remote sensing

MDPI

Article

Context-Aware Human Activity and Smartphone Position-Mining with Motion Sensors

Zhiqiang Gao [1,2], Dawei Liu [1,*], Kaizhu Huang [3], and Yi Huang [2]

1 Department of Computer Science and Software Engineering, Xi'an Jiaotong-liverpool University, Suzhou 215000, China; zhiqiang.gao@xjtlu.edu.cn
2 Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 7ZX, UK; sgzgao5@liverpool.ac.uk (Z.G.); Yi.Huang@liverpool.ac.uk (Y.H.)
3 Department of Electrical and Electronic Engineering, Xi'an Jiaotong-liverpool University, Suzhou 215000, China; kaizhu.huang@xjtlu.edu.cn
* Correspondence: dawei.liu@xjtlu.edu.cn; Tel.: +44-07493697428

Received: 23 September 2019; Accepted: 24 October 2019; Published: 29 October 2019

Abstract: Today's smartphones are equipped with embedded sensors, such as accelerometers and gyroscopes, which have enabled a variety of measurements and recognition tasks. In this paper, we jointly investigate two types of recognition problems in a joint manner, e.g., human activity recognition and smartphone on-body position recognition, in order to enable more robust context-aware applications. So far, these two problems have been studied separately without considering the interactions between each other. In this study, by first applying a novel data preprocessing technique, we propose a joint recognition framework based on the multi-task learning strategy, which can reduce computational demand, better exploit complementary information between the two recognition tasks, and lead to higher recognition performance. We also extend the joint recognition framework so that additional information, such as user identification with biometric motion analysis, can be offered. We evaluate our work systematically and comprehensively on two datasets with real-world settings. Our joint recognition model achieves the promising performance of 0.9174 in terms of $F1$-score for user identification on the benchmark RealWorld Human Activity Recognition (HAR) dataset. On the other hand, in comparison with the conventional approach, the proposed joint model is shown to be able to improve human activity recognition and position recognition by 5.1% and 9.6% respectively.

Keywords: mobile sensing; human activity recognition; smartphone position detection; multi-task learning; machine learning

1. Introduction

Today's smartphones have become more powerful with the advancement of high-performance computation hardware, lower-cost sensing technology, and intelligent operation systems. With near-ubiquitous availability of smartphones equipped with various built-in sensors, smartphone functions are not only limited to the original purposes, such as sound recording, display orientation change, and screen back-light intensity control, but also able to provide an opportunity for the development of innovative and smart sensing applications. These applications make a wide variety of contextual information be extracted considering about the user, the device, and the environment. In this paper, we present a framework that can recognize a user's physical activity as well as the smartphone position on the human body at the same time. It can be used as an enhanced and valuable context-aware service for many applications requiring both human activity and smartphone position information.

User's activity and smartphone's position are two types of contextual information that has inspired a variety of applications. Providing users' physical activity information for the context-aware application is a classical problem known as Human Activity Recognition (HAR). Commonly, human activities that can be recognized include walking, jogging, sitting down, ascending the stairs, and so on. The HAR has enabled many applications such as health monitoring, child and elderly care, and fitness tracking in which user activities are key knowledge. On the other hand, the way a smartphone is carried by a user is another type of contextual information from which many applications can benefit. Typically, a smartphone can be held in hand(s), placed in the pants/jacket pocket, or stored inside a backpack. The difference in smartphone position could affect the way users interact with the smartphone. For example, a user may prefer headphone controls than screen touches when the smartphone is placed inside the pants pocket or backpack; the sound volume of the smartphone should be raised up automatically when a smartphone is placed inside a backpack. Moreover, these two types of contextual information are closely related to each other. It has been shown that the smartphone's position can have a heavy influence on HAR because the same activity measured by a smartphone at different body positions could exhibit distinct physical characteristics [1].

In previous studies, these two contexts were widely discussed in an independent manner, so that the relationship between them has not been fully explored. Some related works studied position recognition methods while the smartphone users are walking [2–4], but without discussing other periodic human activities. The position-ware HAR model is designed to leverage smartphone on-body position information as prior knowledge to improve HAR [1,5–7].

We aim to investigate multiple contexts in a joint manner, which can be considered to be an extension of classical HAR research. The purpose is to recognize the user's activity and the smartphone's position simultaneously and accurately from human movements. Such enhanced context-aware service may contribute to different aspects of sensing applications. Especially in mobile positioning applications, human activity and smartphone position are used as crucial contextual information to trigger different location estimation algorithms [8,9].

We propose a joint recognition framework to explore complementary information among multiple contexts. Intuitively, human activity and position recognition are two closely related tasks. Motivated by this observation, we argue that joint learning of them may leverage useful information and produce an improvement on recognition rates. For this purpose, the Multi-Task Learning (MTL) strategy is an appropriate technique that has the potential to exploit the commonalities and differences across multiple tasks [10,11]. In this study, the designed joint recognition model integrates the MTL strategy with neural network architecture. In contrast to previous works, our approach uses only one global model which outputs results of multiple tasks, which reduces computational demand and has the potential to improve performance. Meanwhile, we propose a data preprocessing approach to deal with the problem caused by smartphone orientation variation. It is a coordinate transformation technique that converts acceleration data measured from the device coordinate system to the earth coordinate system. With the comprehensive experiments, we show the robustness of the proposed framework in solving the joint recognition problem.

The main contributions of this paper are as follows:

- We proposed a joint recognition model to mine multiple contextual information from motion sensor signals of a smartphone. The feasibility and advantage of the proposed approach are demonstrated by its promising results achieved on an existing real-world datasets [6]. More than that, we extended the joint recognition model to an additional task, namely identifying smartphone users with biometric motion analysis [12,13].
- We developed a data preprocessing approach to deal with the problem caused by smartphone orientation variation. To evaluate this method, we collect a dataset that marks the sensor data with orientation labels in addition to human activity and smartphone position.
- To the best of our knowledge, this is the first study that systematically shows the feasibility and performance of mining multiple physical contextual information. We applied our framework

on three machine learning models that have shown promising results in classical HAR tasks, including simple Multilayer Perceptron (MLP) with the statistical feature, and Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) with powerful and automatic data representation ability.

The contents of this article are organized as follows. In Section 2, we first review the previous works about position-aware models, deep learning models and multi-task learning models used in HAR. Section 3 gives a detailed description of our framework, including a proposed preprocessing technique targeting the problem of smartphone orientation variation and neural network architecture for mining multiple information jointly. Section 4 systematically examines two components in our framework by using a series of experiments. Some further discussion is given in Section 5. At last, we highlight our conclusions and future works in Section 6.

2. Related Work

In this section, the related works in smartphone-based HAR will be presented, including position-aware HAR, smartphone on-body position recognition, and deep learning and Multi-Task Learning for HAR.

2.1. Position Recognition and Position-Aware Human Activity Recognition

As the off-the-shelf sensors are embedded in the smartphone, recognizing human activity and smartphone position have been widely investigated separately in previous studies. In general, their efforts are categorized as two branches: one is the smartphone on-body position recognition; the other is position-aware HAR.

The smartphone on-body position recognition has been previously studied to improve context-aware applications [2–4]. For instance, Alanezi [2] proposed a service for smartphone position recognition that can improve performance of fall detection application. However, these studies only explored the smartphone position detection while the users are walking [3,4], without discussing other periodic human activities.

The position-aware HAR uses the smartphone on-body position information as the prior knowledge to improve HAR. Generally speaking, there are two main approaches: (a) two-stage classification, and (b) one-stage classification approach.

The two-stage classification method detects the smartphone positions at first; next, human activities can be recognized by a set of position specified activity classifiers or one generalized activity classifier. The position specified activity classifiers refer to training a classifier for each position, and selecting the activity classifier according to the obtained position information in the first stage [1,5,6]. By contrast, one generalized activity classifier for final HAR takes the features produced by an adjustment technique as input [7]. The adjustment technique narrows the feature difference among smartphone positions. As a result, one generalized classifier can be used to identify the same activity on different smartphone positions.

Even though the two-stage classification approach is feasible in position-aware HAR, such a cascaded model involves multiple models to maintain high performance. Obviously, it requires high computation latency and energy consumption in mobile computing.

In contrast, the one-stage approach is a more efficient position-aware HAR model that recognizes the human activities and smartphone positions in one step with one generalized classifier. Its classification targets are a set of multiple subclasses of predefined activities and positions, where each subclass is defined as a tuple $(activity, position)$. For example, the walking activity with the phone in the trouser pocket or in a backpack is classified separately. Since that, the one-stage classification approach is also suitable for our purpose. However, there are still some limitations in the previous works. Lu et al. [14] proposed a split-and-merge technique to improve HAR. In the split process, the one-stage classification results were used as intermediates and not shown in detail; and only the HAR results were presented in the merge process. Antos et al. [15] achieved 87.1% accuracy in split

process without considering the negative effect on smartphone orientation variation. Besides that, it is worth mentioning that this split-and-merge technique becomes naive when the number of target human activities and smartphone positions increases dramatically.

