Imaging and fusing time series for wearable sensor-based human activity recognition

Yibo Zhang

ARTICLE INFO

Keywords:
- Body sensor networks
- Human activity recognition
- Deep residual network
- Multi-sensor data fusion

ABSTRACT

To facilitate data-driven and informed decision making, a novel deep neural network architecture for human activity recognition based on multiple sensor data is proposed in this work. Specifically, the proposed architecture encodes the time series of sensor data as images (i.e., encoding one time series into a two-channel image), and leverages these transformed images to retain the necessary features for human activity recognition. In other words, based on imaging time series, wearable sensor-based human activity recognition can be realized by using computer vision techniques for image recognition. In particular, to enable heterogeneous sensor data to be trained cooperatively, a fusion residual network is adopted by fusing two networks and training heterogeneous data with pixel-wise correspondence. Moreover, different layers of deep residual networks are used to deal with dataset size differences. The proposed architecture is then extensively evaluated on two human activity recognition datasets (i.e., HHAR dataset and MHEALTH dataset), which comprise various heterogeneous mobile device sensor combinations (i.e., acceleration, angular velocity, and magnetic field orientation). The findings demonstrate that our proposed approach outperforms other competing approaches, in terms of accuracy rate and F1-value.

1. Introduction

In our data-driven and data-rich society (e.g., real-time video feeds from CCTVs, and other sensing data from different data sources), Body Sensor Networks (BSN) [1] have gained widespread attention, including in the academic literature, such as those of human-computer interaction and ubiquitous computing (e.g., user identification [2] and human activity recognition [3]). Constant advances in hardware and software (e.g., inexpensive mobile devices with embedded powerful sensors and wireless technology [4,5]) have also ease human activity recognition using sensor data from BSN, and example applications include healthcare [6,7], heart-rate-based emotion reactions [8,9], activity monitoring [10,11], and commercial applications such as fitness tracking [12] and signal processing in node environment [13]. For example, human activity recognition can leverage time series signal of mobile device sensor data, where representative data features are extracted for classification and discrimination using various algorithms.

Traditionally, human activity recognition using mobile device sensors has been defined as a multivariate time series classification problem. To solve the problem, a key step is feature extraction, for example relying on some statistical features of the raw signal (e.g., variance, mean, entropy, and correlation coefficients) [14], or including some cross-formal coding (e.g., signals with Fourier transform and wavelet transform). These heuristic features are widely used in analyzing time series data.

However, in the deep learning framework, we can build a multi-layer deep structure to automatically extract relevant features. A deep learning model can train data in both supervised and unsupervised manner, and it has significant effects in processing graphical data. Moreover, the representation of features of time series has recently attracted widespread attention. The most successful way is to describe features as visual cues [15]. Depending on supervisory and non-hyper-visual learning techniques in computer vision, time series can be re-coded into images to enable machines to perform image recognition. This technology has been applied in speech recognition [16], classification [17] and radio frequency identification [18], and has shown to be more effective.

Therefore, the proposed architecture in this work integrates a method that transforms sensor data into some visual images, and a framework that enables human activity recognition to be carried out by using deep residual networks in image recognition. Specifically, we summarize the key contributions of this paper to be as follows.
A feature engineering method is developed to transfer sensor-based time series data into different images, by unifying the global and local features from time series.

A fusion framework is proposed to automatically extract image features from the generated images and to recognize user behavior by distinguishing different image features.

Now, we will describe the layout for the remaining of this paper. In the next section, we will briefly review related work. Our proposed approach is described in Section 3. In Section 4, we evaluate the proposed framework on both HHAR [19] and MHEALTH datasets [20] and describe the findings. Specifically, the findings demonstrate that our proposed approach works well with most types of heterogeneous multi-dimensional time series measurements. Finally, we conclude the paper in the last section.

2. Related work

In this section, we will briefly review the related literature on imaging time series, image recognition search, and heterogeneous sources processing.

