Lasagna: Towards Deep Hierarchical Understanding and Searching over Mobile Sensing Data

Cihang Liu[*], Lan Zhang[†*], Zongqian Liu[*], Kebin Liu[*], Xiangyang Li[†], Yunhao Liu[*]
[*]School of Software and TNLIST, Tsinghua University, China
[†]University of Science and Technology of China
{cihang, lan, zongqian, kebin, yunhao}@greenorbs.com, xiangyang.li@gmail.com

ABSTRACT

The proliferation of mobile devices has enabled extensive mobile-data supported applications, *e.g.*, exercise and activity recognition and quantification. Typically, these applications need predefined features and are only applicable to predefined activities. In this work, we address the issue of deep understanding of arbitrary activities and semantic searching of any activity over massive mobile sensing data. The challenges stem from the rich dynamics and the wide-spectrum of activities that a human being could perform. We propose a hierarchical activity representation, extract common bases of motion data in an unsupervised manner by leveraging the power of deep neural networks, and propose a universal multi-resolution representation for all activities without prior knowledge. Based on this representation, we design an innovative system **Lasagna** to manage and search motion data semantically. We implement a prototype system and our comprehensive evaluations show that our system can achieve highly accurate activity classification (with precision 98.9%) and search (with recall almost 100% and precision about 90%) over a diverse set of activities.

CCS Concepts

•**Human-centered computing** → **Ubiquitous and mobile computing systems and tools;** *Mobile computing;* •**Information systems** → *Web search engines;*

Keywords

Hiearchical Semanteme; Activity Recognition; Mobile Sensing; Deep Learning; Semantic Based Activity Search

1. INTRODUCTION

Smart mobile devices, including phones and wearables, have become an indispensable part of people's daily life. Recent analysis shows that, in 2015, the global revenue from smartphones is around 272.28 billion dollars, and that of smart wearables reaches 6 billion dollars. Beyond communication, mobile devices have become new sensing platforms [2] [19] [13], which are equipped with rich sensors, *e.g.*, accelerometer, gyroscope, magnetometer, barometer and heart rate monitor. These on-board sensors enable them to act as an interface between the physical and cyber worlds.

Human activity recognition or classification [6, 9, 26, 46, 48, 50] using smart devices has drawn much attention due to its wide usage and pervasiveness. Those work can be categorized into two groups. The first group is based on physical model, such as walking [48] and smoking [26]. The second group relies on machine learning techniques, which trains classification model for some specific activities using a large amount of data [2, 9, 10, 13, 14, 50]. By exploring the mobile sensing data of different activities respectively, some of them achieve over 90% accuracy in certain scenarios. Especially, with the remarkable development of deep learning techniques, some methods [11, 15] can achieve over 98% accuracy for activity classification using supervised deep learning model. However, existing methods are limited in practical usage due to the following reasons. Firstly, the physical model or classification method is only feasible for one or a few predefined activities. They require either pre-knowledge or labeled data for supervised training. In practice, the human activities often exhibit high diversity and unpredictability and the motion data are often complex (*e.g.*, smoking while walking), which make it quite difficult to model or label all activities. Even it is possible to laboursomely explore different activities one after another, it is still inefficient to apply all recognition models on a piece of unknown motion data. We are in bad need of a universal solution for understanding and querying a rich set of activities. Secondly, existing methods only favor some specific granularity of an activity, while hierarchy is a common nature of human activities. For example, at a fine granularity, doing exercises may include walking, running and jumping. Even when zooming in walking, it is composed of arm swinging and stepping forward, and we can go deeper to explore more subtle motions. Neglecting the hierarchical nature of human activity could not only result in confusion in activity definition and modeling, but also prevent a comprehensive understanding of human activities. Thirdly, with the large population of mobile devices, it is no exaggeration to say that mobile sensing data has become a new member of the Big Data community. We still lack a unified mechanism that can manage and search these data captured along with our daily activities.

In this work, towards a deep understanding of mobile sensing data, we seek for a universal representation for all activities without prior knowledge. The representation should also be able to sufficiently express the activity at multiple resolutions. Based on the similarity measurement of the representation, raw mobile sensing data of different activities can be segmented and categorized automatically. Furthermore, a semantic activity search scheme over mobile sensing data is desired, which can take a raw motion data piece as input, search in mobile sensing database and return a list of ranked data segments that represent the queried activity.

MobiCom'16, October 03-07, 2016, New York City, NY, USA
© 2016 ACM. ISBN 978-1-4503-4226-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2973750.2973752

To develop such practical system, several challenges need to be carefully addressed. First, human activities are very rich, unpredictable, and hard to define and quantify. It is extremely difficult to find a universal semantic representation for arbitrary activities without prior knowledge. Existing hand-crafted feature descriptors cannot fulfil the large spectrum of arbitrary activity space. Second, the hierarchical nature of human activities requires multi-resolution representation. The representation should also adapt to great diversities of mobile sensing data, including temporal difference, individual difference and device difference. Third, the severe time-scale mismatch between query data and searched data raises a big challenge for the activity search strategy design. For example, a typical activity search process can take a short-duration data piece (*e.g.* ten-second data) as input, and search on a long-duration data stream (*e.g.* one-hour data). As activities have a rich set of spatial and temporal scales, there lacks a general segmentation (like the word segmentation for text search) on the mobile sensing data. We need to design both activity representation strategy and search strategy to ensure high accuracy, recall, and efficiency.

To address the above challenges, we propose to extract an elementary basis of diverse motion data, which can span the whole motion data space and capture the discriminative features of all activities. This enables us to embed arbitrary activity using the basis and the interference caused by diversities is eliminated during the embedding. The basis is obtained by unsupervised training with a deep neural network. The multi-resolution receptive fields of different levels in the deep neural network enable us to obtain a hierarchical representation. To manage and search data of different duration, we propose the *activity snapshot* to capture short representative fragments from long-duration data, and an index structure is constructed to accelerate the search process. Our prototype system provides a deep understanding of a rich set of activities, and can achieve accurate and efficient unlabeled activity search over mobile sensing data.

The main contributions of this work are as follows. By exploring the hierarchical nature of human activities, we propose a time-invariant multi-resolution semantic representation for describing arbitrary activities. The representation can be obtained with unsupervised learning. Based on this, we propose the concept of *SBAS* (Semantic Based Activity Search), which expands the boundary of search engine and mobile sensing. We then design a similarity metric distinguishing activities from multi-resolution. With the metric, we propose, design, and implement an innovative system **Lasagna** to manage motion data and search unlabeled activity in a large mobile database. Our implemented prototype is comprehensively evaluated on gigabytes of mobile sensing data collected by us and another well-known dataset. Our extensive evaluation results show that our system can achieve highly accurate activity classification (precision is 98.9%) and search (recall is almost 100% and precision is about 90%) over diverse activities. Besides, our system can be seamlessly concatenated to most of the existing indexing strategies so that response time of search can be benefited by the advanced indexing techniques developed in the literature.

We organize the rest of this paper as follows: Section 2 illustrates the hierarchical nature of human activities and gives an overview of **Lasagna**. Section 3 presents the design of the universal descriptor which can represent arbitrary activities at multi-resolution. Then in Section 4, we explain the semantic analysis and search strategies over large motion database. The prototype system implementation and system evaluation are presented in Section 5. We review related work in Section 6, discuss the promising applications and open issues in Section 7. At last we summarize our work in Section 8.

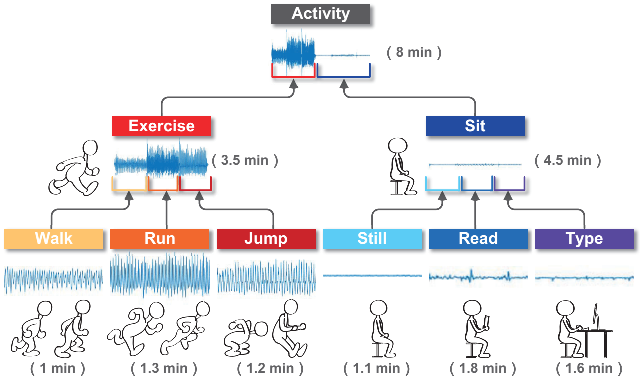


Figure 1: the Hierarchical Nature of Activities

2. SYSTEM OVERVIEW

In this section, we give an overview of this work, including the design space, the design principle as well as the architecture of the proposed system.

2.1 Design Space

Our system, **Lasagna**, is proposed for deep hierarchical understanding over mobile sensing data, just like the traditional Italian food interleaving layers of pasta with layers of sauce. **Lasagna** aims to enable automatic management as well as semantic search over them. We mainly focus on the motion data corresponding to human activities (acceleration and angular velocity) considering its wide applications, which is also more challenging than other types of mobile sensing data since it is complex and unstructured. Our scheme can be easily generalized to handle other data types.

- **Lasagna** can analyze unlabeled raw sensing data collected by different onboard sensors of mobile devices, and embed them into semantically discriminative descriptors at different resolutions.
- **Lasagna** can measure the similarity between semantic descriptors, and automatically categorize data of relevant activities at multiple resolutions.
- **Lasagna** supports semantic activity search over mobile sensing data. The querier can input a piece of raw data of an arbitrary activity, and **Lasagna** will conduct a search over a large database and return a list of ranked data pieces corresponding to the same activity.

The embedding scheme provides a universal compact representation for mobile sensing data at different resolutions, and categorizing data by activities enables effective sensing data management. In this way, **Lasagna** achieves better understanding of human activities, which can benefit users by enriching mobile applications and bringing more accurate and efficient context-aware products into reality. Moreover, **Lasagna** is also a step towards mobile sensing data management system and search engine, which can boost the development of not only mobile industry but also other research and commercial fields, such as medical science, sociology, public security and insurance marketing.