Unlike the previous related works, we aim to investigate a feasible approach that is robust to our joint recognition problem as well as reduces computational demand. We argue that the relationship between HAR and smartphone position recognition has not been fully explored, assuming that these are two closely related tasks. Furthermore, leveraging their relevant information may provide extra robustness to the joint recognition model.

2.2. Deep Learning for Human Activity Recognition

Deep learning is within the scope of machine learning and normally employs deep models to learn data representation automatically from raw input [16]. One appealing feature of deep learning is high representational capability without relying on a mass of expert knowledge. Due to its powerful function, it can relieve the efforts on designing feature that are usually limited by human domain knowledge [17,18]. Recent years have witnessed the achievement of deep learning in many fields, e.g., computer vision, speech recognition, and natural language processing.

Previous studies have explored lots of existing deep models in the domain of HAR. The main efforts were concentrated on the supervised learning scheme. One of the commonly used deep models is Deep Neural Network (DNN). Some studies [19,20] only employed DNN as a classifier, which takes a hand-engineered feature of the sensor data as input. Hammerla et al. [21] used a 5-hidden-layer DNN to improve recognition rates by performing automatic representation learning and classification. Another powerful and popular architecture is Convolutional Neural Network (CNN). In practice, it not only is it skilled in analyzing image data, but it can also capture the local dependency and scale invariance of sensor data. The main research interests of applying CNN include input adaptation (e.g., data-driven [22,23] and model-driven approach [24,25]), task-specified pooling method and weight-sharing [26]. In addition, the Recurrent Neural Network (RNN) is a classical architecture for modeling sequential data by using the temporal correlations between neurons [27]. The mainline of RNN-based models deals with resource-constrained environments while still achieving good performance [26]. On the other hand, some studies focused on learning data representation by using unsupervised deep models, such as Autoencoder [28,29] and Restricted Boltzmann machine [30]. In this study, we only target the aforementioned supervised learning models.

2.3. Multi-Task Learning in Human Activity Recognition

Multi-Task Learning (MTL) is a learning paradigm in machine learning and its purpose is to take advantage of useful information contributed by multiple related tasks to improve the generalization performance of all the tasks [11]. MTL has shown significant advantage to single-task learning because of its ability to facilitate knowledge sharing between tasks [31], e.g., bioinformatics and health informatics [32,33], web applications [34,35] and remote sensing [36–38].

Even though MTL has been used successfully in many applications, there are limited works that focus on MTL-based HAR. Sun et al. [39,40] proposed a personalized HAR method by applying MTL. In their models, each task corresponds to a specific person. Peng et al. [41] developed a model to recognize complex human activity based on MTL, which leverages the classical HAR as a related task to complex HAR. By using the representation learned from classical HAR as a low-level shared feature, state-of-the-art results of complex HAR are achieved. Our work is inspired by the above studies, which allows us to explore the useful information in HAR and position recognition.

3. Methods

Our joint recognition task is defined as a supervised learning task following classical HAR research. A workflow is shown in Figure 1. Similar to previous research, the motion sensor signals of the smartphone are collected in the beginning. In our case, the used sensors include the accelerometer

and rotation vector sensor that are both easy to obtain in modern smartphones. The acceleration data returned by accelerometer has shown effectiveness in describing body motions in previous HAR research [42]. The rotation vector sensor describes the orientation of the device in the form of a vector that can be used to help the coordinate transformation of acceleration data, which we will address later in this section. After that, each input data instance is a segmentation of continuous sensor signals within a given window size. In this study, we are particularly interested in two components which can have a significant influence on the recognition rate. One of the core components in our framework is a data preprocessing method which aims to remove the negative effect of smartphone orientation variation on recognition. It is a coordinate transformation technique and output acceleration data under the earth coordinate system that is independent of smartphone orientations. Following that, we design a joint recognition model that is qualified for output results of multiple tasks and has the potential to improve recognition performance on each task.

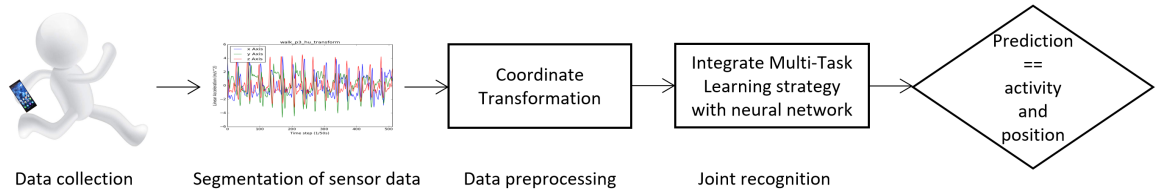

Figure 1. A framework of joint human activity and smartphone position recognition.

3.1. Task Definition

Our model is a typical MTL paradigm for multi-channel time series recognition. In our case, each input data instance, sampled from motion sensors signals within a fixed size window, is assigned with class labels of multiple tasks. Given M tasks $\{T^m\}_{m=1}^M$ that are assumed to be related, we aim to leverage the knowledge contained in all or some related tasks to improve the learning of a model for T^m by using MTL strategy.

Each data sample of our joint recognition task is given by $(\mathbf{X}, \mathbf{y}^1, \ldots, \mathbf{y}^m, \ldots, \mathbf{y}^M)$. The input instance $\mathbf{X} \in \mathbb{R}^{d \times l}$ is multi-channel time series, where l is sampling window size and d is the channel number of time series. $\mathbf{y}^m \in \mathbb{R}^{c^m}$ is a label in the form of one hot vector with class number c^m of m-th task. $\mathbf{X} = (\mathbf{S}, \mathbf{Q})$ is composed of two types of sensor signals, where $\mathbf{S}$ and $\mathbf{Q}$ are collected from accelerometer and rotation vector sensor respectively. Specifically, at time-step t, the accelerometer expresses acceleration force of a device for three axes by a vector $\mathbf{s}_t = [s_t^{(x)}, s_t^{(y)}, s_t^{(z)}]^\top$; the ration vector sensor describes the orientation of device by as a quaternion vector, e.g., $\mathbf{q}_t = [cos(\theta/2), v^{(x)}sin(\theta/2), v^{(y)}sin(\theta/2), v^{(z)}sin(\theta/2)]^\top$, in which the device has rotated through an angle θ around an axis (x, y, or z), the $v^{(x)}$, $v^{(y)}$, and $v^{(z)}$ represent unit rotation of three axes.

A data preprocessing method that is a coordinate transformation function $f_c(\cdot)$, will be applied on input instance $\mathbf{X}$ before learning how to recognize jointly. This function transforms the acceleration sensor signals $\mathbf{S}$ from smartphone standard built-in coordinate system to earth coordinate system by using rotation vector sensor signals $\mathbf{Q}$, which is defined as $\mathbf{E} = f_c(\mathbf{X} = (\mathbf{S}, \mathbf{R}))$. In the downstream recognition model, the transformed results $\mathbf{E}$ will be seen as input data instance.

In the stage of recognition, the purpose is to learn one global model to output classification results of multiple tasks simultaneously. For each task T^m, the goal is to learn a mapping function $f^m : \mathbf{E} \rightarrow \mathbf{y}^m$ that is able to estimate a conditional probability distribution $P(\hat{\mathbf{y}}^m | f_c(\mathbf{X}))$ based on the training set and predict class labels of test instances by $\bar{\mathbf{y}}^m = \arg\max_{\hat{\mathbf{y}}^m} P(\hat{\mathbf{y}}^m | f_c(\mathbf{X}))$. The joint learning of multiple tasks follows the general definition of MTL, where the total loss of multiple tasks is optimized together.

3.2. Coordinate Transformation

For a recognition model, the basic assumption for the data instances are Independent and Identically Distributed (i.i.d.) so that samples in training and test sets can be selected randomly

and uniformly. Supposing the smartphones are always placed in fixed orientations, the original acceleration data measured along with a built-in standard 3-axis coordinate (in Figure 2a) of the smartphone are satisfied with that assumption. As a result, using original acceleration data as input, it is easy to obtain a relatively reliable recognition model. Meanwhile, since the acceleration pattern for a specific movement is consistent, a clear physical interpretation can be found for movements and used for recognition. For example, when a user holds a smartphone in portrait orientation, a clear physical interpretation can be made to the measurement on these three axes. For example, jumping or taking an elevator can be measured on the *y*-axis; horizontal movement such as skating or riding a bus can be measured on the *x* or *z*-axis. Stepping or running could be measured in all three axes in which one or two would be dominant and have a significantly larger value.

(**a**) (**b**)

Figure 2. Two coordinate systems defined in Android smartphones: (**a**) Built-in standard 3-axis coordinate system: *x* axis is horizontal and pointing to the right of device; *y* axis is vertical and pointing to the up of device; *z* axis is vertical to *x* and *y* and pointing out of screen; (**b**) Earth coordinate system: *x* axis is tangential to the ground at the device's current location and points approximately East; *y* is tangential to the ground at the device's current location and points toward the geomagnetic North Pole; and *z* points toward the sky and is perpendicular to the ground plane.