Imaging Time Series. Encoding time series as images plays an important role in many classification tasks. In [21], for example, the authors investigated the use of recurrence plots as data representation for time series classification. In their approach, texture features are extracted on recurrence patterns from time series by using visual descriptors, such as Gabor and Local Binary Patterns, and Support Vector Machine (SVM) classifiers are then used. In [17], the authors used the Gramian Angular Fields (GAF) and Markov Transition Fields (MTF) to encode time-series signals as images, before classifying the time-series images using a Tiled Convolutional Neural Network model. Images can also be generated by calculating the distance between two points of time series [18], or applying some different image processing features, including HOG, BRISK, and SURF. For example, the approach in [15] uses compression distance for time series classification by extending the recurrence plot paradigm. These methods suggest that features of time series can be extracted from generated images and be used to perform different recognition or classification tasks.

Image Recognition Search. We will now briefly review recent related work relating to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) since AlexNet [22]. VGGNet [23] demonstrates the potential to increase network performance by increasing the number of network layers and depth based on previous convolutional neural network (CNN) network architectures. Based on VGGNet, ResNet [24] explicitly fits a residual mapping that is realized by feedforward neural networks with shortcut connections. The network depth is then increased to obtain improved results. A notable part of GoogleLeNet [25], another well-known network, is that the architecture greatly improves the utilization rate of computing resources. In the well-designed network, computational overhead remains constant with the increase of model depth and width. Recently, ResNet with GoogLeNet were combined to form Inception-ResNet-v1 and Inception-ResNet-v2 [26], which reportedly achieved improved results with less training time. In 2017, SENet [27], the winner of the ILSVRC 2017 classification category, recalibrates channel-wise feature responses without relying on new spatial dimensions.

Heterogeneous Sources Processing. There have been several related works on processing heterogeneous sources, for example relating to action recognition tasks based on RGB-D or video. One such example work is that of Kong and Fu, who demonstrated how one can compress and project the heterogeneous features to a shared space, and capture useful information for action recognition by learning both independent private space and shared space [28]. In [29], the authors proposed to encode the RGB and depth video into structured dynamic images, and use a c-ConvNet to leverage the conjoint information of the heterogeneous modalities. In [30], the authors proposed the two-stream ConvNet architecture. First, they use RGB and optical flow frames to decompose video into spatial and temporal components, where each stream performs video recognition independently. Then, softmax scores are combined using late fusion for final classification. The approach in [31] is based on the two-stream architecture, where spatial and temporal features are learned pixel by pixel. The more recent approach in [32] fuses geometric features for Skeleton-Based action recognition, using a new smoothed score fusion technique to learn classification from different streams.

We observe that in human activity recognition, generally conventional machine learning and simple neural network are used. Therefore, in this paper we proposed a feature engineering method to import more representative information in the generated GAF images, and explore the potential of more effective approaches for feature extraction and more complex network structures, in order to significantly improve the experimental accuracy.

3. Proposed architecture

In this section, we will first perform a preliminary investigation on GAF and ResNet. Then for our activity recognition challenge, we will explore the feature engineering in GAF Images and present the proposed fusion ResNet framework.

3.1. Preliminary investigation in GAF and ResNet

Gramian Angular Fields (GAF) [17] represent time series data in a polar coordinate system instead of the Cartesian coordinates. In the Gramian matrix, each element is the cosine of the summation or difference of pairwise temporal values. Given a time series \( X = \{x_1, x_2, \ldots, x_n\} \) of \( n \) real-valued observations, a normalization method is used to process \( X \) so that all values fall in the interval [-1,1]. The formula of feature quantification is as follows:

\[
\tilde{x}_i^d = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)}
\]

(1)

It is the same as the range of the cosine function, so we can regard the normalized time series \( \tilde{X} \) as the angular cosine and the time step as the radius with the equation below:

\[
\begin{align*}
\phi &= \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\
\rho &= \frac{t_i}{T} \quad t_i \in \mathbb{N}
\end{align*}
\]

(2)

In the above equation, \( t_i \) denotes the time step and \( N \) is a constant factor to regularize the span of the polar coordinate system. In other words, we represent time series using polar coordinates. The time series and corresponding polar coordinate presentation are shown in Fig. 1. There are two important properties in this encoding map, based on Eq. (2). First, it is bijective. The unique inverse function of Eq. (2) produces one, and only one, result \( \phi \) in the polar coordinate presentation, as \( \cos(\phi) \) is monotonic decreasing when \( \phi \in [0, \pi] \). Second, polar coordinates preserve absolute temporal relations, a feature that Cartesian coordinates lack. In Cartesian coordinates, the area is defined by \( S_{ij} = f(x(i))dx(i) \).