2.2 Design Principle

A careful system design is required to fulfill the aforementioned challenging functionalities. We start with exploring the features of human activities and mobile sensing data, and then discuss the design principles.

Hierarchical nature of activity is an important feature for activity definition and recognition. Fig.1 illustrates a simple example of understanding the same piece of acceleration data collected by a smart watch at different resolutions. As depicted, at the coarsest granularity (the root node), it represents an 8-minute human activity. At a finer granularity, it can be recognized as a 3.5-minute exercise followed by a 4.5-minute sitting. When we achieve deeper understanding, the exercise is composed of walking, running and jumping and while the user is sitting there, he/she first sits still, then reads and types at last. More subtle motions can be discovered when we look at an even finer granularity. The hierarchical nature and other features of human activities require us to design **Lasagna** respecting the following principles:

(a) Multi-Resolution: Our system should comprehensively understand the hierarchical semanteme of activities, which requires our embedding scheme sufficient to express complex motion data at multiple resolutions. Also our search strategy should be capable of matching motion data at different granularities.

(b) Universality: Facing the rich dynamics, unpredictability and the wide-spectrum of human activities,our system should provide a universal solution for understanding, representation and querying arbitrary activities. Lack of prior knowledge and no labeled data greatly increase the challenge.

(c) Adaptivity: Mobile sensing data generated by human activities exhibit great diversities, including spatial-temporal difference, individual difference and device difference, which could cause dissimilarity between motion data of the same activity. Our system should capture the essential features of activities, and both the embedding scheme and searching strategy should adapt to these diversities.Moreover, when performing search, typically searching an unlabeled activity (by inputting a short-duration data, *e.g.* ten-second data) in a long-duration data (*e.g.* one-hour data from multiple sensor readings), the severe time-scale mismatch between query data and searched data also raises a big challenge for the search strategy design.

(d) Efficiency and Scalability: When providing search service, response time is an important factor for good user experience. To deal with large amounts of sensing data, we need to make our search strategy efficient and our system scalable.

2.3 Architecture and Typical Workflow

Considering these principles, we carefully design our system to achieve all aforementioned functionalities. The system architecture and typical workflow is shown in Fig.2. There are two roles for users, one is data provider who collects raw data using his/her smart devices and stores the data in raw database, the other is data querier who inputs a piece of sensing data corresponding to the target activity and query the data pieces of the same activity in the database. Our system consists of three main components, including *model training*, *index construction* and *activity search*, which can be implemented on a PC for personal usage or outsourced to a cloud for public service and minimizing the overhead of clients.

Model Training. We propose to extract elementary bases of diverse motion data, which can be used to embed raw motion data to discriminative hierarchical descriptors. More specifically, this component feed the data in the raw database into the *Motion Basis Extraction* module, which is a well-designed deep neural network with multi-resolution receptive fields at different layers. And then a set of motion bases at multiple resolutions are extracted by unsupervised training. Given a piece of raw data, the *Hierarchical Descriptor Extraction* module generates its hierarchical descriptor using the motion bases.

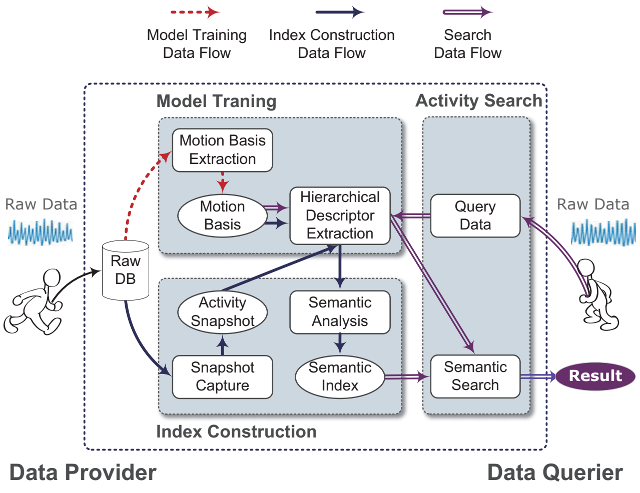


Figure 2: Lasagna Framework

Index Construction. To manage and search data of different durations, we design a *Snapshot Capture* module to capture short representative fragments (Activity Snapshot) from long-duration data. All activity snapshots are embedded to multi-resolution descriptors through the *Hierarchical Descriptor Extraction* module. The *Semantic Analysis* module analyzes the distribution of all snapshots in the descriptor space and constructs semantic index to accelerate search.

Activity Search. This component takes the query data as input, and embeds the raw data into a hierarchical descriptor in the same way. Then the *Semantic Search* module uses the descriptor to perform a nearest neighbor search using the index structure, and outputs a result list of motion data pieces to the querier, where the list is ranked by semantic similarity to the querying activity.

3. HIERARCHICAL SEMANTIC DESCRIPTOR

As mentioned before, we are looking for a universal descriptor which can embed features of arbitrary activities without prior knowledge. Besides, the descriptor should be capable of representing complex activities at multi-resolution and adaptive to diversities of mobile sensing data. Existing work uses statistics (*e.g.*, average and deviation) or more sophisticated descriptors (*e.g.*, DWT [45], autoregressive model [22], and HMM [27]) to describe mobile sensing data, which unfortunately cannot fulfil our requirements.

In linear algebra, a minimum set of vectors that can span the whole vector space is called a basis, *i.e.*, every vector can be represented as a linear combination of the basis vectors. This inspires us to introduce the concept *motion basis* for the motion data space. In this work we propose to extract motion basis, which captures the discriminative semantic features of all activities and can be used to embed arbitrary motion data. The motion basis is obtained by unsu-

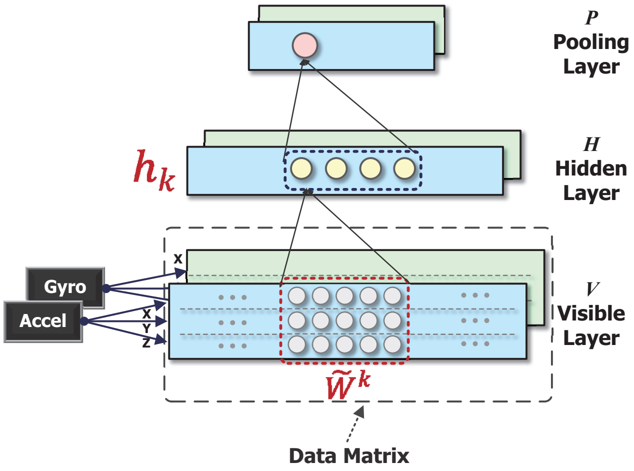


Figure 3: Convolutional Restricted Boltzmann Machine

pervised training of a deep neural network from diverse unlabeled raw data. The network is composed of multiple stacked Convolutional Restricted Boltzmann Machines(CRBM) [17], whose different receptive fields at different levels enable us to extract bases at multiple resolutions. Hence, desired hierarchical descriptors can be generated using motion bases, and semantic similarity between activities can be quantified. In the rest of this section, we present the details of our design.

3.1 Motion Basis Learning

When a person is performing some activity, multiple sensors collect data, maybe in an unsynchronized way or at different sample rates. Before feeding these raw data to the training model, we need to pack them into a coordinated structure, which we refer to as data matrix.

Data Matrix. Given a sensor (*e.g.*, accelerometer, gyroscope, magnetometer and compass), its data can be represented in the format $\mathbf{X} = \{\mathbf{x}_t\}, t = 1 \cdots T$, where $\mathbf{x}_t$ is a vector composed of data from different axes and t is the time index. Data from multiple sensors are organized in a data matrix I in which each $\mathbf{X}$ is regarded as a channel. Given multiple sensors, the size of I is $D_I \times T_I \times C_I$, where D_I is the number of axes of each sensor, T_I is the maximum time index (the total sample number) and C_I stands for the number of sensors. For the example data matrix depicted in Fig.3, $D_I = 3$, $C_I = 2$. We refer to $I^c_{i,j}$ as the j^{th} data sample of the i^{th} axis collected by the c^{th} sensor, $1 \leq i \leq D_I, 1 \leq j \leq T_I, 1 \leq c \leq C_I$. When there are unsynchronized data from different sensors, we align them according to their timestamps. If data from different sensors are sampled at different rates, we interpolate the sparser data to make all the channels aligned.

We pack diverse unlabeled raw data (*e.g.*, accelerometer and gyroscope data of different activities) to data matrixes and feed them to a deep neural network (formed by multiple stacked CRBMs) to extract the motion bases at different resolutions through unsupervised learning.