However, in practice, users may freely place smartphones on different orientations, which has a negative effect on recognition. With the smartphone orientation variation, the original acceleration values on three dimensions will also change, so that the acceleration data of the same movement presents different patterns, as shown in Figure 3a–d. in other words, the data of the same movement is actually sampled from different coordinate spaces and belongs to different data distributions. Therefore, using original data from undefined orientations as input in the testing stage is equivalent to recognizing data instances Out-of-Distribution, which will dramatically degrade the performance. The intuitive interpretation is that the physical interpretation developed above for original acceleration data will be untenable.

To build a recognition model robust to smartphone orientation variation, the key idea is to transform all training and testing instances to the same data distribution. The intuitive approach is to find an appropriate coordinate system in which a physical interpretation can always be relied on. One solution is to track the users' motion in a global reference frame, which aims to transform the original acceleration data into a fixed earth coordinate system [43] (as shown in Figure 2b). After transformation, the measurement of acceleration relies on earth coordinate axes and belongs to a unified data distribution, which is independent of smartphone orientation. In our study, all acceleration data will be transformed into earth coordinates before recognition.

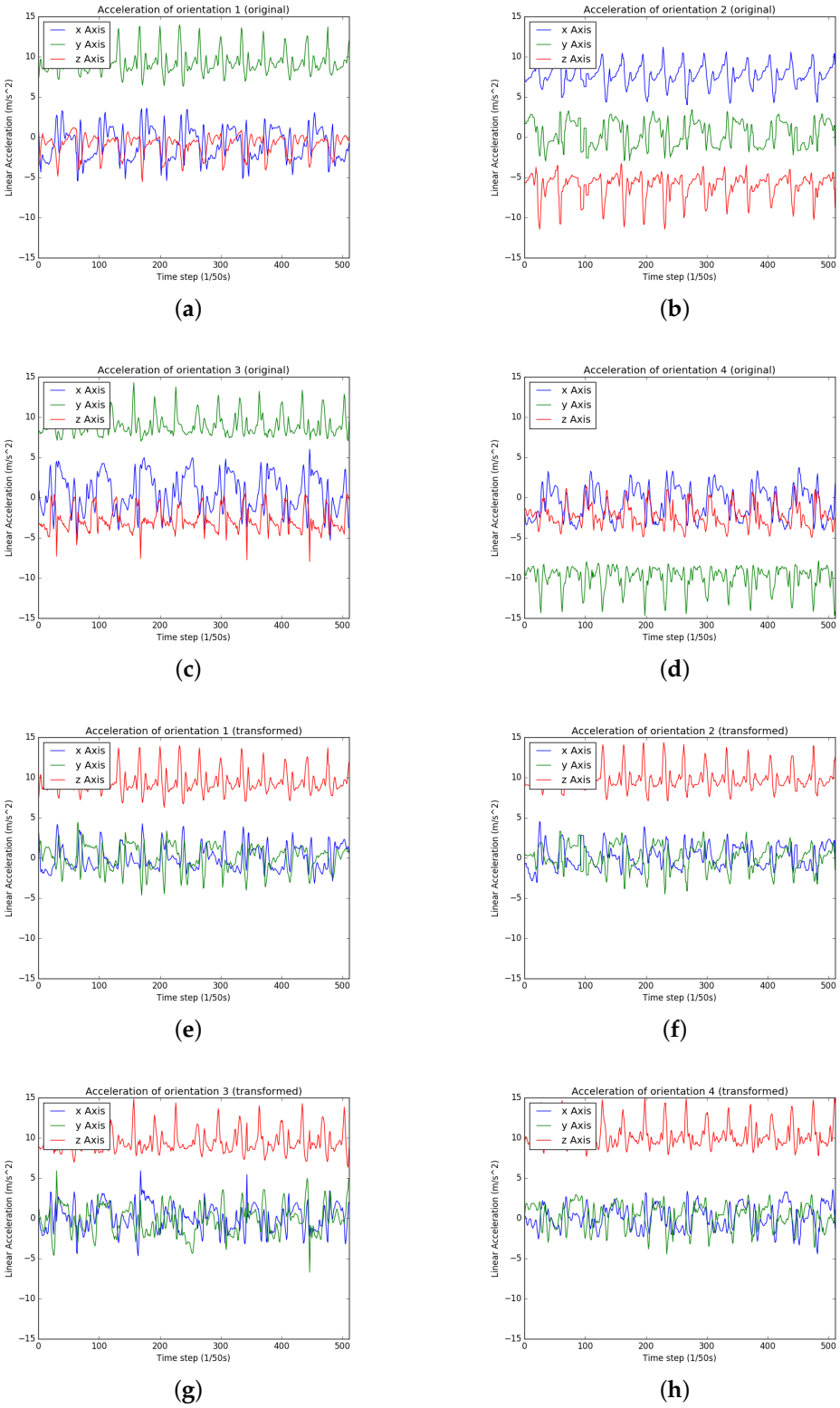

Figure 3. While a user is walking with a smartphone placed in the pocket, the acceleration data of 4 orientations are shown. The (**a**–**d**) are visualizations of original acceleration data measured in the built-in standard 3-axis coordinate system. With the same movement, the data from different orientations perform differently. In the (**e**–**h**), the transformed original acceleration data under the earth coordinate system shows the consistent pattern.

The coordinate transformation technique applied in this study is a quaternion-based method. It is commonly used in a location estimation system for transforming acceleration measurement and calculating device position under the world coordinate system. Compared with the Euler angle-based approach that may face the problem of gimbal lock and has been used by Ustev et al. [43], the quaternion base method is more effective and convenient. The simple form of quaternion vector collected from rotation vector senor is expressed as: $\mathbf{q}_t = [cos(\theta/2), v^{(x)}sin(\theta/2), v^{(y)}sin(\theta/2), v^{(z)}sin(\theta/2)]^\top = [a, b, c, d]^\top$. The key to transformation is the transformation matrix that can be calculated by description of Diebel et al. [44] :

$$\mathbf{R}_t = \begin{pmatrix} 1-2c^2-2d^2 & 2bc-2ad & 2bd+2ac \\ 2bc+2ad & 1-2b^2-2d^2 & 2cd-2ab \\ 2bd-2ac & 2cd+2ab & 1-2b^2-2c^2 \end{pmatrix} \tag{1}$$

Continually transforming original acceleration data to earth coordinate system is given by:

$$\mathbf{e}_t = \mathbf{R}_t \mathbf{s}_t \tag{2}$$

where $\mathbf{e}_t$ is the output in earth coordinate system, and $\mathbf{s}_t$ is the original acceleration data measured in the device coordinate system.

3.3. Multi-Task Learning for Joint Recognition

MTL leverages useful information among tasks by jointly optimizing their training objectives. Each training objective represents a hypothesis made from a task and delivers a training signal to guide the update of model parameters. From the perspective of machine learning, MTL can be viewed as a form of inductive transfer. Inductive transfer improves a model by introducing an inductive bias, which leads to a model tending to prefer the hypothesis of a specific task. In MTL, the inductive biases will be provided by all the tasks, making the model prefer hypotheses that can explain all tasks [10].

MTL explores the commonalities and differences among tasks, where two types of data representations are maintained in the model. A shared representation denotes the commonalities among tasks, which is interpreted as the learned common knowledge. It will improve the generalization ability of the model and reduce the risk of overfitting on specific task [10]. In contrast, task-specific representations reveal differences discovered among tasks. As a result, MTL has the potential to improve performance while effectively limiting the number of employed parameters in the model.

To perform MTL, we make use of a hard parameter sharing method to learn shared and task-specific data representation. The architecture of our joint recognition model is illustrated in Figure 4, including shared and task-specific layers.

On the input side of the model, a backbone network is composed of several shared hidden layers parametrized by θ. The backbone network follows supervised information propagated from all tasks to learn its parameters, in order to generate a shared representation vector $\mathbf{v} \in \mathbb{R}^{d^s}$. In this study, we will investigate several fashioned neural network architectures for representation learning as backbone networks, and the detailed structure will be introduced in Section 3.4.

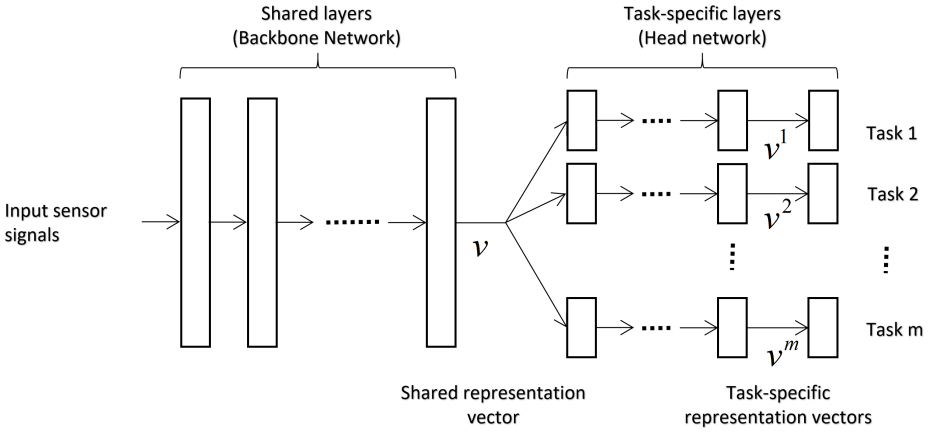

Figure 4. Hard parameter sharing for Multi-Task Learning integrated in neural networks.