![Proposed encoding map of transforming rescaled time series into polar coordinate.](image-url)
and we have $S_{i,j+k} = S_{i,j+k}$ if $f(x(t))$ has the same values on $[i, i + k]$ and $[j, j + k]$. However, in polar coordinates, if the area is defined as $S'_j = \int_{0}^{\theta} f(x(t)) \, dx(t)$ then $S_{i,j+k} \neq S'_{i,j+k}$. This means that the absolute value of time step $i$ and time step $j$ also determine the result, not only the time interval $|i - j|$.

After transforming to the polar coordinate system, by exploiting the angular perspective and considering the trigonometric sum between each point, the temporal correlation within different time intervals can be identified. The Gramian Angular Summation Field (GASF) is defined as follows:

$$ G = \begin{bmatrix}
\cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\
\cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\
\vdots & \ddots & \vdots \\
\cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n)
\end{bmatrix} $n \in \mathbb{R}$$

(3)

$$ = \begin{bmatrix}
\tilde{x}_1^\prime & \ldots & \tilde{x}_n^\prime \\
\tilde{x}_2^\prime & \ldots & \tilde{x}_n^\prime \\
\vdots & \ddots & \vdots \\
\tilde{x}_n^\prime & \ldots & \tilde{x}_n^\prime
\end{bmatrix} \cdot \begin{bmatrix}
\mathbf{I} - \mathbf{X}^2 \\
\mathbf{I} - \mathbf{X}^2 \\
\mathbf{I} - \mathbf{X}^2
\end{bmatrix} $$

(4)

In the above equation, $f$ is the unitary row vector $[1, 1, \ldots, 1]$.

After transforming time series to the polar coordinate system, each time step is taken as a one-dimensional metric space. By defining the inner product $<x, y> = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$, $G$ is a Gramian matrix:

$$ = \begin{bmatrix}
\cos(\phi_1 - \phi_1) & \cdots & \cos(\phi_1 - \phi_n) \\
\cos(\phi_2 - \phi_1) & \cdots & \cos(\phi_2 - \phi_n) \\
\vdots & \ddots & \vdots \\
\cos(\phi_n - \phi_1) & \cdots & \cos(\phi_n - \phi_n)
\end{bmatrix} $n \in \mathbb{R}$$

(6)

$$ = \begin{bmatrix}
\tilde{x}_1^\prime & \ldots & \tilde{x}_n^\prime \\
\tilde{x}_2^\prime & \ldots & \tilde{x}_n^\prime \\
\vdots & \ddots & \vdots \\
\tilde{x}_n^\prime & \ldots & \tilde{x}_n^\prime
\end{bmatrix} \cdot \begin{bmatrix}
\mathbf{I} - \mathbf{X}^2 \\
\mathbf{I} - \mathbf{X}^2 \\
\mathbf{I} - \mathbf{X}^2
\end{bmatrix} $$

(7)

Fig. 2 presents the proposed encoding maps of both GASF and GADF. GAF images provide a way to preserve temporal correlations, since the position moves from top to bottom or from left to right along with time increasing. GAF images contain temporal dependency because $G_{i,j|j,j+k)}$ represents the relative dependency by superposition/differencing of directions with respect to time interval $k$. The main diagonal $G_{i}$ is the special case when $k = 0$, which contains the original angular information. With the encoding matrix, the time series can be approximately reconstructed as GAF images, and their deep hidden features can be learned in the deep neural network.