Convolutional Restricted Boltzmann Machine. As illustrated in Fig.3, the basic building block CRBM is a three-layer architecture, including a visible layer $\boldsymbol{V}$, a hidden layer $\boldsymbol{H}$ and a pooling layer $\boldsymbol{P}$. The visible layer adopts the input data matrix I, whose size is $D_I \times T_I \times C_I$, here $D_I = 3, C_I = 2$. For each channel of the input I, a group of K kernels $\{\widetilde{W}^k\}, k = 1, \cdots, K$, are applied to perform convolution operation. Each $\widetilde{W}^k$ is a $m_{\widetilde{W}} \times n_{\widetilde{W}}$ coefficient matrix. And its size is also referred to as the $receptive\ field$ of the kernel. For the example shown in Fig.3, $m_{\widetilde{W}} = 3$,$n_{\widetilde{W}} = 5$. Considering the accelerometer channel in this example, after one convolution with a kernel $\widetilde{W}^k$, 3×5 units in this channel are mapped to one unit of the corresponding channel in the hidden layer. After a series of convolutions with a kernel $\widetilde{W}^k$ sliding along the time dimension, the accelerometer data in I is mapped to a row vector $\boldsymbol{h}_k$ in the hidden layer, whose length is $T - n_{\widetilde{W}^k} + 1$. The conditional probability of each unit in the hidden layer is computed as

$$P(\boldsymbol{h}_{k,j} = 1|I) = \sigma((\widetilde{W}^k *_v I)_j + b_k),$$
$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

where $*_v$ indicates the valid convolution and b_k indicates the visible-to-hidden bias. The inference value of each $\boldsymbol{h}_{k,j}$ is obtained through a Gibbs Sampling process [5]. That is, we select a random number from a uniform distribution $U[0, 1]$, and set $\boldsymbol{h}_{k,j} = 0$ if the conditional probability is lower than the number, otherwise we set $\boldsymbol{h}_{k,j} = 1$. The pooling layer adopts the pooling kernel with a size of $1 \times n_p$. By uniformly dividing each row $\boldsymbol{h_k}$ in the hidden layer into non-overlapped segments with length of n_p ($n_p = 4$ in Fig.3), the pooling layer shrinks the length of the hidden layer into $\lfloor \frac{T - n_{\widetilde{W}^k} + 1}{n_p} \rfloor$ and each unit equals to the sum of the values in the corresponding segment.

For CRBM, the objective of the training process is to learn a group of equal-size ($m_{\widetilde{W}} \times n_{\widetilde{W}}$) kernels $\{\widetilde{W}^k\}$ that can automatically extract the descriptive information of the data. That is to say, for an input data matrix I, little information is lost after the convolution and the output $\boldsymbol{h}$ can be used to $reconstruct$ the original I. The reconstruction is defined as

$$\boldsymbol{v} = \sum_k (W^k *_f \boldsymbol{h}_k) + a, \tag{2}$$

where $*_f$ stands for the full convolution and a is the scalar hidden-to-visible bias. W^k reverses $\widetilde{W}^k$, which means $W^k_{i,j} = \widetilde{W}^k_{i',j'}$ ($i' = m_{\widetilde{W}} - i + 1, j' = n_{\widetilde{W}} - j + 1$).

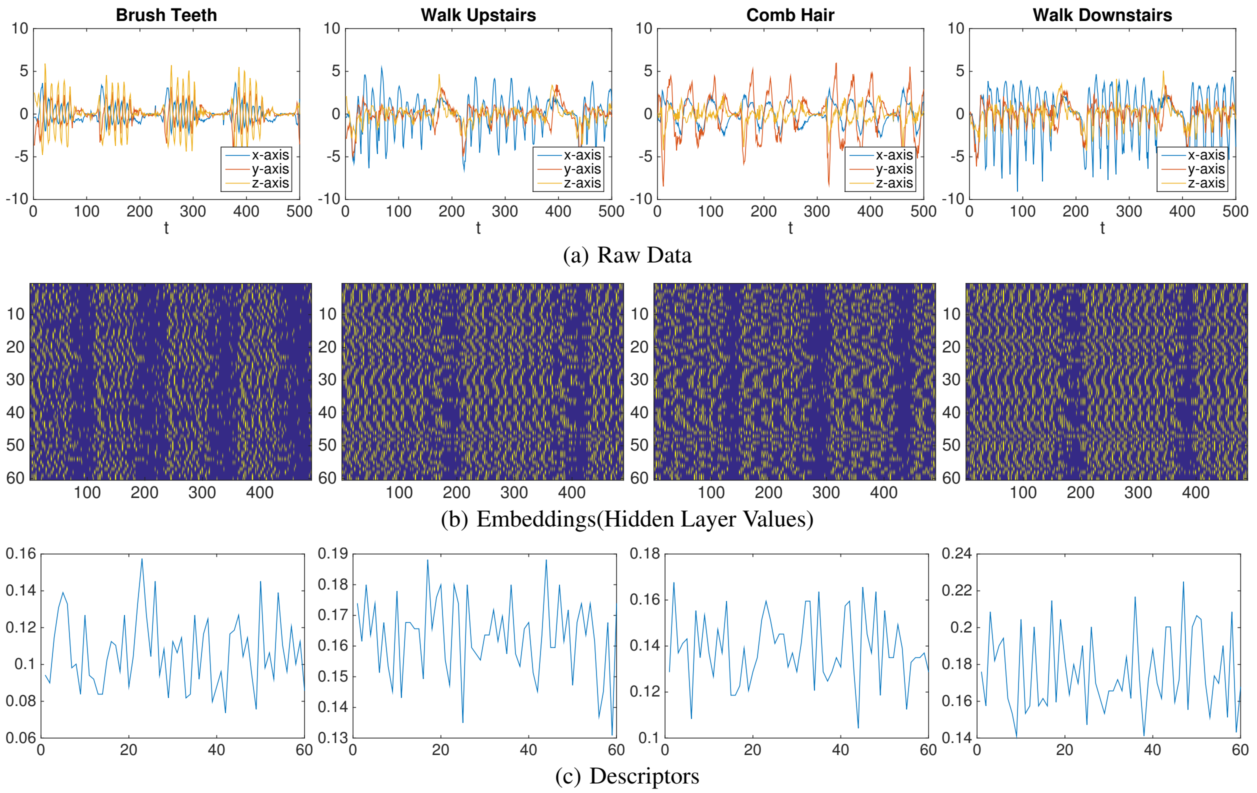
Two metrics, $error$ and $sparsity$ are adopted to evaluate a model. The $error$ is defined as the mean Euclidean distance between all the corresponding rows of $\boldsymbol{v}$ and I, and the $sparsity$ is defined as the mean value of all the conditional probabilities in the hidden layer computed by Eq.1.

Therefore, an unsupervised training can be achieved by minimizing the error. Specially, to avoid getting a trivial solution, a sparsity constraint is added to require that each value in the pooling layer is no more than 1. The training process can be performed by Contrastive Divergence proposed by Lee [18] and Hinton [8].

Moreover, comparing Eq.2 with the typical linear combination of basis in a vector space, we find that, the only difference is that Eq.2 uses full convolution ($*_f$) rather than multiplication. For each channel, the set $\{W^k\}$ can be seen as a basis, and therefore the combination of corresponding $\boldsymbol{h}_k$ can be referred to as an embedding (coordinate) in the space spanned by the basis.

3.2 Descriptor Extraction

After the motion basis is learned, for an input data matrix I in visible layer, the hidden layer outputs a $K \times (T_I - n_{\widetilde{W}} + 1) \times C_I$ data matrix, where K equals the number of convolution kernels we adopt, which is also the embedding of each channel of the input data. Fig.4 illustrates embedding results of four pieces of acceleration data corresponding to different activities (brushing teeth,



(a) Raw Data

(b) Embeddings(Hidden Layer Values)

(c) Descriptors

Figure 4: From Raw Data to Descriptors (Acceleration Channel)

walking upstairs, combing hairs and walking downstairs), with a configuration $K = 60$. Each row of an embedding matrix visualizes an $\boldsymbol{h}_k$ and the j-th light yellow pixel in the row indicates that $\boldsymbol{h}_{k,j} = 1$. These figures intuitively show that the distribution of the light yellow pixels varies for different activities, which means that these embedding results are discriminative in the spanned space of the motion basis.

Therefore, to capture the discriminative feature of different embedding results, for each channel $c (1 \le c \le C_I)$, we propose the descriptor $f_c(I)$, which is defined as:

$$f_c(I)_k = \frac{\sum_j \boldsymbol{h}^c_{k,j}}{T_I}, 1 \le k \le K. \tag{3}$$

The descriptor $f_c(I)$ is a histogram of the light yellow pixels, and the histogram is normalized by the length of the input data matrix.

Time-invariant Property. A typical case in activity search is inputting a short-duration data (*e.g.* 10 second data of running) to search a long-duration data (*e.g.* one hour data of running), the time-scale mismatch between query data and searched data raises critical challenges for search strategy design. By analyzing the descriptor, we find that the descriptor has a remarkable time-invariant property, which can help to solve the time-scale mismatch problem.

Lemma 1. *Given two data matrixes I and I' of a same activity, whose lengths are T_I and $T_{I'} (T_I \neq T_{I'})$. For the their descriptors $f(I)$ and $f(I')$, we have $f(I) \approx f(I')$.*

Based on the definition of our descriptor (Eq.3), it characterizes the distribution of embedding features, which is a set of histograms normalized by the length of the input data (*i.e.* the duration of the activity). As a result, repeating the same activities (with reasonable temporal diversity) won't change the feature distribution. Here we omit the details of the proving process. But we can get the intuition when we let I' be the repetition of I (*e.g.* $T_{I'} = 2T_I, I'_{t+T_I} = I'_t = I_t$). For each k, the numerator and denominator of $f_c(I')_k$ in Eq.3 are both approximately proportional to the times of repetition. Hence we achieve the time-invariant descriptor.