After that, in the head network, each task is assigned with a branch of task-specific layers. For the m-th task, its task-specific layers, parameterized by θ^m, include some hidden layers for producing task-specific representation vector and an output layer for final prediction. The joint recognition model outputs all predictions simultaneously, which means there is no prioritization between tasks in our study. Specifically, in each hidden layer, a fully connected layer followed by a Rectified Linear Unit (ReLU) function is used to transform $\mathbf{v}$. With few layers, a task-specific feature $\mathbf{v}^m \in \mathbb{R}^{d^m}$ will be produced, where the dimension is uniformly set as $d^m = 100$. In the final output layer, a fully connected layer first generates an unnormalized probability vector $\mathbf{a}^m$, and then a $softmax$ function outputs the predicted probability distribution. The probability for each class is given by:

$$P(\hat{\mathbf{y}}^m_{(u)}|\mathbf{E}) = \frac{e^{\mathbf{a}^m_{(u)}}}{\sum_{u=1}^{c^m} e^{\mathbf{a}^m_{(u)}}}, \tag{3}$$

where the $\mathbf{a}^m_{(u)}$ and $\hat{\mathbf{y}}^m_{(u)}$ is the u-th element among c^m. The predicted class label is assigned to the one with the highest probability, i.e., $\arg\max_{\hat{\mathbf{y}}^m} P(\hat{\mathbf{y}}^m|\mathbf{E})$).

In the training stage, the recognition model is trained by minimizing the discrepancy between predictions and labels. The cross-entropy cost function is used as the training objective to reflect the discrepancy between the prediction and ground truth:

$$\mathcal{L}^m = \sum_{n=1}^{N} H(\mathbf{y}^m_n, f^m(\mathbf{E}_n; \theta, \theta^m)) \tag{4}$$

where $H(\cdot,\cdot)$ is the cross-entropy for two distributions, and $f^m(\cdot)$ parameterized by θ and θ^m is a task-specific mapping function representing the predicted conditional probability distribution $P(\hat{\mathbf{y}}^m|\mathbf{E})$, and subscript n denotes the n-th training sample among N. The optimal parameter θ^* can be obtained by jointly minimizing loss functions of all tasks on the training dataset:

$$\theta^* = \arg\min_{(\theta,\theta^1,\ldots,\theta^M)} \Big(\sum_{m=1}^{M} \eta^m \mathcal{L}^m \Big) \tag{5}$$

$$\eta^m = \frac{c^m}{\sum_{m=1} c^m}, \forall m \in 1, \ldots, M \tag{6}$$

where η^m is the weight of m-th task.

3.4. Backbone Networks

We plan to exploit three kinds of neural network architectures as backbone networks separately. The designed architectures have different representational abilities to sequential data, including the

Multilayer Perceptron (MLP) from classical models, and Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) from deep learning models. The input data instance, acceleration data under earth coordinate system, is considered to be sequential data $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_t, \ldots, \mathbf{e}_T]$ with a length T, whose each time-step contains three axes' values $\mathbf{e}_t = [e_t^{(x)}, e_t^{(y)}, e_t^{(z)}]^\top$.

3.4.1. Multilayer Perceptron (MLP)

MLP is a type of simple artificial neural network consisting of only feed-forward neurons. It can take input from the previous layer and execute the non-linear transformation with hidden layers.

A 3-layer MLP takes a hand-crafted statistical feature vector as input, as shown in Figure 5. To capture the statistical information of sequential data, 13 kinds of feature values are calculated on each axis: mean, variance, standard deviation, minimum value, max value, skewness, kurtosis, jitter, mean value crossing rate, mean of autocorrelation, standard deviation of autocorrelation, mean of autocovariance, and standard deviation of autocovariance. All feature values of three axes are concatenated to form the input feature vector with the dimension $d = (3 \times 13)$. The hidden layers are fully connected layers followed by ReLU activation functions, and the number of neurons in each layer is empirically stetted as 512.

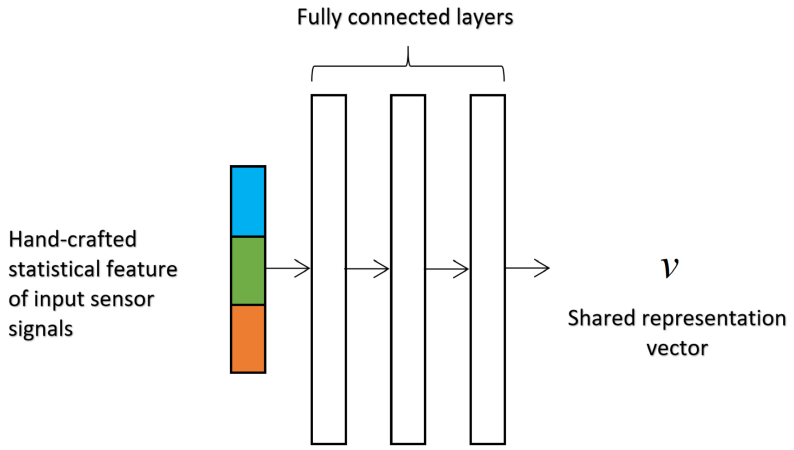

Figure 5. Architecture of simple Multilayer Perceptron with hand-crafted statistical feature as input.

3.4.2. Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) is a popular deep feed-forward neural network that is inspired by the natural visual perception mechanism of the living creatures. The main components of CNN are designed to take advantage of the 2D structure of an input image, including convolution operations for feature extraction and pooling operations for down-sampling. The convolutional layers exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. Combined with pooling layers, CNN can learn discriminative data representations, which leads to extremely effective analysis systems.

CNN has great potential to identify salient patterns from sequential data. For this purpose, a CNN architecture, stacking multiple one-dimensional convolutional layers and pooling layers, can be adapted to sequential data. Each one-dimensional convolutional layer contains several convolutional filters that have two valuable characteristics. One is local connectivity pattern. A filter can capture local temporal dependency of sub-sequence for feature extraction. After a filter sliding across the entire sequence, a feature map for the whole sequence will be derived based on a set of subsequences. Another one is the shared parameters mechanism. It makes a filter learn a specific response field for inputs, in order to give the strongest response to inputs with the salient pattern. Moreover, combined with the pooling layer, only the discriminative feature produced from a salient pattern will be delivered to subsequent layers. As a result, the convolution operation allows CNN to identify the local salience of

the inputs regardless of their positions in the sequence. This mechanism endows the obtained feature with translation invariance. Meanwhile, with several convolutional filters (specified by different parameters), multiple salient patterns learned from different aspects are jointly considered in the CNN.

As shown in Figure 6a, the one-dimensional convolution in layer-1 operates on sliding windows of elements (width k). The convolutions in deeper layers are defined in a similar way. The hyper-parameter of designed CNN architecture is shown in Figure 6b, which is an example of employing sensor signals within window size 50 as input. In general, CNN processes the input sequential data layer by layer, where each layer uses the output feature maps of the previous layer as input. In layer-(l), the feature maps produced by $F^{(l)}$ one-dimensional convolutional filters are considered to be a sequence, e.g., $\mathbf{Z}^{(l)} = [\mathbf{z}_1^{(l)}, \dots \mathbf{z}_t^{(l)}, \dots \mathbf{z}_{T^{(l)}}^{(l)}]$ with a length $T^{(l)}$, where the $\mathbf{z}_t^{(l)} \in \mathbb{R}^{F^{(l)}}$ represents the feature vector at time-step t. Formally, in layer-$(l+1)$, each feature element in $\mathbf{z}_t^{(l+1)}$ produced by a one-dimensional convolution filter is defined as:

$$z_t^{(l+1)} = \sigma\Big(\sum_{f=1}^{F^{(l)}} \hat{\mathbf{z}}_t^{(l,f)} \mathbf{w}^{(l+1,f)} + b^{(l+1,f)}\Big), \tag{7}$$

where

- F^l is the number of feature maps in layer-(l);
- $\mathbf{w}^{(l+1,f)} \in \mathbb{R}^k$ is a weight vector in filter that covers the f-th feature map in layer-(l), and k is filter size;
- $b^{(l+1,f)}$ is a bias value for the f-th feature map;
- $\hat{\mathbf{z}}_t^{(l,f)} \in \mathbb{R}^{1\times k}$ denotes the f-th feature map in a subsequence in layer-(l); corresponds to the entire subsequence including all feature maps is defined as $\hat{\mathbf{Z}}_t^{(l)} = \mathbf{z}_{t:t+k-1}^{(l)} = [\mathbf{z}_t^{(l)}, \dots, \mathbf{z}_{t+k-1}^{(l)}]$.

Additionally, the feature maps in the previous layer are pooled over the local temporal neighborhood by the max pooling operation, which progressively reduces the spatial size of representations.