To learn the GAF image features, a deep CNN is employed to extract a compact representation of each generated GAFs. The CNN features of each time series are extracted with the pre-trained Residual Network model. Deep Residual Network [24] solves the puzzling degradation problem in the network over 20 layers, and can extract the depth features of the image more deeply. The pre-trained ResNet34 model contains 5 sets of convolutional layers (i.e., conv1, conv2, conv3, conv4, conv5). Each set includes a building block of several convolutional layers with the increasing kernel size, while the kernel size are same in each building block. Totally, there are 33 convolutional layers and 1 fully connected layer in the network. Each building block has a shortcut connection that skips the block for identity mapping and adds the residual mapping of the block to become the final underlying mapping. For the first building block of conv3, conv4, and conv5, we need perform a linear projection to identity mapping for match the dimensions, before it calculates with residual mapping. The Building blocks defined as follows:

$$ y = F(x, W_i) + x $$

(8)

Here, $x$ and $y$ are the input and output vectors of the layers. A linear projection $W_i$ is also performed as follows:

$$ y = F(x, W_i) + W_i x $$

(9)

3.2. Feature engineering in GAF images

According to Eqs. (1) and (2), once time series $X$ are converted into polar coordinates, their maximum and minimum values are represented by the extreme value of $\alpha$. However, such normalization method has two limitations in sensor data based action recognition. First, the absolute magnitude value will be lost, and the observers' actions (e.g., walking and biking) have their periodicity in the representation of the sensor data, which can be approximated as swaying along a central line as shown in Fig. 1. We argue that if different user actions have approximate periodicity, then normalized time series $X$ would be approximated and further lead to loss of important features (e.g., range and interval) that can potentially be extracted from the original data. Second, data instability will have a negative impact on GAF image reconstruction. As the selected time sequence for generating GAF images is short, most GAF images do not cover the small amount of large instantaneous values. However, because of the existence of such unexpected values, a few generated GAF images eventually result in significant differences in images with similar actions, and further affecting the classification performance.

To solve the above problem, we propose a special normalization method, which allows all fragments intercepted from time series to be mapped to one polar coordinate system. In order to maximize the rippling of each fragment ripples on the surface of the polar coordinate system, while ensuring all converted values to remain meaningful (i.e., no overflow), we use the maximum and minimum values in all time series data in one channel as the upper and lower limits of the interval of the polar coordinate system. In this way, given two different time series $X$ and $X'$, if $X_\text{c} > X'_\text{c}$, where $x_\text{c}$ from $X$ and $x'_\text{c}$ from $X'$, we have $X_\text{c} > X'_\text{c}$ in re-scaled time series $\tilde{X}$ and $\mathbf{X}'$. Thus, to maintain absoluteness and stability of the original data, we can use the following revised formula for feature quantification (instead of Eq. (1)):

$$ \tilde{X}_\text{c} = \frac{(X_\text{c} - \text{MAX}) + (X_\text{c} - \text{MIN})}{\text{MAX} - \text{MIN}} $$

(10)

In the above equation, MAX and MIN are the maximum and minimum values of all time series data in one channel. In our experiments, we used 4 sets of MAX and MIN values in each isomorphic multi-dimensional data, namely: horizontal, longitudinal, vertical, and the resultant of such three directions (in order to reduce the negative impact introduced by different holding states of smartphones).

We remark that the normalization using maximum and minimum values from a single intercept sequence is also useful. We observe that
Fig. 3. Global GAF and local GAF in the same sequence of time series, recorded when one subject is ‘standing’.

it can be challenging to distinguish static activities such as standing and sitting based on sensor data, as for such static (or almost completely static) actions, the sensor data is almost unchanged during the period of intercepting fragments, and appears nearly straight in polar coordinates, as shown in Fig. 3. Therefore, we preserve the original GAF generation algorithm, approximate to enlarging the range of data values in polar coordinate graph, so as to be able to mine the depth characteristics of static action.

We define images normalized by Eq. (1) as the local GAF; similarly, images normalized by Eq. (10) as global GAF. To improve recognition performance, and facilitate the preserving of the original features of time series and mining of deeper features, we unify the global and local GAF images in the following proposed fusion ResNet framework. This doubles the dimension of the generated images. The evaluation in the Section 4 will demonstrate that such unification and fusion processes can significantly improve human activity recognition performance in comparison to other competing approaches.