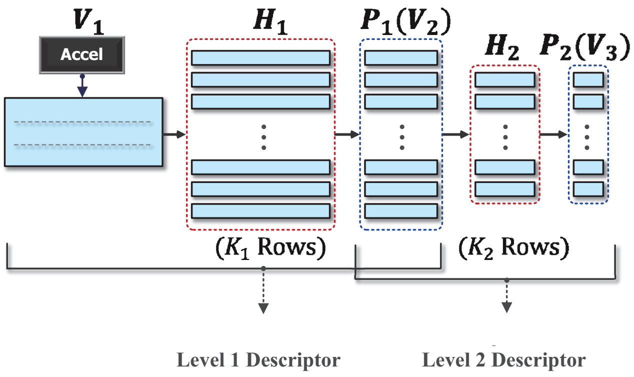
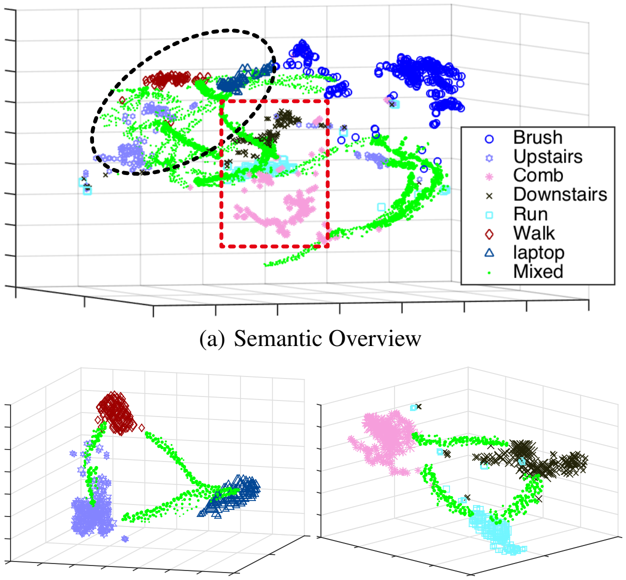


Figure 5: Hierarchical Descriptor Generation

3.3 Hierarchical Descriptor

Until now, we have obtained a descriptor at a specific resolution corresponding to the receptive field of the convolution kernels with a single level of CRBM. To achieve a hierarchical descriptor for describing an activity at multi-resolution, we propose to use a deep neural network (formed by multiple stacked CRBMs),



(a) Semantic Overview

(b) Zoom In

Figure 6: Semantic Analysis

where the higher level of CRBM has a larger receptive field, hence a coarser resolution. Fig.5 illustrates this idea by taking the accelerometer channel as an example. K_1 convolution kernels are applied in the first building block $\mathcal{M}_1$ and we can get the descriptor f^1 (including f_1^1 for accelerometer and f_2^1 for gyroscope). In the pooling layer $\boldsymbol{P}_1$, besides adding the sparsity constraints in the training process, the pooling kernel is more like a filter, that can summarize information of embeddings in the hidden layer. For an input data matrix $I(D_I \times T_I \times C_I)$, while the descriptor is extracted from the embedding, by utilizing pooling kernel with a length of n_p, a new, compact data matrix can be generated with size $K \times \lfloor \frac{(T_I - n_{\widetilde{W}} + 1)}{n_p} \rfloor \times c_I$. Thus, by training another building block $\mathcal{M}_2$ taking all data matrixes outputted from pooling layer $\boldsymbol{P}_1$ as input, a second-level motion basis containing K_2 components can be learned. After that, we treat the pooling layer $\boldsymbol{P}_1$ as the visible layer $\boldsymbol{V_2}$ of a new building block $\mathcal{M}_2$, and then a second-level descriptor f^2 can be extracted. In this way, f^2 describes the motion characteristics at a coarser level, since the pooling operation in $\mathcal{M}_1$ enables the convolution kernels in $\mathcal{M}_2$ to analyze motion data at a larger scale. Accordingly, for motion data I, the descriptor is defined as $f(I) = \{f_c^l(I)\}$, where $l = 1, 2, \cdots$ indicates the level and $c = 1, 2$ indicates the accelerometer channel and gyroscope channel. More building blocks $\mathcal{M}_3, \mathcal{M}_4$... added to the structure, more semantic layers of the motion can be extracted, which allows us to understand the motion comprehensively.

3.4 Semantic Similarity

Based on the hierarchical descriptor, the semantic similarity between descriptors can also be decomposed into multiple levels. For two descriptors $f(I_1) = \{f_c^l(I_1)\}$ and $f(I_2) = \{f_c^l(I_2)\}$:

First of all, similarity $\theta_{l,c}(f_c^l(I_1), f_c^l(I_2))$ in the same level l and channel c can be measured by commonly used vector based metrics. Furthermore, we can fuse the knowledge from different channels (sensors) and get the multi-channel similarity Θ_l $(l = 1, 2, 3, \ldots)$. And similarly, multi-level similarity Θ_c $(c = 1, 2)$ and comprehensive similarity Θ can also be obtained.

For simplicity, in this work, we use the fusing strategy of concatenating the descriptors from different sensors and different levels. Euclidean distance between the descriptors is adopted in getting $\theta_{l,c}$ (as well as Θ_l, Θ_c and Θ). Evidently, a larger θ (or Θ) indicates a smaller semantic similarity between two descriptors.

4. SEMANTIC ANALYSIS & SEARCH

Based on the proposed multi-level model, the hierarchical descriptor we propose can automatically highlight the descriptive information hidden in the motion data, and the similarity between activities can also be quantified at different resolutions. However, facing with the raw database, we are still far from activity management and search based on the proposed descriptor. The raw database collects the motion data streams with long time duration, and different activities are concatenated together. It is intractable to get different activities separated for further comparison.

To address these issues, we propose a semantic analysis strategy for raw motion data. By taking snapshots on large pieces of data, the activity semanteme can be extracted and purified. Moreover, after indexing the extracted activity semanteme, accurate and efficient semantic based activity management and search can be realized.

4.1 Activity Snapshot

It is difficult to segment the motion data streams to separated activities due to two major challenges. First, human activities possess a hierarchical nature, there are varying ways of defining activities at different resolutions (*e.g.* from the basic hand swinging to running and then basketball playing). Second, the duration of different activities varies from person to person, and from time to time, which prevents us from segmenting them with equal-length time window.

To address these problems, we propose a new activity *snapshot* data structure to capture the activity fragments in a data stream at different granularities.

Snapshots are captured according to the parameters *focus position* p and *scale* q. For a data matrix I whose length is T_I, the snapshot captured under (p, q) collects the data from $p - q$ to $p + q$ ($p - q > 0, p + q \leq T_I$). Thus, we continuously slide the focus position p along the time axis with a step length, and at each position, multiple snapshots are captured by adjusting the scale q.

To avoid unnecessary enumeration and reduce the computation overhead, we propose the minimal resolution r_{min} as the step length and the maximal resolution r_{max} to limit the max size of a snapshot ($2q < r_{max}$).

4.2 Semantic Analysis

When a new query is uploaded, by performing NN-Search within the snapshots, the probable results that are semantically similar to the query can be obtained. However, trivially performing linear comparison through the snapshots will inevitably bring unacceptable computation overhead. Thus, it is urgent to perform an analysis on these snapshots.

Fig.6 visualizes the distribution of snapshots' descriptors in a 3D space. Each descriptor, in other words, a point in the figure corresponds to a snapshot. We divide the snapshots into two categories: $unitary$ snapshots and $mixed$ snapshots. The former refers to snapshots that capture a unitary activity, while the latter refers to those span multiple activities. In this example, the groundtruth of these activities is obtained by labelling the training data manually.

As illustrated in Fig.6(a), the snapshots are well aggregated in several clusters. When we zoom in, in Fig.6(b), we observe that the distribution of snapshots exhibits very good properties which can significantly facilitate our further analysis. On the one hand, snapshots gathering together tend to be unitary snapshots of the same activity. On the other hand, mixed snapshots lie between clusters. And the distances from a mixed snapshot's descriptor to its adjacent cluster heads are relevant to the weights of activities it spans. Based on the first observation, we can extract unitary activities in an unsupervised manner. Then according to the second observation, we can approximate a mixed snapshot's constitution by representing its descriptor through a linear combination of several nearby cluster heads.

4.3 Semantic based Activity Search

As more and more new users join the system who continuously generate data, the scale of motion database could be huge which raise critical challenge to activity search efficiency. The result of semantic analysis can help to provide efficient and accurate Semantic Based Activity Search (SBAS). In order to speed up the activity search, we propose a clustering-based indexing scheme. We firstly adopt the density based algorithm DBSCAN to cluster the snapshots in database. Then we can extract a series of cluster heads as representative descriptors which are used to set up an indexing structure. Note that our approach is fully compatible with other existing indexing approaches.

After the analysis, suppose that we have already built a representative descriptor set $\mathcal{S}$. And all snapshots are mapped to their closest representative descriptors, which means they are semantically similar. Then when the query q is uploaded, the SBAS can be performed in a progressive procedure:

(1) Semantic Level Search: Search for a subset of representative descriptors $\mathcal{S}'$ that are most semantically similar to the descriptor of the query q.

(2) Snapshot Level Search: Search for the most relevant snapshots among those are mapped to $\mathcal{S}'$.

(3) Snapshot Splicing (if possible): Rank the snapshots retrieved in procedure (2) and splice the snapshots that have an intersection in time duration.

By dividing the search process into different steps, the progressive strategy can avoid the comparison between distant snapshots and greatly reduce the computation overhead.

In SBAS, if the querier input query data with labels, the semantic mapping between text labels and motion data can be gradually set up. Based on our semantic analysis, we can achieve automatic labeling over the raw motion database and therefore a text-based activity data search could be enabled.

5. IMPLEMENTATION AND EVALUATION

We design and implement a proof-of-concept prototype **Lasagna**, to manage and search mobile sensing data of arbitrary activities. The system consists of an application which collects data from various sensors in mobile devices, and a service whose architecture is illustrated in Fig.2. Here we present the system implementation and measurement to show the feasibility and practicability of our system.

5.1 System Implementation

Software implementation: We develop two versions of sensing data collecting application. One is implemented in Java for Android platform, the other is implemented in HTML and Javascript for Tizen. We implement all components (Fig.2) of the service with Python and Matlab. For the deep neural network, we modify the CRBM proposed by Lee et al. [17] and construct the network with multiple CRBMs.