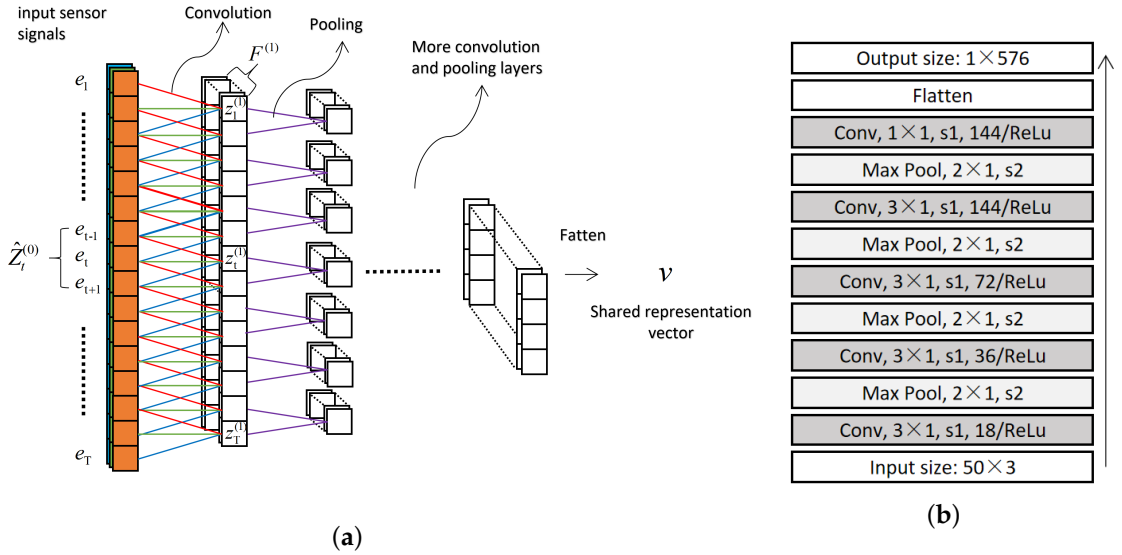

Figure 6. Architecture of designed Convolutional Neural Network: (**a**) One-dimensional convolution and max pooling; (**b**) Hyper-parameter of designed architecture that is an example employing sensor signals within window size 50 as input. The "Conv, 3 × 1, s1, 18/ReLU" denotes a 1D convolutional layer with filter size 3, stride 1, filter number 18, ReLU activation function. The "Max Pooling, 2 × 1, s2" represents a 1D max pooling layer with size 2, stride 2.

3.4.3. Long Short-Term Memory (LSTM)

The Long Short-Term Memory network is extended from the Recurrent Neural Network (RNN). RNN is an artificial network where connections between computation units form a directed cycle, which allows information to flow forward along the sequence. The RNN can use its computation units to achieve dynamic temporal memory and learn the temporal context of input data within the sequence. In addition, LSTM improves the computation unit of RNN, which selectively memorizes the temporal information by introducing the gate mechanism.

The LSTM has an inherent advantage in capturing temporal order information of sequence. This is achieved by the recurrent connection between LSTM units that captures the long-term temporal dependency of sequence. The LSTM unit maintains a hidden state $\mathbf{h}_t$ for every element of a sequence to selectively store/access the sequential information changing over time. Meanwhile, it recurrently performs the same transition function on each hidden state, which relies on the recorded information of previous elements. At time-step t, a simple form of transition function is denoted as:

$$\mathbf{h}_t = LSTM(\mathbf{e}_t, \mathbf{h}_{t-1}), \tag{8}$$

based on the new input $\mathbf{e}_t$, and $\mathbf{h}_{t-1}$ from the previous LSTM unit. As a result, the recurrent connection grantees that the LSTM learns the temporal relationships of sequence. In addition, to manipulate the temporal relationship on a long scale, a gate mechanism is introduced in the LSTM unit.

In detail, an LSTM unit includes four components: a memory state $\mathbf{c}_t$ that can be updated, erased, and read out, three kinds of gates that control the information flow, e.g., the input gate $\mathbf{i}_t$, output gate $\mathbf{o}_t$, and forget gate $\mathbf{f}_t$ are used to control the reading, writing, and memory updating respectively. For a one-layer LSTM, at time-step t, except for $\mathbf{e}_t$ and $\mathbf{h}_{t-1}$, the memory state $\mathbf{c}_{t-1}$ from the previous LSTM unit also participates into current update. The updating mechanism of an LSTM unit is shown as below:

$$\mathbf{i}_t = sigmod(\mathbf{W}_i\mathbf{e}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{9}$$

$$\mathbf{f}_t = sigmod(\mathbf{W}_f\mathbf{e}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{10}$$

$$\mathbf{o}_t = sigmod(\mathbf{W}_o\mathbf{e}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{11}$$

$$\tilde{\mathbf{c}}_t = tanh(\mathbf{W}_c\mathbf{e}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{12}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \tag{13}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ tanh(\mathbf{c}_t) \tag{14}$$

where $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_c, \mathbf{W}_o, \mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_c, \mathbf{U}_o$ are weights matrix, $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o$ are bias-vectors. The operator $\circ$ stands for element-wise multiplication.

A standard practice for modeling sequence using LSTM is shown in Figure 7. The sensor signals are processed by a 2-layer LSTM in a sequential way from the beginning to the end. At each time-step, the LSTM unit takes the triaxial sensor values as input; and the hidden unit of the LSTM unit is set as 100. The last hidden state that models the dependency relationship of all previous data in layer-2 is considered to be the learned representation of the whole sequence.

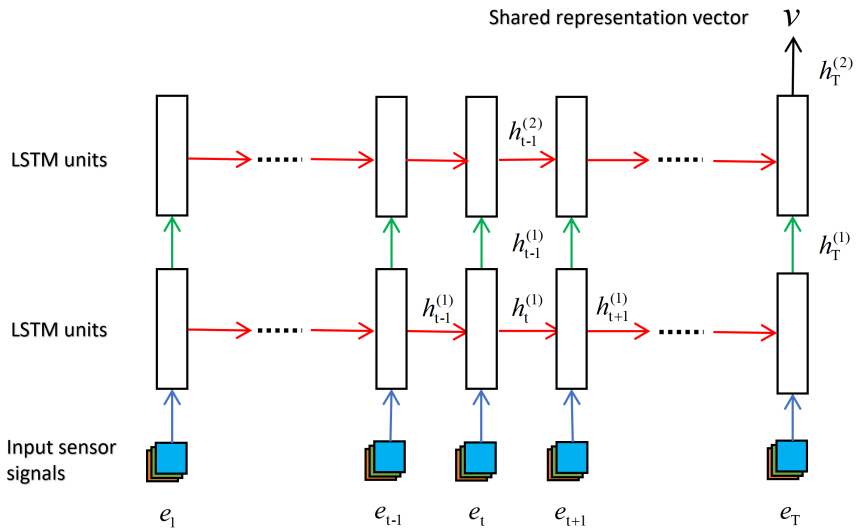

Figure 7. Brief illustration of designed Long Short-Term Memory network.

4. Results and Analysis

In this section, we evaluate our framework for joint activity and the position recognition task. The detailed information of experimental datasets is introduced first. The experimental settings are given, including basic evaluation rules and optimization hyper-parameters. Following that, the benefits of two main components in our framework, coordinate transformation and joint recognition with MTL strategy will be reported separately.

4.1. Dataset

The two components of our framework will be evaluated on two datasets individually. The widely investigated human activities and smartphone positions in various kinds of literature are investigated in these datasets.

To verify our solution to the smartphone orientation variation problem, we collected a dataset labeling the sensor data with smartphone orientations, positions, and human activities. We collected a dataset with motion sensor data of 4 activities from 2 subjects. The human activities include walking, ascending stairs, descending stairs, and running. For each activity, there were 3 different Android devices placed at different body positions: a *Huawei Nova 2s* placed in a pants pocket, a *Samsung A9 Star* held in hand, a *XiaoMi RedMi 4X* inside backpack pocket. The smartphones in pants pockets and hand are placed on 4 orientations; and 8 orientations are labeled when a smartphone was stored in a backpack. In total, each activity was measured for around 80 min with a frequency of 50 Hz. The sensor type includes acceleration, gyroscope, magnetic, gravity, linear acceleration and rotation vector. Although this dataset covers a relatively small number of activities, on-body positions, and users, it is still fair enough for testing algorithm considering the smartphone orientation in the recognition task. This is because investigating the problem of smartphone orientation variation is not affected by other factors.

To evaluate our joint recognition strategy, we exploited the RealWorld HAR [6] dataset which collected the sensor data of 8 activities in 7 on-body positions from 15 volunteers. To the best of our knowledge, this dataset includes the most smartphone positions so far. The experimental data was gained from different physical characteristics of eight males and seven females (age 31.9 ± 12.4, height 173.1 ± 6.9, and weight 74.1 ± 13.8). The activities performed by each one includes climbing downstairs (A1), climbing upstairs (A2), jumping (A3), lying (A4), standing (A5), sitting (A6), running/jogging (A7), and walking (A8). Every user was equipped with a set of smartphones (*Samsung Galaxy S4*) and a smart-watch (*LG G Watch R*). These devices were located on seven different on-body positions (chest (P1), forearm (P2), head (P3), shin (P4), thigh (P5), upper arm (P6), and waist (P7)). For each activity,

the sensor data on different on-body positions were collected concurrently at a sampling rate of 50 Hz. The more important, the recorded videos show that the data was collected in a real-world setting. For instance, users could stroll in the city or jog in the forest; the users' movements were performed in their preferred ways, such as walking at different speeds, sitting/standing while eating, or holding the phone. However, in this dataset, the subjects commonly place the smart devices in fixed orientations and the rotation vector sensor is unavailable. Therefore, it cannot be used for verifying our assumption about coordinate transformation.