3.3. Fusion ResNet framework

Our proposed fusion ResNet framework is presented in Fig. 4, which fuses acceleration and angular velocity information, and learns the generated GAF image pixels correspondences between acceleration and angular velocity features. Specifically, given four transformed polar coordinate data series, we encode them to four types of GAF images (i.e., global GAF images of Acceleration, local GAF images of Acceleration, global GAF images of angular velocity, and local GAF images of angular velocity). Every image has four channels corresponding to three directions (horizontal, longitudinal, vertical) and the resultant of such three directions.

The accelerometer records different periodic acceleration data, while the gyroscope records different periodic angular velocity data correspondingly, and their fusion then discriminates the action. To facilitate this, two images should have the same pixel size, and channels of a GAF image should correspond to the channels of the other GAF images.

We calculate fusion maps $m$ for given GAF images with input from accelerometer $I_{acc}$ and gyroscope $I_{ang}$, as:

$$m^d = f(I_{acc}^d, I_{ang}^d; w^d),$$  \hspace{1cm} (11)

where the function $f$ outputs convolutional maps, which are parameterized by convolutional kernels $w$, and $d$ represents the directions of horizontal, longitudinal, vertical, and their corresponding resultant direction. For simplicity, we only fuse them at the end of one convolution set of residual network. Hence, we use the following calculation:

$$y^d = f(F(x^d_{acc}) + x^d_{acc}, F(x^d_{ang}) + x^d_{ang}; w^d)$$  \hspace{1cm} (12)

3.4. Implementation

On the mobile devices, we intercept every 5 sec of data in the dataset into one time series. In our implementation, we use equidistant sampling to extract 100 samples. For instance, if the frequency of time series data is 100HZ, we will sample every five time steps over 500 consecutive time steps to extract a sequence with the size is 100 for subsequent process. Then, the size of the GAF images is regulated to 100*100 pixels. We use two types of sensors to generate two sets of GAF images, each of them with 4 channels (i.e., horizontal, longitudinal, vertical and their resultant) for containing multi-dimensional time series measurements. Thus, 2 sensors * 2 type GAF images * 100*100 pixels * 4 channels ≈ 224*224*3, where such size is suit for Deep Residual Network [24]. After each convolutional layer, we use Batch Normalization [33] before the activation layer. The size of the mini-batch used is 256. The learning
rate starts from 0.1. When the error rate is stable, the learning rate is divided by 10. We set the weight decay to 0.0001 and a momentum of 0.9 and without dropout. For fusion, we only propagate back to the injected fusion layer, since full back-propagation did not result in an improvement. And we only fuse between the layers with the same output resolution.

4. Evaluation

The following datasets are used in our evaluation.

**HHAR.** The Heterogeneity Human Activity Recognition (HHAR) dataset from smartphones and smartwatches is a dataset, which has been devised to benchmark human activity recognition algorithms (classification, automatic data segmentation, sensor fusion, feature extraction, etc.) in real-world contexts; specifically, the dataset is gathered with a variety of device models and use-scenarios, in order to reflect sensing heterogeneity to be expected in real deployments. HHAR contains the readings of two motion sensors (accelerometer and gyroscope) commonly found in smartphones. Readings were recorded when users executed activities scripted in no specific order, only carrying smartphones and smartwatches. This dataset contains 6 activities (biking, sitting, walking, standing, climbing up stairs, and walking down stairs) by recording 9 users, and 6 types of mobile devices (4 smartphones and 2 smartwatches). These smartphone models yielded different maximum sampling frequencies: 200Hz, 150Hz, 100Hz and 50Hz, approximately. The smart watches also varied in the supported maximum sampling rate, e.g., 200Hz and 100Hz. Different distance values are used to handle the data at different frequencies when using equidistant sampling. In our experiment, accelerometer and gyroscope measurements are model inputs, while activities are used as labels. Different types of devices collect data at different frequencies, but we segment raw measurements into 5-sec samples for all types of devices. In total, 14,268 samples were used in the evaluation.