Hardware configuration: Two kinds of smart watches are used for data collection, one is Samsung Galaxy Gear S (with Tizen OS) and the other is Sony SmartWatch3 (with Android Wear OS). The service is setup on a PC with a 2.5GHz Intel® Core™ i7 CPU and a 16GB 1600MHz DDR3 memory.

5.2 Data Collection and Datasets

To comprehensively evaluate our system, we collect our own dataset and also adopt a public dataset [33] for evaluation and comparison. The details of two datasets are as follows:

Dataset[#1]: This dataset is collected by us. 10 volunteers (7 males, 3 females) wear smart watches on right wrist. 320 hour (2.7 GB) data of accelerometer and gyroscope are collected in both controlled and uncontrolled environments. Data are manually labeled for ground truth.

Dataset[#1][Controlled] includes 11 common activities: getting up, lying down, brushing teeth, combing hair, drinking water, pouring water, walking, running, walking upstairs, walking downstairs and typing. Each activity is performed by our volunteers and the time duration varies from 10 seconds to 10 minutes.

Dataset[#1][Uncontrolled] includes but not limited to the 11 activities in the controlled environment. Volunteers wear smart watches for whole days and act freely, while data are continuously collected. We design buttons for pre-defined activities for users to label the current data by a simple tap. And also an "add a new activity" button is used for volunteers to add a new activity. To reduce the effect caused by inaccurate labeling, *e.g.* the delayed label caused by response time, we remove the start and end of the data of each activity.

To minimize power consumption caused by continuous data collecting and evaluate the accuracy of our method with a low sampling rate, we set the sampling rate to 20Hz. Our app periodically writes the motion sensors' readings to the smart watch's local storage. The stored data will be uploaded to the server when WiFi or USB connection is available.

Dataset[#2]: This dataset is collected by Shoaib et al. [33]. 10 males with Samsung Galaxy SII attached on the right wrist. 323.9 MB data of accelerometer and gyroscope are collected in controlled environment, including 7 activities: walking, standing, jogging, sitting, biking, walking upstairs and walking downstairs. Each activity is performed for 3-4 minutes.

5.3 Model Training

As presented in [18], a deep neural network can be efficiently trained using greedy level-wise training. For our multi-level CRBM model, we pack all the controlled and uncontrolled raw data of Dataset[#1] without labels into data matrixes as the input, which has three rows (for three axes) in each of the two channels (for accelerometer and gyroscope). Then we conduct a greedy level-wise training to minimize reconstruction error under the sparsity constraint without supervision. Here the two metrics *error* and *sparsity* (defined in Section 3.1) can evaluate the performance of a deep learning model and the embedding. Smaller error indicates higher embedding accuracy, and better sparsity (sparser model) indicates more efficient descriptor and also avoids trivial solutions. In state-of-the-art deep learning systems, tuning model parameters is an manual process, which is directed mostly by experience. In our experiment, to achieve optimal model, we tune all model parameters (including convolution kernel number at each level, kernel width, etc.) and explore more than 50 different configurations of these parameters.

It is worth noting that kernel number and level number affect different aspects in our model. Kernel number affects the accuracy

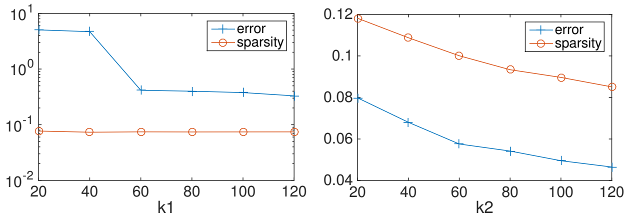


Figure 7: Kernel Number Selection

of the embedding, while level number mainly affects the coarseness of the extracted activity features (*i.e.* the semantic level of activity). So we determine these two parameters separately in our work.

Kernel Number.

Kernel number at each level is an important parameter for our deep neural network, which determines the size of the descriptor, and also affects the error and sparsity of the model. There is a trade-off in determining the kernel number. On one hand, a larger kernel number raises the embedding accuracy, since more information can be represented by more kernels. As shown in Fig.7, the error reduces as K_1 and K_2 increase. On the other hand, a larger number of kernels will bring longer representation (feature vectors), which will bring extra cost for storage, model training and searching. To achieve a balance, we select $K_1 = 60$ and $K_2 = 60$ after which the declination of error slows down significantly. Under this configuration, we can have a classification accuracy of 98.2% (Table 2), which we think sufficient for accurate activity search.

Moreover, as we can see in Fig.7, K_1 and K_2 influence the performance (*error* and *sparsity*) differently. This is because the input of level 2 is the output of level 1. As we designed, the motion basis helps to filter out the diversities and interferences. After the convolution and max-pooling operations in level 1, the input of level 2 is smoother than that of level 1, which leads to smaller errors using the learned motion basis in level 2.

Kernel Width.

Kernel width $n_{\widetilde{W}}$ is another important parameter determining the receptive field of each level. When we extract a multi-resolution descriptor for an activity, kernel width indicates the granularity of each hierarchy. In our experiment, facing diverse unknown motion data, we don't want to miss any important feature of activities, so we set the kernel width to a small number, $n_{\widetilde{W}} = 10$, which is approximate to 0.5-second resolution at the first level. The receptive field increases level by level due to the pooling operation, thus produces coarser and coarser descriptor.

Level Number.

The multi-level CRBM provides us hierarchical view of activity features. When stacking more levels of CRBMs, more comprehensive information can be obtained in the multi-resolution descriptor, and more accurate the descriptor will be. As we will present in the upcoming accuracy evaluation (Table 2), the classification accuracies for one-level model, two-level model and three-level model are 97.8%, 98.2% and 98.9% respectively. However, increasing the model level will cause the following impacts: (1) higher training cost is needed to train more CRBM; (2) CRBM in higher level is fed with the data shrunk by the pooling operation in the lower level. Hence, to guarantee the model quality, severalfold raw data is required for training a high level model. (3) more levels in the model mean more dimensions of the hierarchical descriptor, which also increase the complexity for similarity comparison and indexing construction. As a result, the design goal of our deep learning model is to achieve high classification accuracy with as less level as possible. Since the improvement is not significant when we use a 3-level model, to achieve a trade-off, we adopt the two-level model in the rest of the experiments. But a 3-level model is also trained to compare with the two-level model to study the effect of level number.

5.4 Effectiveness of Semantic Descriptor

5.4.1 *Activity Classification Accuracy*

After training the multi-level CRBM model, we extract the motion basis of each level and generate hierarchical descriptors of motion data using these bases. Considering there is no existing work addressing the semantic based activity search over unlabeled motion data, the most related area is activity classification. To objectively evaluate the accuracy and discriminative property of our descriptor, we compare our work with two related works [33] and [11] for the same activity classification task using the same Dataset[#2].

We choose them for comparison because they are both recent highly relative and representative works on mobile device based activity classification, which achieve remarkable accuracy. Shoaib et al. propose a classic strategy with a rich set of predefined features (both in time-domain and frequency-domain) [33]. A set of predefined features (in Table 1) are extracted as the descriptor for data segments. Then classifiers like SVM, are trained with 90% labelled data and the accuracy is tested using the rest 10% data. The work of Wenchao and Zhaozheng, which is published in ACM MultiMedia 2015, adopts the up-to-date supervised Convolutional Neural Network. They train the CNN with data segments and their transformations (wavelet decomposition and discrete Fourier Transformation) in a supervised manner. Then 21 (C_7^2) two-class SVMs are trained based on the output of the CNN to determine the final classification.

For the same task, our model is trained using all the raw data in Dataset[#1] as presented in Subsection 5.3 in an unsupervised manner. With our model, hierarchical descriptors of data segments in Dataset[#2] are extracted for classification. For a fair play, we adopt the same SVM in [33] to conduct classification for 7 activities. Table 2 presents the average classification accuracies of two related works and our method. Here, x-level indicates the classification accuracies using our hierarchical descriptor extracted from the x-level CRBM model. According to the results, our descriptor outperforms the predefined features in [33] greatly. Compared with the 98.75% accuracy of the supervised deep learning model [11], our 2-level descriptor has a comparable accuracy (98.2%) and our 3-level descriptor has an even a higher accuracy (98.9%). The evaluation reveals another fact that, accelerometer has a better performance on activity recognition than gyroscope, and compound data from multiple sensors can increase the accuracy.

We note that, Dataset[#1] and Dataset[#2] consist of different activities performed by different groups of people with different devices. Our model is trained using Dataset[#1] and evaluated on Dataset[#2], and achieves a significantly high accuracy (98.9%), which show that our hierarchical descriptor can capture the discriminative features of arbitrary activities without supervision, and has good universality and adaptivity.

Table 1: Feature Sets

Feature Set	Features in the Feature Set
FS1	Mean, standard deviation
FS2	Median, zero crossings, root means square
FS3	Variance, zero crossings, root means square
FS4	Sum of first five FFT coefficients, spectral energy

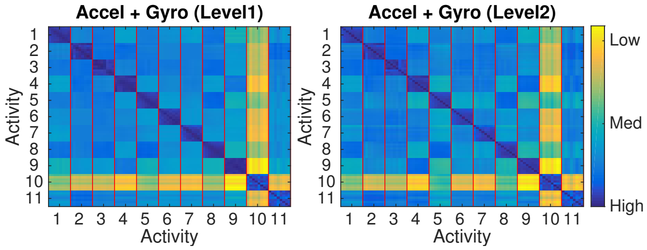


Figure 8: Fusing Knowledge from Accel and Gyro

Table 2: Classification Accuracy

Sensor	[33]	[11]	1-level	2-level	3-level
Accel	80.3%	-	94.6%	96.1%	98.4%
Gyro	71.8%	-	82.1%	82.9%	91.4%
Acc+Gyro	90.3%	98.75%	97.8%	98.2%	98.9%

Uncontrolled Data. To further explore the effectiveness of our descriptor, we perform a classification task on our uncontrolled Dataset[#1][Uncontrolled], which is much more challenging due to more noise and unknown actions compared to the controlled data. Same model and strategy as that are used for evaluating the accuracy on Dataset[#2] are adopted to distinguish 11 different activities, and the results are presented in Table 3. We can see that even for uncontrolled daily activities, an accuracy over 90% can still be guaranteed using a low level model (1-level or 2-level). With a higher level, the model accuracy can be further improved.