4.2. Experimental Setting

We defined an experimental setting following several rules. First, *F*1-score was seen as the evaluation metric to report the performance of recognition models. Second, without special instructions, the default evaluation mode was *10-fold cross validation* with stratified sampling to guarantee that all folds own the same ratio of target classes. Apart from that, we respectively reported the performances of three types of neural network architectures that have different representational abilities. It is worth mentioning that recognition models are all subject-dependent models, where the training and test data are attached to the same subjects. Considering this, the results given in the next sections were calculated by the aggregating results of all subjects.

4.3. Optimization Hyper-Parameters

The machine learning model requires several hyper-parameters in the optimization procedure. Normally, an optimal approach for setting the hyper-parameters is to select them from a set of hyper-parameter candidates by prior experiences, such as grid search. However, due to large computational demands for deep learning architecture, the aforementioned hyper-parameter search method is infeasible.

In this study, we set manually the same hyper-parameters given to all the recognition models. The maximum training epoch was set to 1000. In each epoch, the recognition model was trained with several iterations, where each iteration contains a batch of training data with batch size 64. To minimize the joint cost function, we applied a stochastic gradient descent algorithm by using Adam optimizer [45] with a fixed learning rate of 10^{-4}. The gradient was clipped by setting the global norm value [46] as 5. Meanwhile, there was a 20% dropout [47] on learned representation before inputting the output layer. We used early termination to select the best model. When facing the divergence, i.e., the loss of model on testing dataset no longer decreases more than 50 epochs, we selected the model with the minimum loss on the testing dataset.

4.4. Coordinate Transformation

In this section, the coordinate transformation as a preprocessing approach will be evaluated by using the collected dataset. To this end, the acceleration data before and after coordinate transformation will be employed as an input of the model separately. The baseline performance was built by using a *10-fold cross validation* where the training and test data are Independent and Identically Distributed (i.i.d.). Moreover, a *Leave-one-orientation-out cross validation* was taken as main evaluation mode that samples the training and test data from exclusive orientations. Additionally, the coordinate transformation method was quantitatively evaluated by changing the different window sizes of acceleration data.

The benefit of coordinate transformation can be observed from Figure 8 that shows *F*1-scores of single-task models for activity and position recognition. The input data instances of these models are original or transformed acceleration data within 10 seconds. Using original acceleration data without the coordinate transformation, recognition models in *10-fold cross validation* produce a strong baseline. However, the performance in *Leave-one-orientation-out cross validation* decreases dramatically. As discussed in Section 3.2, such phenomenon is caused by smartphone orientation variation problem which produces test data samples Out-of-Distribution. After a coordinate transformation, there is no

longer any apparent gap between two evaluation modes. It is obvious that coordinate transformation improves the generalization ability of models on test dataset collected from unseen smartphone orientations. This is mainly achieved by transforming all acceleration data to a unified earth coordinate system so that the data of all orientations are expressed with identical data distribution. Although our dataset contains a relatively small number of activities and positions, these results still are persuasive because the coordinate transformation technique is independent of activities, positions or subjects.

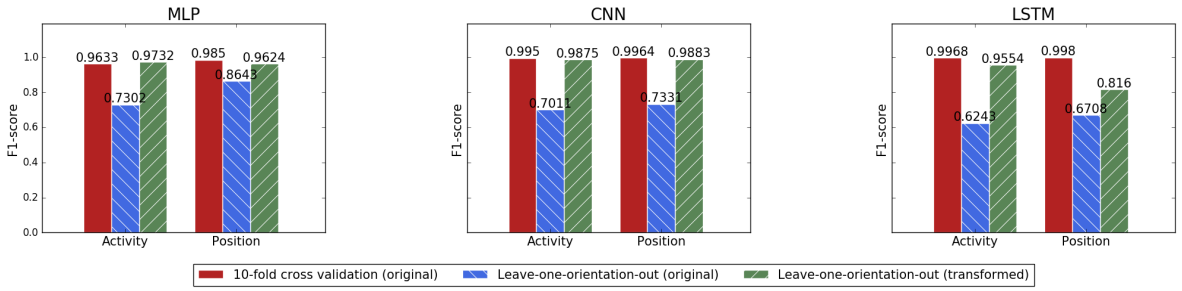

Figure 8. Performance of recognition models using original or transformed acceleration data in two different evaluation modes.

The results in Figure 9 are presented for evaluating the impact of window size with respect to coordinate transformation. All results were produced under the *Leave-one-orientation-out cross validation* in which the acceleration data under the earth coordinate system are taken as input. Given the acceleration data with different window sizes $\{10.0s; 7.5s; 5.0s; 2.5s\}$, the corresponding total numbers of data instances are $\{8005; 10,766; 16,265; 32,840\}$ respectively. In general, the longer the window size, the higher the recognition performance will be derived. Interestingly, the results of models present different changing tendencies along with progressively shrinking window sizes. The performance of CNN and MLP tend to descend when smaller window sizes were employed. In contrast, the performances of LSTM were improved with shortening the sequence length and reached the highest score at the 2.5*s*.

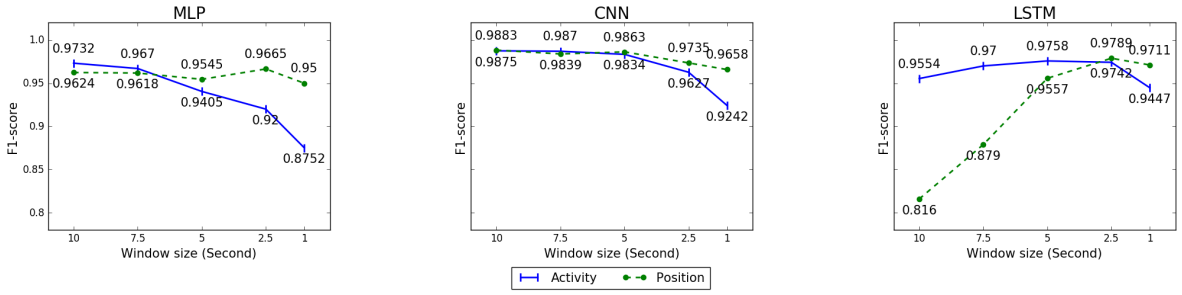

Figure 9. A quantitative analysis for coordinate transformation by using acceleration data within different window sizes.

4.5. Multi-task Learning for Joint Recognition

We evaluated our joint recognition approach with the RealWorld HAR dataset [6] which contains more human activities, smartphone positions, and subjects. To make a fair comparison, we adopted the same setting in the original research [6]. When the window size is one-second-long and overlaps in half, the number of data instances is 892, 808.

The performances of our solution were evaluated by three types of experiments. First, we contrasted our results with the original results, the state-of-the-art solutions [6] which aim to recognize dynamic and stationary activities by employing smartphone positions as prior knowledge. Second, to show the advantage of our method, we added an extra related task to identify smartphone users from human movement. Finally, considering that jointly recognizing smartphone positions and users

from human movement is more reasonable, we applied our solution only on dynamic activities for evaluation of actual use.

To further examine how MTL leverages useful information from related tasks, the experiments were conducted on two types of learning strategies. Taking results in Table 1 as an example, in each type of model, the **ST** and **MT** respectively denote the single-task model and joint recognition model-based MTL strategy, and the number of task-specific layers (hidden layers of classifiers) is marked at the end of learning types. For instance, the **ST-3L** represents a single-task model using a backbone network for feature extraction and 3-layer-classifier for prediction; the **MT-0L** and **MT-3L** denote the multi-task models using 0 and 3 task-specific layers receptively. However, the **MT-0L** means using shared representation directly as input for multi-task recognition. Correspondingly, the model of **MT-3L** further transforms a shared feature vector to multiple task-specific features, where each task has 3 simple non-linear hidden layers. Additionally, the **S-A/P** refers to results of subject-specific and activity/position-specific model used by Sztyler et al. [6].

Table 1 presents *F*1-scores of models when the sensor data are collected from stationary and dynamic activities. The performance of our models all surpass the original results significantly. Our best *F*1-scores of activity and position recognition are around **0.89** and **0.98**, which much higher than **0.84** and **0.89** of **S-A/P**, even with a simple MLP model that was also used by Sztyler et al. [6]. Moreover, based on the MTL strategy, using one global model for joint recognition is conducive to reducing the number of models than **S-A/P**. The **S-A/P** is a position-aware HAR solution whose best performances were produced by stacking three levels of Random Forest classifiers: in the beginning, dynamic and static activities were distinguished using first-level classifiers; next, the second-level classifier was used to identify where the smartphone is placed on the human body; the third-level included a set of activity recognition classifiers each belonging to a specific position. Meanwhile, all their results were reported under the subject-specific models, where a position-aware model was trained for each subject on the data of all activities and positions.