**MHEALTH.** The Mobile Health (MHEALTH) dataset is devised to benchmark methods of human activities recognition based on multi-modal wearable sensor data. Sensors placed on the subject’s chest, right wrist and left ankle are used to measure the motion experienced by diverse body parts (i.e., acceleration, angular velocity, and magnetic field orientation). Moreover, the sensor positioned on the chest provides 2-lead ECG measurements. Similar to [34], the activities are categorized into 6 classes, namely: lying, sitting/standing, walking, running, cycling and other activities. All sensing modalities are recorded at a sampling rate of 500Hz. Similar to the evaluation undertaken using the HHAR dataset, raw measurements are also segmented into 5-sec samples and the total number is 1318.

### 4.1. Performance metric

According to the objective definition, human activities recognition of smartphone users belongs to multi-classification problem. In a multi-classification problem, each class can be regarded as an independent set of samples as positive class while the others as negative class. The performance of human activities recognition of smartphone users is evaluated by Accuracy and Macro $F_1$ score.

Accuracy is the proportion of correctly identified examples in all examples, which represents the rate at which users are correctly identified. It is defined as

$$\text{accuracy} = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} \mathcal{L}(y_i = y_0)$$

where $\hat{y}_i$ is the predicted value of the i-th sample, $y_i$ is the corresponding true value and $\mathcal{L}$ is the indicator function.

$F_1$-Score is an index used to measure the accuracy of multi-classification models in statistics, which is the harmonic mean of precision and recall:

$$F_1 \text{- score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Moreover, Macro $F_1$ Score first calculate the total number of true positives, false negatives and false positives then calculate $F_1$:

$$\text{Macro} F_1 \text{- score} = \frac{2 \cdot P_{\text{Macro}} \cdot R_{\text{Macro}}}{P_{\text{Macro}} + R_{\text{Macro}}}$$

where $F_1$-score is the $F_1$-score of the i-th class in the test dataset, $P_{\text{Macro}}$ and $R_{\text{Macro}}$ are the Macro precision and recall respectively and they are actually the unweighted mean for all categories:

$$P_{\text{Macro}} = \frac{1}{N} \sum_{k=1}^{N} P_k \quad R_{\text{Macro}} = \frac{1}{N} \sum_{k=1}^{N} R_k$$

### 4.2. Encoding sensor data as images

In this section, we present the results for the verification accuracy of encoding sensors data as images. It is a novel method for human activity recognition of sensor time series data. Minkowski distance of order 2 (i.e., euclidean distance), Minkowski distance of order 3, and the Chebyshev distance are used as our baseline. We use the same network architecture as shown in Fig. 4. In evaluation, for this experiment we adopt the standard 10-fold cross-validation testing.

We compare different strategies to encode images in Table 1, where we report the average accuracy and Macro $F_1$ on HHAR. We first observe that it is feasible for each algorithm to encode the acceleration and angular velocity data of the sensor into images, and then use the deep residual network for activity recognition. This is due to the strong deep feature extraction capability of the deep residual networks. We observe that our algorithm improves both image generation and neural network to achieve better performance.

Next, we observe that in the comparison to the three distances, we achieve favorable performance for processing time series data (e.g., using either global GAF or local GAF only). This shows that the image generation method we use is capable of extracting features of human activities in sensor data.

Moreover, as previously described in Section 3, it is challenging for global GAF to distinguish between static activities such as standing and sitting in human activity recognition based on sensor data (i.e., as shown in Fig. 5), while local GAF cannot preserve features like absoluteness of time series data. Thus, we unified global and local GAF simultaneously, and it should be noted that such combination achieves improved performance.