People Diversity. Here, we evaluate the adaptivity of our descriptor to people diversities. There are 10 different males' data in Dataset[#2]. We perform the classification task on different number of people's data. As shown in Table 4, the accuracy has a slight decrease when the people number increases. Overall, our descriptor still shows a good classification accuracy (above 97.8%) for diverse people's motion data.

Table 3: Classification Accuracy (Dataset[#1][Uncontrolled])

Sensor	1-level	2-level
Accel	88.1%	89.2%
Gyro	80.3%	80.9%
Acc+Gyro	91.0%	92.0%

Table 4: People Diversity

#People	2	4	6	8	10
Accuracy	98.54%	98.16%	97.95%	97.87%	97.80%

5.4.2 Time-invariant Property

In Subsection 3.2, we mentioned that our descriptor has a time-invariant property, which can match motion data of the same activity at different time-scales. Here we validate the time-invariant property by experiment. We randomly cut out 8 data segments for each activity in Dataset[#1] with incremental time duration (5s, 10s, 15s,..., 40s), and extract the Level 1 and Level 2 descriptors of each segment. Similarities are calculated among different activities' descriptors, and also among descriptors of the same activity with different time duration. Fig.9 shows all similarities, with all 11 activities lined up along both axes and each activity has 8 strips for 8 segments of different duration, forming a 88 × 88 grid. The color of each cell of the grid indicates the similarity between corresponding segments. The darker color implies higher similarity, while the lighter color implies lower similarity

As depicted in Fig.9(a), since segments of the same activity are grouped adjacently along both axes, the dark 8 × 8 blocks along the diagonal line confirm that descriptors of the same activity are highly similar to each other, even though they are representing data at different time-scales, *i.e.* our descriptor is time-invariant. Besides, the bright (low similarity) non-diagonal cells show that our descriptors are sufficient to distinguish different activities.

Activities at multiple resolutions. Furthermore, Fig.9(b) illustrates the similarities measured with only Level 2 descriptors (not the 2-level hierarchical descriptors, but the descriptors extracted only from the second level of CRBM model). We notice that, the overall color discrimination is not as clear as Fig.9(a) (using Level 1 descriptors). This fact is caused by the hierarchical semanteme of human activities and our descriptor. The Level 2 descriptor captures much coarser semantic features of motion than the Level 1 descriptor. And at this resolution, the coarse feature is not sufficient to distinguish semblable activities like walking, running, walking upstairs and walking downstairs. But it better distinguishes the 1st activity (brushing teeth, the light blue one) and 10th activity (typing, the light yellow one) from all other activities, since they are hand movements and others are more like body movements.

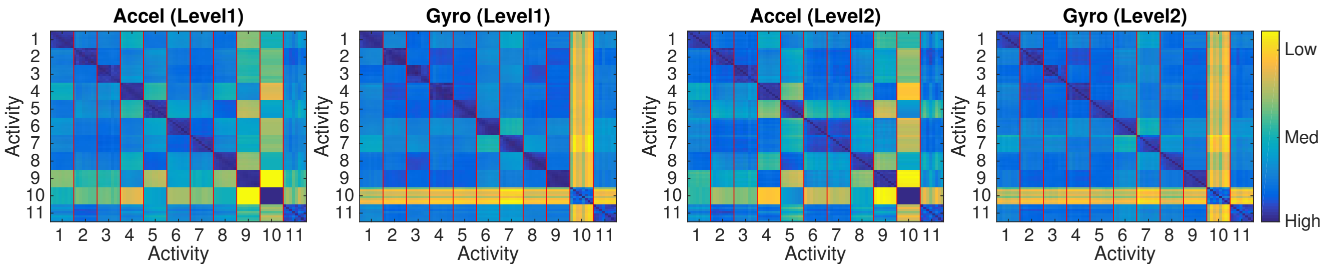
Descriptor with different sensors. Comparing the subfigures of Fig.9(a) and Fig.9(b), we find that descriptors extracted from accelerometer and gyroscope contribute different in activities recognition at different semantic level. At a higher level, the data of gyroscope shows better discrimination, so a descriptor compounding information from multiple sensors can significantly enhance the classification accuracy, as shown in Fig.8 and Table 2.

5.5 Semantic Analysis and SBAS

Fig.6(a) and Fig.6(b) have demonstrated that semantic analysis can be achieved by taking snapshots of motion data streams. In this part, we will further measure the performance of semantic analysis and semantic based activity search (SBAS) in both controlled and uncontrolled environments. In the experiments, we use 2-level descriptors of compound data from both accelerometer and gyroscope.

Semantic Analysis. Note that, in semantic analysis we have *unitary* snapshot (capturing a single activity) and *mixed* snapshot (capturing mixed activities). The raw data in the uncontrolled environment contains mixed activities naturally. For the controlled environment, we randomly select segments from 11 different activities and manually splice them to get a 2 hour mixed motion data. Therefore, snapshots are captured respectively on the uncontrolled data and spliced controlled data, and then clustered using the density based DBSCAN algorithm. We propose a metric named *aggregative degree* to evaluate the performance of the semantic analysis, which is defined as for a certain activity, the proportion of the unitary snapshots that are clustered into a same class.

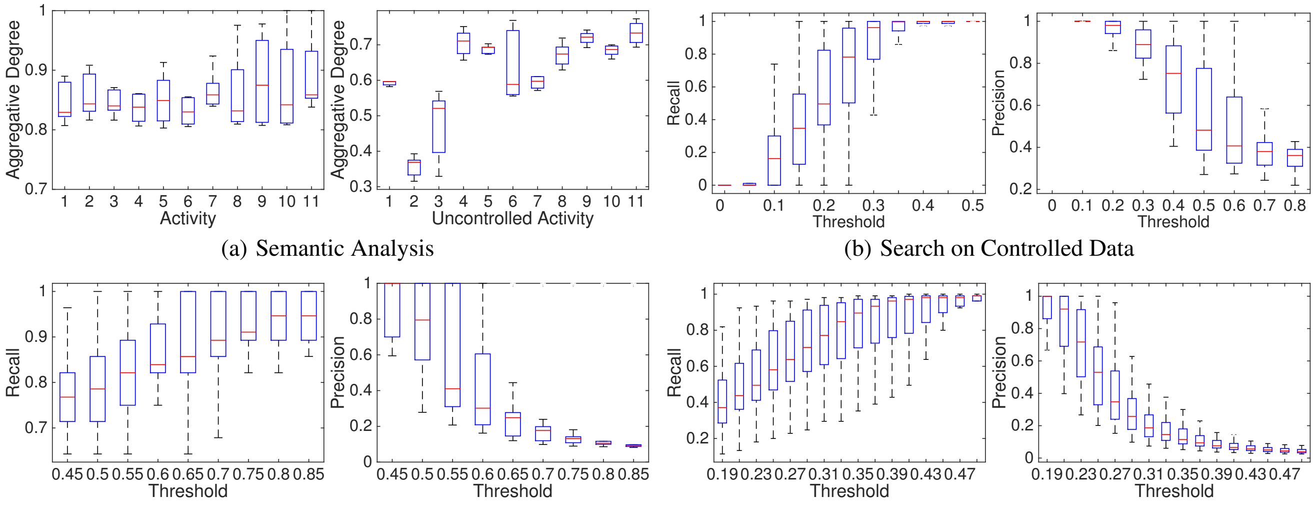
The measurement results are depicted in Fig.10(a). For controlled data, the minimal aggregative degree of all the activities is over 0.8, which means for each activity, over 80% unitary snapshots are mapped into a same cluster. For uncontrolled data, we have a decreased aggregative degree for each activity. The reason is two-fold: (1) the activities in uncontrolled environment are much more complex, which results in ambiguous clustering; (2) mislabeling of the motion data in uncontrolled environment may cause inaccurate ground truth. But for most of the activities, the aggregative degrees still stay around 0.7.



(a) Level 1 Descriptor

(b) Level 2 Descriptor

Figure 9: Similarities Between Different Activities with Different Time Duration



(a) Semantic Analysis

(b) Search on Controlled Data

(c) Search on Uncontrolled Data

(d) Search for Mixed Activity

Figure 10: Semantic Analysis and Search Accuracy.

SBAS. Then we perform SBAS based on the result of semantic analysis. During the progressive search procedure, a distance threshold is used to filter out the "distant" representative descriptors and the "distant" snapshots. We employ two metrics *precision* and *recall* to evaluation the data retrieval performance of SBAS. Taking the activity label as a groundtruth, precision is the fraction of the retrieved data pieces that are correct, while recall is the fraction of the correct data pieces that are successfully retrieved. Both factions are measured by time duration, and the larger the better.