Table 1. Recognition rates (*F*1-score) of human activity, smartphone position and smartphone user when using sensor data of all activities.

Model Type	Learning Type	Activity	Position	User	Average
MLP	**ST-3L**	0.8837	0.9852	0.8364	0.9018
	MT-3L	**0.8910**	**0.9866**	**0.8484**	**0.9087**
CNN	**ST-3L**	0.8714	**0.9862**	0.8181	0.8919
	MT-3L	**0.8855**	0.9848	**0.8300**	**0.9001**
LSTM	**ST-3L**	**0.8932**	**0.9810**	0.8511	**0.9084**
	MT-3L	0.8919	0.9802	**0.8515**	0.9079
Random Forest	**S-A/P**	0.84	0.89	–	0.865

Although the models can maintain a certain degree of accuracy, their performances still can be further corrected because of the existing sensor data of stationary activities (e.g., lying (A4), standing (A5) and sitting (A6)) seem to confuse the classifiers. For example, identifying smartphone positions or users from stationary activities is not reasonable. In this case, the models still work due to the gravity measurements contained in acceleration data which might be useful. Therefore, to clearly observe the feasibility, we further evaluated our models on only dynamic activities.

The performance of all tasks is both improved by only employing dynamic activities data in Table 2. Especially the activity recognition and user identification in **LSTM-MT-3L**, the *F*1-scores have reached to **0.9240** and **0.9174** respectively. Meanwhile, the best average *F*1-scores are all raised than results in Table 1, e.g., **+3.71%** (**LSTM-MT-3L**), **+2.31%** (**CNN-MT-3L**) and **+0.93%** (**MLP-MT-3L**). It suggests that jointly mining human activity, smartphone position and user information from dynamic movement is more reasonable in practice, which is in line with the intuition.

Table 2. Recognition rates (*F*1-score) of human activity, smartphone position and smartphone user when using sensor data of dynamic activities.

Model Type	Learning Type	Activity	Position	User	Average
MLP	**ST-3L**	0.8968	**0.9859**	0.8625	0.9151
	MT-0L	0.8536	0.9742	0.8129	0.8802
	MT-3L	**0.9002**	0.9857	**0.8680**	**0.9179**
CNN	**ST-3L**	0.9019	**0.9893**	0.8693	0.9202
	MT-0L	0.8880	0.9831	**0.8836**	0.9183
	MT-3L	**0.9067**	0.9854	0.8774	**0.9232**
LSTM	**ST-3L**	**0.9265**	**0.9917**	0.9115	0.9432
	MT-0L	0.9080	0.9878	0.9043	0.9334
	MT-3L	0.9240	0.9906	**0.9174**	**0.9440**

The benefit of MTL can be observed by contrasting single-task models with joint recognition models. On the whole, the promotions of all average *F*1-scores are slightly higher than the single-task models except for **LSTM-MT-3L** in Table 1. The highest one achieved is **+0.82**% in **CNN-MT-3L** in Table 1. For specific tasks, such as user identification, all performances are consistently improved under joint models, where the maximum improvement is **+1.21**% in **MLP-MT-3L** in Table 1. Even the MTL strategy cannot perfectly improve the performance of all tasks, it works well for the reduction of computation demand and latency by efficiently leveraging shared parameters. In contrast with single-task models, the joint model only employs one backbone network but achieves comparable results. Additionally, adding task-specific layers has an obvious effect on results. As can be seen, the performances in learning type of **MT-0L** in Table 2 are not optimal because the shared feature of multi-tasks commonly plays a role for regularization in small size dataset.

5. Discussion

Jointly mining multiple physical context information from the motion sensor is a relatively new research question. Despite its importance, there is very little research that investigates the same question on the RealWorld HAR dataset. Moreover, to our knowledge, the position-aware HAR presents one of the best approaches to the benchmark dataset of RealWorld HAR. Therefore, we focus on comparing this state-of-the-art approach with our proposed joint method. In our paper, we actually implemented three different models (in both the single-task and multi-task setting) to solve this problem. In general, the LSTM-based joint model could outperform the other three competitors including MLP, CNN, and the Random Forest.

The experimental results show that our approach can improve the model's generalization ability on the data collected from unseen smartphone orientations. Although the proposed coordinate transformation method still lacks accuracy compared to the state-of-the-art solutions, we must highlight that it is still feasible. Most of the commonly used coordinate transformation methods may suffer from the problem of error propagation causing by gravity pollution, magnetic interference, inherent sensor noise and so on. For instance, as investigated by Shen [48], the rotation matrix calculation is most accurate when measured in the stationary position because it assumes only acceleration due to gravity is present. In the research community of location estimation and orientation tracking, many efforts have been made to minimize existing errors [48,49]. In terms of our joint recognition task, the minimum requirement for our coordinate transformation technique is to recover the consistent acceleration pattern and keep its periodicity. Although the error propagation may be inevitable, this will not affect the periodicity and consistency of acceleration patterns, as shown in Figure 3. At the same time, the experimental results also verify the irrelevance of error propagation to our application.

We applied our framework on three types of widely used neural networks to provide learned lessons for future research. One of interesting results is that models illustrate different robustness to changing the sampling window size in Figure 9, such as the apparent performance gap between CNN

and LSTM. Practically, large length of sequence makes LSTM hard to train when there is no essential method to maintain the memorized information. One convenient remedy is extracting features from ordered subsequences before applying LSTM, e.g., a hybrid model [50] combining CNN and LSTM. On the other hand, in Table 1, there is no significant gap among different models and the simple MLP achieves comparable results with deep models. Such a phenomenon lays on the other side of the recent popular belief, employing deep learning models with complex architecture and high representational capability is the first candidate for HAR. On the contrary, the models clearly demonstrate different representational abilities when the data of stationary activities are removed in Table 2. Apparently, the LSTM is a suggested model since it is superiors to others.

The deep neural network models are computationally expensive and memory-intensive, which impedes their wide deployment in devices with restricted resources. In recent years, compression and acceleration for deep neural networks have become a valuable research topic [51]. Our approach leveraging shared architecture for computation demand reduction is theoretically feasible. To further support this solution, we calculate the number of model parameters in Table 3. Comparing the single-task model (**ST-3L**) with the multi-task model (**MT-3L**), the total number of parameters is reduced significantly. In experiments, the model hyper-parameters were configured empirically to ensure high recognition performance. Actually, further cutting parameters can be achieved by carefully changing or searching for their hyper-parameters. For instance, it is valuable to shrink the model size of MLP by finding reasonable hyper-parameters for hidden neurons and hidden layers in a fully connected layer.

Table 3. The number (million) of parameters employed by single-task models and multi-task model of each type.

Model Type	ST-3L	MT-3L
MLP	0.62 m × 3	0.76 m
CNN	0.14 m × 3	0.29 m
LSTM	0.19 m × 3	0.26 m

6. Conclusions and Future Work

In this study, we presented a framework for jointly mining human activity and smartphone position from motion sensors. This framework can be used as an enhanced context-aware service to improve many existing sensing applications. We proposed a data preprocessing approach to eliminate the negative effect of smartphone orientation variation on recognition. It is a coordinate transformation technique based on quaternion, which leverages the acceleration sensor and rotation vector sensor in an Android smartphone. Our method transforms the original acceleration data to a global earth coordinate system that is independent of smartphone orientation. We evaluated the proposed approaches with a collected dataset that contains labels not only for human activity and smartphone position but also for smartphone orientations. The evaluation results illustrated that the proposed method improved the generalization ability of our model on different orientations' data.

On the other hand, a joint recognition model is proposed to output the results of multiple tasks. We designed this model at the base of the MTL strategy to explore the commonalities and differences among related tasks. It is different from those previous widely investigated approaches that involve multiple models and require high computational resources, such as a position-aware model. Our joint recognition model can produce the highest performance by using one global model. In experiments, we show that our approach significantly outperformed the original results on RealWorld HAR dataset [6].

As future work, the enhanced context-aware service can be tested on sensor data collected from a huge amount of smartphone users including the elderly population and children. Such recorded data will provide an opportunity to explore the similarity and difference of human movements when users have diverse physical characteristics. For this reason, we plan to explore the combination of advanced feature extraction technique [52] and MTL strategy. Furthermore, obtaining a joint recognition with a

small amount of label data is also valuable, which might be achieved by integrating the MTL with the approach of learning from few examples [53].

Author Contributions: Conceptualization, Z.G., D.L. and K.H.; methodology, Z.G., D.L. and K.H.; software, Z.G.; validation, Z.G.; formal analysis, Z.G. and K.H.; investigation, Z.G.; resources, D.L. and K.H.; data curation, Z.G., D.L. and K.H.; Writing, original draft preparation, Z.G.; Writing, review and editing, Z.G., D.L., K.H. and Y.H; visualization, Z.G.; supervision, D.L. and K.H.; project administration, D.L. and K.H.; funding acquisition, D.L. and K.H.