<table>
<thead>
<tr>
<th>Encoding images Method</th>
<th>Accuracy</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minkowski order 2</td>
<td>0.8907</td>
<td>0.8909</td>
</tr>
<tr>
<td>Minkowski order 3</td>
<td>0.9082</td>
<td>0.9084</td>
</tr>
<tr>
<td>Chebyshev</td>
<td>0.8732</td>
<td>0.8730</td>
</tr>
<tr>
<td>global GAF</td>
<td>0.9129</td>
<td>0.9131</td>
</tr>
<tr>
<td>local GAF</td>
<td>0.9370</td>
<td>0.9384</td>
</tr>
<tr>
<td>global GADF</td>
<td>0.9124</td>
<td>0.9126</td>
</tr>
<tr>
<td>local GADF</td>
<td>0.9371</td>
<td>0.9372</td>
</tr>
<tr>
<td>global GADF + local GAF</td>
<td>0.9676</td>
<td>0.9674</td>
</tr>
<tr>
<td>global GADF + local GAF</td>
<td>0.9663</td>
<td>0.9661</td>
</tr>
</tbody>
</table>

4.3. Fusing heterogeneous data

For these experiments, we use the same combined transformed images (global and local GAF). We need a series of experiments to obtain the best fusion strategy and determine the best layer to fuse. We also adopt the standard 10-fold cross-validation for testing in the experiments described in this section.

First, we compare different fusion strategies from Feichtenhofer et al. [31] where we report the accuracy and macro-F1 values in Table 2. The
fusion layer is injected at the last convolutional block, and its input is the output of conv3 from the two heterogeneity sensors data streams. Although it is still providing coarse information, some features are already highly informative. After the fusion layer a single processing stream is used. We see that Multy performs considerably lower than a strategy without fusion while others perform better (Concatenation performs best). Simple fusion via Summation, Maximum or Average have slightly lower performance than Concatenation. It suggests that simply calculating the feature maps is potentially a good enough fusion technique, with a 1–2% improvement since a one or two percentage points improvement in a structure that already has excellent performance is an achievement.

Then, a comparative summary of the fusion from different convolutional blocks is presented in Table 3. Our fusion layer is added at the bottom of all convolutional blocks. To be more specific, the number of Res34 convolution blocks from conv2 to conv4 is 3, 4, and 6, respectively. So we have added fusion layer after conv2,3, conv3,4, and conv4,6, respectively, and while fused by concatenation, we noted that the optimal results are observed in conv3 (i.e., the best fusion point).

### 4.4. Comparison with state-of-the-art approaches

To demonstrate the utility of our proposal, we choose the following state-of-the-art method as the baselines, namely: Random Forest(RF), Support Vector Machine(SVM), Restricted Boltzmann Machine(RBM), MultiRBM, and two deep learning techniques: Fully Convolutional Network(FCN) and Long Short-Term Memory(LSTM). RF [19], which selects popular frequency domain and time-domain features from [40] and ECD features from Hammerla et al. [41], uses random forest as the classifier. SVM selects the same features as the RF model, except that the SVM model uses support vector machine as the classifier [19]. RBM is based on stacked restricted Boltzmann machines, and this model uses frequency domain representations as inputs [42]. MultiRBM processes each sensor input with a single stacked restricted Boltzmann machine, then merges the results with another stacked restricted Boltzmann machine [43]. Two deep learning methods, FCN [38] consists of three stacked 1D convolution blocks, each of them consists of a 1D convolutional layer accompanied by a ReLU activation function and followed by batch normalization. After final convolution block, a global average pooling layer and a softmax classification layer are applied to pass. The other network, LSTM [39], which possess a vanishing gradient problem, are an improvement over the general recurrent neural networks and recognized as the preferred neural network for time series data.

We perform leave-one-user-out evaluation (with training on all but one user), and present the performance evaluation findings in Table 4. We observe that even the simplest generated image by distance algorithm can be recognized by using the residual network model, and the accuracy and F1 can exceed 80%. Their performance is better than that of the baseline experiments, regardless they are two hand-crafted feature based algorithms, RF and SVM, two deep models, RBM and MultiRBM, or two deep learning based models, FCN and LSTM. In our method, GADF and GASF images generated by GAF have a significant performance improvement of more than 10% compared with the comparative traditional machine learning experiments, more than 6% compared with the comparative deep learning experiments, and the accuracy and F1 reaches 93.4%.