First, we evaluate SBAS over spliced controlled data. As shown in Fig.10(b), a larger distance threshold yields higher recall and lower precision. So a proper threshold should be chosen to balance recall and precision. For the controlled data, a 0.8 recall (80% of the correct data pieces are successfully retrieved) and a 0.9 precision (90% of the retrieved data pieces are correct) can be achieved when the threshold is set to 0.3. When the threshold is tuned to 0.4, we can almost retrieve all correct results, while the precision is still above 0.7. Fig.10(c) shows the performance of SBAS for uncontrolled data. When we set the threshold to 0.5, we get a 0.78 recall and 0.8 precision. In the uncontrolled environment, due to the complex human motion and mislabeling, it is hard to retrieve all correct data even with a large distance threshold.

Then, we make an attempt to search mixed activity, for example brush-comb (brushing teeth and combing hair), type-drink (typing and drinking water), etc. The search is conducted on the controlled data and the result is presented in Fig.10(d). We find that when the threshold is 0.25, the recall is around 0.6 and the precision falls below 0.6. Comparing with Fig.10(b), we notice a sharp decline of the precision as the threshold increase, *i.e.*, many incorrect data pieces are retrieved. The reason could be that the mixed activity contains features of multiple activities and is located between activities in the descriptor space (Figure.6(b)). So its component activities can be easily taken as the correct results. We will further explore mixed activities in our future work.

Table 5: Time Overhead

Data Size	1 min	10 min	1 h	1 d	10 d
Indexing Time (s)	0.001	0.02	0.55	7.89	71.63
Search Time (s)	0.0008	0.002	0.052	0.28	8.83

At last, we discuss the practicability of **Lasagna** including energy consumption, efficiency and scalability.

Energy Consumption. According to our evaluation, keeping running data collecting application at backstage only leads to about 10% additional power consumption, which is affordable for most COTS mobile devices. Moreover, since it is more energy efficient to launch a system service than an application, this part of consumption could be further reduced if data collecting service can be attached in the smart watch operating system.

Efficiency and Scalability. We consider the procedures in performing SBAS, including feature extraction, index setup and semantic search. First, for each input data matrix, feature extraction consists of data inference and quantification. As we can see in Eq.1 and Eq.3, both parts are linearly correlated with the length of the input data matrix. Then for the index setup, the server performs a density-based clustering algorithm over the feature vectors. In the end, the server performs a search on the index and returns the result. For a querier, it brings negligible overhead to perform feature extraction over the query data matrix. For the server, as shown in Table 5, it takes 7.89 seconds to index one-day sensing data, and for each search query, it takes only 0.28 seconds. For ten-day data, the index and query time are 71.63 seconds and 8.83 seconds respectively. Considering the evaluation is performed on a PC, the runtime can be greatly reduced using a much powerful sever. Moreover, there is a plenty of work devoted to improving the performance of data indexing and retrieving based on feature vector. For boosting each process, our system is compatible with most of the vector based indexing and search strategies.

6. RELATED WORK

To the best of our knowledge, there are few approaches addressing the problem of semantic based activity search over unlabeled motion data. **Lasagna** is related to existing work is in the following areas.

6.1 Recognition on Mobile Sensing Data

The recognition strategies on mobile sensing data can be divided into two categories.

(1) *Physical Model Based Methods*: Those methods construct a physical model of an event, and then explore the correlation between the physical model and patterns of mobile sensing data. For example, Fang-jing et al. design *cyber-physical handshake* [46], which allows two users to naturally exchange personal information with each other after detecting and authenticating the handshaking patterns between them. Abhinav et al. tackle the problem of recognizing smoking behavior using a wristband equipped with a 9-axis inertial sensor [26]. Lan Zhang et al. integrate the temporal and spatial constraints while walking and achieve meter-second-level tracking with COTS smartphones [48]. Nirupam Roy et al. propose *WalkCompass*, a system that estimates the walking direction by analyzing the relationship between human walking and its effect on the phone [30]. Yanzhi Ren et al. propose a user verification scheme leveraging gait patterns derived from acceleration readings to mitigate against user spoofing attacks [29].

Moreover, leveraging the acoustic signal, Zheng Sun et al. exploit the relationship between the pointing gesture and the Doppler effect, and propose *Spartacus*, which enables spatially-aware interaction for mobile devices [37]. Similarly, by waving a hand from one device towards another, users can directly transfer files between them using *AirLink* [3].

(2) *Machine Learning Based Methods*: Machine learning are widely applied in achieving activity recognition and classification. Frequently used classifiers can be divided into two categories, supervised classifiers (*e.g.* KNN, SVM, Decision Tree, etc.) and unsupervised ones (*e.g.* K-Means, Markov Model, etc.). Generally speaking, supervised strategies can provide better accuracy while unsupervised strategies are more computationally efficient and do not require labeled data.

Based on the user-annotated acceleration data, Ling Bao et al. from MIT Media lab propose to extract a set of frequency domain features for 20 daily activities, and the trained decision tree classifiers provide an overall accuracy of 84% [2]. Kwapisz et al. use phone-based accelerometers and collect labeled data of 6 activities from 29 users. The classification accuracy ranges from 77.6% to 96.9% [13]. Anjum et al. collect the readings of accelerometer and gyroscope and perform classification over 9 activities using 4 different classifiers. The average accuracy ranges from 73.8% to 95% [1]. Through supervised training, Heng-Tze Cheng builds a semantic attribute structure. The experimental results show that the proposed approach achieves 70-80% precision and recall in recognizing unseen new activities [4]. Shoaib et al. extensively evaluate the activity recognition performance with four motion sensors and four feature sets. Seven physical activities are targeted and the adopted supervised classifiers include SVM, KNN, Decision tree, and so on [33].

On the other hand, there are also unsupervised strategies devoted to address the problem of activity recognition. Yongjin Kwon et al. adopt a set of time-frequency domain features. The unsupervised strategies (*e.g.* mixture of Gaussian, DBSCAN etc.) can achieve around 90% accuracy when the number of activities is unknown [14]. Tâm Huynh et al. propose to recognize daily routines as a probabilistic combination of activity patterns. The mean recognition precision of the unsupervised strategy is around 77% [10]. Peng Wang et al. propose pattern-based Hidden Markov Model (pHMM) that can learns the patterns and the model simultaneously from the time series data [42]. Yasuko Matsubara et al. present *AutoPlait*, which can automatically identify all distinct patterns in a time-series using Hidden Markov Model(HMM), and spot the time-position of each variation [23]. Besides traditional recognition/classification tasks, machine learning based methods can also be used for device unlocking [6, 9, 20, 38], invalid user detection [7,32,50], keystroke eavesdropping [41], keyword recognition with inertial sensor [49], speech recognition [21, 24, 25, 47], GPA forecasting [43], non-infrastructure localization [39], and so on.

All the existing work have made great use of the mobile sensing data. However, the physical model is only applicable to one or a few predefined specific activities. Supervised classifiers need additional labeled data for supervised training. Existing unsupervised strategies can only deal with the activity that is seen in the training data. The rich dynamics and the wide-spectrum of human activities make it quite laboursome to explore different activities one after another. And it is also inefficient to apply all recognition models on a piece of unknown data. Besides, those methods only focus on a specific granularity of an activity, neglecting the hierarchical nature of human activity.

Different from existing work, we propose a universal hierarchical descriptor for arbitrary activities without requiring prior knowledge. Based on this descriptor, we design an innovative system supporting managing and semantic search on a rich set of motion data.

6.2 Deep Learning Based Recognition

Recent years, many efforts have been devoted to training deep models for recognition tasks, and most of them focus on image and video understanding.

Facenet [31] learns a model that can embed a face image into the Euclidean space. Similarity between two faces can be measured without external classifiers. DeepID [34–36] is a series of work focusing on face recognition. By analyzing face images at multiple resolutions and increasing the depth of the network, the recognition accuracy reaches up to 99 percent. Besides, using the semantic and temporal constraints between video frames, [28] [44] study the embedding of unlabeled video frames. Text descriptors for images can also be automatic generated with deep learning [12] [40].

As data streams from multiple heterogeneous sensors, mobile sensing data possess quite different properties from image data. And the richness, dynamic and diversity of human activities increase the challenge of understanding arbitrary motion data. There is a few work dealing with mobile sensing data recognition using deep learning. DeepEar [16] uses labeled acoustic dataset to train a deep neural network to perform ambient scene classification, stress detection, emotion recognition and speaker identification, etc. Wenchao trains a convolution neural network to get the probability distribution of different activities, and an additional group of SVMs are trained to determine the final activity classification [11]. Those two methods achieve good recognition and classification accuracy. However, they rely on the supervised training with a large set of labeled data and also they neglect the hierarchical property of the data.

7. DISCUSSION

7.1 Promising Applications

Lasagna enables hierarchical understanding and efficient management over mobile sensing data, which could bring a lot of promising applications in the future. For individual, an diary-like record of daily activities can be generated automatically. Rich statistics can be provided by the automatic recording, which can help people have better understanding of their daily lives, or inform them when abnormal activities or bad habits are detected. In such a way, better time and health management can be achieved. For mobile service providers, through a deep and long term understanding of human activities, **Lasagna** can help them improve the behavior targeting advertising since users' real-time behaviors are available. Also, mobile products design can be improved with the real-time and statistical data. More importantly, our system provides SBAS in massive mobile sensing data, which could bring a whole set of new ways to explore and make use of mobile sensing data. For example, government and organizations can use SBAS to study the residents' health condition, the correlation between common diseases and people's exercise habits, etc. Besides, **Lasagna** can facilitate the context-aware computing functionalities. For example, operating system can automatically adjust the system setting to conduct context-aware energy saving.

