lauderdale, FL, USA, 315–323.
[17] Xiaonan Guo, Jian Liu, Cong Shi, Hongbo Liu, Yingying Chen, and Mooi Choo Chuah. 2018. Device-free personalized fitness assistant using WiFi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–23.
[18] S. Ha, J. Yun, and S. Choi. 2015. Multi-modal Convolutional Neural Networks for Activity Recognition. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. 3017–3022.
[19] Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. In *IJCAI*.
[20] Keng hao Chang, Mike Y. Chen, and John F. Canny. 2007. Tracking Free-Weight Exercises. In *UbiComp*.
[21] Samitha Herath, Mehrtash Tafazzoli Harandi, and Fatih Murat Porikli. 2017. Going Deeper into Action Recognition: A Survey. *Image Vision Comput.* 60 (2017), 4–21.
[22] HM Sajjad Hossain, MD Abdullah Al Haiz Khan, and Nirmalya Roy. 2018. DeActive: scaling activity recognition with active deep learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–23.
[23] Inhwan Hwang, Geonho Cha, and Songhwai Oh. 2017. Multi-modal human action recognition using deep neural networks fusing image and inertial sensor data. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 278–283.

[24] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (jul 2014), 1325–1339.
[25] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Ahmed Sohel, and Farid Boussaïd. 2017. A New Representation of Skeleton Sequences for 3D Action Recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 4570–4579.
[26] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *IMWUT* 2 (2018), 185:1–185:17.
[27] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
[28] Oscar D. Lara and Miguel A. Labrador. 2013. A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials* 15 (2013), 1192–1209.
[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (27 5 2015), 436–444.
[30] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. 2278–2324.
[31] O. Levy and L. Wolf. 2015. Live Repetition Counting. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 3020–3028.
[32] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. 2017. Skeleton-based action recognition with convolutional neural networks. *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (2017), 597–600.
[33] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2019. Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition. In *CVPR*.
[34] Shanhong Liu. 2018. Fitness & Activity Tracker. https://www.statista.com/study/35598/fitness-and-activity-tracker/.
[35] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *ICCV*.
[36] Dan Morris, T. Scott Saponas, Andrew Guillory, and Ilya Kelner. 2014. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In *CHI*.
[37] Bobak Mortazavi, Mohammad Pourhomayoun, Gabriel Alsheikh, Nabil Alshurafa, Sunghoon Ivan Lee, and Majid Sarrafzadeh. 2014. Determining the Single Best Axis for Exercise Repetition Recognition and Counting on SmartWatches. *2014 11th International Conference on Wearable and Implantable Body Sensor Networks* (2014), 33–38.
[38] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. 2011. Multimodal Deep Learning. In *ICML*.
[39] Ronald Poppe. 2010. A survey on vision-based human action recognition. *Image Vision Comput.* 28 (2010), 976–990.
[40] Valentin Radu, Nicholas D. Lane, Sourav Bhattacharya, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. 2016. Towards Multimodal Deep Learning for Activity Recognition on Mobile Devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)*. ACM, New York, NY, USA, 185–188.
[41] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D. Lane, Cecilia Mascolo, Mahesh K. Marina, and Fahim Kawsar. 2018. Multimodal Deep Learning for Activity and Context Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 157 (Jan. 2018), 27 pages.
[42] Iasonas Kokkinos Riza Alp Guler, Natalia Neverova. 2018. DensePose: Dense Human Pose Estimation In The Wild. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[43] T. F. H. Runia, C. G. M. Snoek, and A. W. M. Smeulders. 2018. Real-World Repetition Estimation by Div, Grad and Curl. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9009–9017.
[44] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2018. Synthesizing and reconstructing missing sensory modalities in behavioral context recognition. *Sensors* 18, 9 (2018), 2967.
[45] Abraham. Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639.
[46] Andrea Soro, Gino Brunner, Simon Tanner, and Roger Wattenhofer. 2019. Recognition and Repetition Counting for Complex Physical Exercises with Deep Learning. *Sensors* 19, 3 (2019).
[47] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart Devices Are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*. ACM, New York, NY, USA, 127–140.
[48] Terry Taewoong Um, Vahid Babakeshizadeh, and Dana Kulic. 2016. Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), 2385–2390.
[49] Eduardo Velloso, Andreas Bulling, Hans-Werner Gellersen, Wallace Ugulino, and Hugo Fuks. 2013. Qualitative activity recognition of weight lifting exercises. In *AH*.
[50] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 588–595.

[51] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2019), 3–11.

[52] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2014. Learning Actionlet Ensemble for 3D Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (2014), 914–927.

[53] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *AAAI*.

[54] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *IJCAI*.

[55] Ming Zeng, Le T. Nguyen, Bo Yu, Ole J. Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional Neural Networks for human activity recognition using mobile sensors. *6th International Conference on Mobile Computing, Applications and Services* (2014), 197–205.