Funding: This research was funded by the Natural Science Foundation of China under grant number 61701417 and 61876155,the Natural Science Foundation of Jiangsu Province under grant number BK20181189,XJTLU Research Development Fund under grant number RDF-14-02-45,the CERNET under grant number NGII20161010,XJTLU Key Program Special Fund under grant number KSF-E-05, KSF-A-01 and KSF-P-02.

Acknowledgments: The authors would like to thank the anonymous reviewers for their hard work.

Conflicts 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.

References

1. Coskun, D.; Incel, O.D.; Ozgovde, A. Phone position/placement detection using accelerometer: Impact on activity recognition. In Proceedings of the 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 7–9 April 2015; pp. 1–6.
2. Alanezi, K. *Impact of Smartphone Position on Sensor Values and Context Discovery*; Technical Report; University of Colorado Boulder: Boulder, CO, USA, 2013.
3. Fujinami, K. On-Body Smartphone Localization with an Accelerometer. *Information* **2016**, *7*. [CrossRef]
4. Miluzzo, E.; Papandrea, M.; Lane, N.D.; Lu, H.; Campbell, A. Pocket, Bag, Hand, etc.—Automatically Detecting Phone Context through Discovery. In Proceedings of the PhoneSense, Zurich, Switzerland, 2 November 2010.
5. Martín, H.; Bernardos, A.M.; Iglesias, J.; Casar, J.R. Activity logging using lightweight classification techniques in mobile devices. *Pers. Ubiquitous Comput.* **2013**, *17*, 675–695. [CrossRef]
6. Sztyler, T.; Stuckenschmidt, H.; Petrich, W. Position-aware activity recognition with wearable devices. *Pervasive Mob. Comput.* **2017**, *38*, 281–295. [CrossRef]
7. Yang, R.; Wang, B. PACP: A Position-Independent Activity Recognition Method Using Smartphone Sensors. *Information* **2016**, *7*, 72. [CrossRef]
8. Yan, H.; Shan, Q.; Furukawa, Y. RIDI: Robust IMU double integration. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 621–636.
9. Shin, B.; Kim, C.; Kim, J.; Lee, S.; Kee, C.; Kim, H.S.; Lee, T. Motion Recognition-Based 3D Pedestrian Navigation System Using Smartphone. *IEEE Sens. J.* **2016**, *16*, 6977–6989. [CrossRef]
10. Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1706.05098.
11. Zhang, Y.; Yang, Q. A survey on multi-task learning. *arXiv* **2017**, arXiv:1707.08114.
12. Gadaleta, M.; Rossi, M. IDNet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recognit.* **2018**, *74*, 25–37. [CrossRef]
13. Ren, Y.; Chen, Y.; Chuah, M.C.; Yang, J. Smartphone based user verification leveraging gait recognition for mobile healthcare systems. In Proceedings of the 2013 IEEE International Conference on Sensing, Communications and Networking (SECON), New Orleans, LA, USA, 24–27 June 2013; pp. 149–157. [CrossRef]
14. Lu, H.; Yang, J.; Liu, Z.; Lane, N.D.; Choudhury, T.; Campbell, A.T. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2010; pp. 71–84.
15. Antos, S.A.; Albert, M.V.; Kording, K.P. Hand, belt, pocket or bag: Practical activity tracking with mobile phones. *J. Neurosci. Methods* **2014**, *231*, 22–30. [CrossRef]
16. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
17. Bengio, Y. Deep learning of representations: Looking forward. In *Proceedings of the International Conference on Statistical Language and Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 1–37.

18. Huang, K.; Hussain, A.; Wang, Q.F.; Zhang, R. *Deep Learning: Fundamentals, Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 2.
19. Vepakomma, P.; De, D.; Das, S.K.; Bhansali, S. A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In Proceedings of the 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), Cambridge, MA, USA, 9–12 June 2015; pp. 1–6.
20. Walse, K.H.; Dharaskar, R.V.; Thakare, V.M. Pca based optimal ann classifiers for human activity recognition using mobile sensors data. In *Proceedings of the First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 429–436.
21. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2016; pp. 1533–1540.
22. 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. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
23. Zeng, M.; Nguyen, L.T.; Yu, B.; Mengshoel, O.J.; Zhu, J.; Wu, P.; Zhang, J. Convolutional neural networks for human activity recognition using mobile sensors. In Proceedings of the 6th International Conference on Mobile Computing, Applications and Services, Austin, TX, USA, 6–7 November 2014; pp. 197–205.
24. Ha, S.; Yun, J.M.; Choi, S. Multi-modal convolutional neural networks for activity recognition. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, China, 9–12 October 2015; pp. 3017–3022.
25. Jiang, W.; Yin, Z. Human activity recognition using wearable sensors by deep convolutional neural networks. In Proceedings of the 23rd ACM international conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 1307–1310.
26. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2018**, *119*, 3–11. [CrossRef]
27. Edel, M.; Köppe, E. Binarized-blstm-rnn based human activity recognition. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Madrid, Spain, 4–7 October 2016; pp. 1–7.
28. Almaslukh, B.; AlMuhtadi, J.; Artoli, A. An effective deep autoencoder approach for online smartphone-based human activity recognition. *Int. J. Comput. Sci. Netw. Secur.* **2017**, *17*, 160–165.
29. Wang, A.; Chen, G.; Shang, C.; Zhang, M.; Liu, L. Human activity recognition in a smart home environment with stacked denoising autoencoders. In Proceedings of the International Conference on Web-Age Information Management, Nanchang, China, 3–5 June 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 29–40.
30. Radu, V.; Lane, N.D.; Bhattacharya, S.; Mascolo, C.; Marina, M.K.; Kawsar, F. 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, Heidelberg, Germany, 12–16 September 2016; pp. 185–188.
31. Pang, S.; Liu, F.; Kadobayashi, Y.; Ban, T.; Inoue, D. A learner-independent knowledge transfer approach to multi-task learning. *Cogn. Comput.* **2014**, *6*, 304–320. [CrossRef]
32. Li, Y.; Wang, J.; Ye, J.; Reddy, C.K. A multi-task learning formulation for survival analysis. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1715–1724.
33. He, D.; Kuhn, D.; Parida, L. Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction. *Bioinformatics* **2016**, *32*, i37–i43. [CrossRef] [PubMed]
34. Ahmed, A.; Das, A.; Smola, A.J. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 24–28 February 2014; pp. 153–162.
35. Dong, H.; Wang, W.; Huang, K.; Coenen, F. Joint Multi-Label Attention Networks for Social Text Annotation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 1348–1354.

36. Xiong, W.; Lv, Y.; Cui, Y.; Zhang, X.; Gu, X. A Discriminative Feature Learning Approach for Remote Sensing Image Retrieval. *Remote Sens.* **2019**, *11*, 281. [CrossRef]
37. Qi, K.; Liu, W.; Yang, C.; Guan, Q.; Wu, H. Multi-task joint sparse and low-rank representation for the scene classification of high-resolution remote sensing image. *Remote Sens.* **2016**, *9*, 10. [CrossRef]
38. Chang, T.; Rasmussen, B.P.; Dickson, B.G.; Zachmann, L.J. Chimera: A Multi-Task Recurrent Convolutional Neural Network for Forest Classification and Structural Estimation. *Remote Sens.* **2019**, *11*, 768. [CrossRef]
39. Sun, X.; Kashima, H.; Tomioka, R.; Ueda, N.; Li, P. A new multi-task learning method for personalized activity recognition. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, Vancouver, Canada, 11–14 December 2011; pp. 1218–1223.
40. Sun, X.; Kashima, H.; Ueda, N. Large-scale personalized human activity recognition using online multitask learning. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 2551–2563. [CrossRef]
41. Peng, L.; Chen, L.; Ye, Z.; Zhang, Y. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 74. [CrossRef]
42. Zhao, Z.; Chen, Z.; Chen, Y.; Wang, S.; Wang, H. A class incremental extreme learning machine for activity recognition. *Cogn. Comput.* **2014**, *6*, 423–431. [CrossRef]
43. Ustev, Y.E.; Durmaz Incel, O.; Ersoy, C. User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. In Proceedings of the 2013 ACM conference on Pervasive and Ubiquitous Computing Adjunct Publication, Zurich, Switzerland, 8–12 September 2013; pp. 1427–1436.
44. Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix* **2006**, *58*, 1–35.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
47. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
48. Shen, S.; Gowda, M.; Roy Choudhury, R. Closing the Gaps in Inertial Motion Tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*; ACM: New York, NY, USA, 2018; pp. 429–444. [CrossRef]
49. Zhou, P.; Li, M.; Shen, G. Use it free: Instantly knowing your phone attitude. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–11 September 2014; pp. 605–616.
50. Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; Abdelzaher, T. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 351–360.
51. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges. *IEEE Signal Process. Mag.* **2018**, *35*, 126–136. [CrossRef]
52. Yang, X.; Huang, K.; Zhang, R.; Hussain, A. Learning latent features with infinite nonnegative binary matrix trifactorization. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 450–463. [CrossRef]
53. Zhang, S.; Huang, K.; Zhang, R.; Hussain, A. Learning from few samples with memory network. *Cogn. Comput.* **2018**, *10*, 15–22. [CrossRef]