7.2 Open Issues

In this work, we propose and implement a proof-of-concept system, which provides semantic based activity search on mobile sensing data. But there are still a few steps to take before SBAS is ready for practical use by massive users. First, issues in traditional text based search engine also need to be addressed for SBAS. For example, at the server side, efficient activity segmentation and indexing need to be carefully designed. Compared to text processing, performing reasonable segmentation over continuous motion data without activity vocabulary and then construct efficient index are even more challenging. Second, SBAS should be able to support more complex queries. **Lasagna** works well in searching with single "keyword" (*i.e.* activity), while designing strategies that can accept queries like multiple keywords or regular expressions is challenging and necessary. Moreover, since people may also need to perform semantic activity search by text or pictures, how to associate text (or picture) semanteme with motion data to support cross-modality search also need careful design. At last, a comprehensive understanding of mobile sensing data and the search function may lead to privacy concerns. The service providers may be curious about user privacy and there could even be malicious attackers. New privacy protection mechanism is required when the mobile sensing data is better understood and searched.

8. CONCLUSION

In this work, we address the issue of deep understanding and semantic search of arbitrary mobile sensing data. We extract common bases of motion sensing data leveraging unsupervised deep learning. To discover the hierarchical nature of human activities, we propose a universal multi-resolution embedding for all activities requiring no pre-knowledge. Based on the embedding, we design an innovative system **Lasagna** to manage and search motion data semantically. We implement a prototype system and the comprehensive evaluations show that the prototype can achieve highly accurate activity classification (precision is 98.9%) and search (recall is almost 100% and precision is about 90%) over diverse activities.

Lasagna is a first step towards mobile sensing data search engine. It opens up new possibilities to many promising applications and benefits the development of mobile industry and other research and commercial fields. Meanwhile, many open issues are raised and to be solved in the future work.

9. ACKNOWLEDGEMENT

We want to express our sincere appreciation to our shepherd and all the anonymous reviewers for valuable comments and suggestions. The research is supported in part by NSFC Major Program under Grants No. 61190110, NSFC under Grants No. 61572281, No. 61472218, and China Postdoctoral Science Foundation under Grant No. 2015M580101, 2016T90096. The research of Li is partially supported by NSF ECCS-1247944, NSF CMMI 1436786, NSF CNS 1526638, National Natural Science Foundation of China under Grant No. 61520106007. Lan Zhang and Yunhao Liu are the co-corresponding authors.

10. REFERENCES

[1] ANJUM, A., AND ILYAS, M. U. Activity recognition using smartphone sensors. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pp. 914–919.

[2] BAO, L., AND INTILLE, S. S. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing* (2004), Springer, pp. 1–17.

[3] CHEN, K.-Y., ASHBROOK, D., GOEL, M., LEE, S.-H., AND PATEL, S. Airlink: sharing files between multiple devices using in-air gestures. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 565–569.

[4] CHENG, H.-T. *Learning and Recognizing The Hierarchical and Sequential Structure of Human Activities*. PhD thesis, CARNEGIE MELLON UNIVERSITY, 2013.

[5] GEORGE, E. I., AND MCCULLOCH, R. E. Variable selection via gibbs sampling. *Journal of the American Statistical Association 88*, 423 (1993), 881–889.

[6] GUO, Y., YANG, L., DING, X., HAN, J., AND LIU, Y. Opensesame: Unlocking smart phone through handshaking biometrics. In *Proceedings of IEEE INFOCOM* (2013), pp. 365–369.

[7] HARBACH, M., VON ZEZSCHWITZ, E., FICHTNER, A., DE LUCA, A., AND SMITH, M. It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *SOUPS 2014*, pp. 213–230.

[8] HINTON, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation 14*, 8 (2002), 1771–1800.

[9] HONG, F., WEI, M., YOU, S., FENG, Y., AND GUO, Z. Waving authentication: Your smartphone authenticate you on

motion gesture. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (2015), pp. 263–266.

[10] HUYNH, T., FRITZ, M., AND SCHIELE, B. Discovery of activity patterns using topic models. In *Proceedings of the 10th international conference on Ubiquitous computing* (2008), ACM, pp. 10–19.

[11] JIANG, W., AND YIN, Z. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference* (2015), pp. 1307–1310.

[12] KIROS, R., SALAKHUTDINOV, R., AND ZEMEL, R. Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning* (2014), pp. 595–603.

[13] KWAPISZ, J. R., WEISS, G. M., AND MOORE, S. A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter 12*, 2 (2011), 74–82.

[14] KWON, Y., KANG, K., AND BAE, C. Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications 41*, 14 (2014), 6067–6074.

[15] LANE, N. D., AND GEORGIEV, P. Can deep learning revolutionize mobile sensing? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (2015), ACM, pp. 117–122.

[16] LANE, N. D., GEORGIEV, P., AND QENDRO, L. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 283–294.

[17] LEE, H., GROSSE, R., RANGANATH, R., AND NG, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, pp. 609–616.

[18] LEE, H., PHAM, P., LARGMAN, Y., AND NG, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (2009), pp. 1096–1104.

[19] LI, Z., LI, M., WANG, J., AND CAO, Z. Ubiquitous data collection for mobile users in wireless sensor networks. In *INFOCOM, 2011 Proceedings IEEE* (2011), IEEE, pp. 2246–2254.

[20] LIU, J., ZHONG, L., WICKRAMASURIYA, J., AND VASUDEVAN, V. User evaluation of lightweight user authentication with a single tri-axis accelerometer. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (2009), ACM, p. 15.

[21] LIU, X., ZHOU, Z., DIAO, W., LI, Z., AND ZHANG, K. When good becomes evil: Keystroke inference with smartwatch. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 1273–1285.

[22] LÜTKEPOHL, H. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[23] MATSUBARA, Y., SAKURAI, Y., AND FALOUTSOS, C. Autoplait: Automatic mining of co-evolving time sequences. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 193–204.

[24] MICHALEVSKY, Y., BONEH, D., AND NAKIBLY, G. Gyrophone: Recognizing speech from gyroscope signals. In *23rd USENIX Security Symposium* (2014), pp. 1053–1067.

[25] OWUSU, E., HAN, J., DAS, S., PERRIG, A., AND ZHANG, J. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications* (2012), ACM.

[26] PARATE, A., CHIU, M.-C., CHADOWITZ, C., GANESAN, D., AND KALOGERAKIS, E. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services* (2014), pp. 149–161.

[27] RABINER, L. R., AND JUANG, B.-H. An introduction to hidden markov models. *ASSP Magazine, IEEE 3*, 1 (1986), 4–16.

[28] RAMANATHAN, V., TANG, K., MORI, G., AND FEI-FEI, L. Learning temporal embeddings for complex video analysis. In *The IEEE International Conference on Computer Vision* (2015).

[29] REN, Y., CHEN, Y., CHUAH, M. C., AND YANG, J. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. *IEEE Transactions on Mobile Computing 14*, 9 (2015), 1961–1974.

[30] ROY, N., WANG, H., AND ROY CHOUDHURY, R. I am a smartphone and i can tell my user's walking direction. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services* (2014), ACM, pp. 329–342.

[31] SCHROFF, F., KALENICHENKO, D., AND PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 815–823.

[32] SHAHZAD, M., LIU, A. X., AND SAMUEL, A. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (2013), ACM, pp. 39–50.

[33] SHOAIB, M., BOSCH, S., INCEL, O. D., SCHOLTEN, H., AND HAVINGA, P. J. Fusion of smartphone motion sensors for physical activity recognition. *Sensors 14*, 6 (2014), 10146–10176.

[34] SUN, Y., CHEN, Y., WANG, X., AND TANG, X. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems 27*. 2014, pp. 1988–1996.

[35] SUN, Y., LIANG, D., WANG, X., AND TANG, X. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873* (2015).

[36] SUN, Y., WANG, X., AND TANG, X. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 2892–2900.

[37] SUN, Z., PUROHIT, A., BOSE, R., AND ZHANG, P. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services* (2013), ACM, pp. 263–276.

[38] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842* (2014).

[39] TUNG, Y.-C., AND SHIN, K. G. Echotag: accurate infrastructure-free indoor location tagging with smartphones.

In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (2015), ACM, pp. 525–536.

[40] VINYALS, O., TOSHEV, A., BENGIO, S., AND ERHAN, D. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3156–3164.

[41] WANG, H., LAI, T. T.-T., AND ROY CHOUDHURY, R. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (2015), ACM, pp. 155–166.

[42] WANG, P., WANG, H., AND WANG, W. Finding semantics in time series. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (2011), pp. 385–396.

[43] WANG, R., HARARI, G., HAO, P., ZHOU, X., AND CAMPBELL, A. T. Smartgpa: how smartphones can assess and predict academic performance of college students. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 295–306.

[44] WANG, X., AND GUPTA, A. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2794–2802.

[45] WANG, X., SMITH, K., AND HYNDMAN, R. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery 13*, 3 (2006), 335–364.

[46] WU, F.-J., CHU, F.-I., AND TSENG, Y.-C. Cyber-physical handshake. In *ACM SIGCOMM Computer Communication Review* (2011), vol. 41, pp. 472–473.

[47] XU, Z., BAI, K., AND ZHU, S. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks* (2012), pp. 113–124.

[48] ZHANG, L., LIU, K., JIANG, Y., LI, X.-Y., LIU, Y., AND YANG, P. Montage: Combine frames with movement continuity for realtime multi-user tracking. In *Proceedings of IEEE INFOCOM* (2014), pp. 799–807.

[49] ZHANG, L., PATHAK, P. H., WU, M., ZHAO, Y., AND MOHAPATRA, P. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (2015), ACM, pp. 301–315.

[50] ZHENG, N., BAI, K., HUANG, H., AND WANG, H. You are how you touch: User verification on smartphones via tapping behaviors. In *IEEE 22nd International Conference on Network Protocols (ICNP)* (2014), pp. 221–232.