We present the confusion matrix in Fig. 6. It can be observed that we can effectively solve the challenges of predicting sitting and standing, climbStair – up and climbStair – down, although they are very similar. In other words, our algorithm can mine subtle differences in deep characteristics between similar activities. The majority of prediction errors are from predicting climbing and walking, and this is arguably due to

### Table 2

<table>
<thead>
<tr>
<th>Images</th>
<th>Fusion Method</th>
<th>Accuracy</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>GASF</td>
<td>No Fusion</td>
<td>0.9424</td>
<td>0.9442</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>0.9564</td>
<td>0.9594</td>
</tr>
<tr>
<td></td>
<td>Multy</td>
<td>0.9377</td>
<td>0.7385</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>0.9572</td>
<td>0.9567</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.9556</td>
<td>0.9556</td>
</tr>
<tr>
<td></td>
<td>Concatenation</td>
<td>0.9676</td>
<td>0.9674</td>
</tr>
<tr>
<td>GADF</td>
<td>No Fusion</td>
<td>0.9463</td>
<td>0.9463</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>0.9546</td>
<td>0.9544</td>
</tr>
<tr>
<td></td>
<td>Multy</td>
<td>0.9307</td>
<td>0.9310</td>
</tr>
<tr>
<td></td>
<td>Maximum</td>
<td>0.9577</td>
<td>0.9575</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>0.9562</td>
<td>0.9564</td>
</tr>
<tr>
<td></td>
<td>Concatenation</td>
<td>0.9663</td>
<td>0.9661</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Images</th>
<th>Fusion layers</th>
<th>Accuracy</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>GASF</td>
<td>Conv2,3</td>
<td>0.9549</td>
<td>0.9548</td>
</tr>
<tr>
<td></td>
<td>Conv3,4</td>
<td>0.9676</td>
<td>0.9674</td>
</tr>
<tr>
<td></td>
<td>Conv4,6</td>
<td>0.9588</td>
<td>0.9586</td>
</tr>
<tr>
<td>GADF</td>
<td>Conv2,3</td>
<td>0.9541</td>
<td>0.9537</td>
</tr>
<tr>
<td></td>
<td>Conv3,4</td>
<td>0.9663</td>
<td>0.9661</td>
</tr>
<tr>
<td></td>
<td>Conv4,6</td>
<td>0.9572</td>
<td>0.9575</td>
</tr>
</tbody>
</table>

**Fig. 5.** Confusion matrix of single global GAF. Predicting Sitting as Standing leads the majority of errors, with a minor errors about climbing up/down of stairs.
the staircase landing (i.e., it is challenging to define whether a user is climbing or walking if the individual passes the stair landing more than 3s).

### 4.5. Evaluation using MHEALTH dataset

Finally, we undertake more evaluation using the MHEALTH dataset to verify the generalization of our method. Similar to our evaluation using the HHAR dataset, we perform leave-one-user-out evaluation. Simply, we use 18-layers ResNet, with fusion data by concatenation after conv3_2. We show the results and comparison with the state-of-the-art in Table 5. On each setting of these method, the mean and the standard deviation of accuracy are recorded.

We observe that our method achieves excellent performance in the MHEALTH dataset. For MHEALTH dataset, it has merged classification categories, which reduces the difficulty of recognition in a certain extent. Under the same experimental conditions, the superiority of our method is proved with average accuracy in 99.2%. In other words, our method has good generalization in other datasets as well as other types of sensor data.

### 5. Conclusion

In this paper, a deep learning network architecture was proposed for human activity recognition based on mobile sensor data. Specifically, we proposed a novel method to encode time series into GAF images by unifying global and local time series features. This new processing method can be trained in mainstream image recognition residual networks. We designed a number of experiments to verify the feasibility of imaging time series. And we proposed a fusion ResNet to solve the problem of heterogeneity data. Our architecture enhances the discriminative power of the deep features and our results demonstrate its utility, given its performance (e.g., it achieves perfect accuracy in small datasets). In the future, we intend to extend this work to other mobile sensing tasks, as well as extending the evaluation using other (bigger) datasets.

### Acknowledgment

This research is supported by the National Key Research and Development Program of China (2016YFB1000201). The authors would like to thank all the contributors for the open-source tools and datasets used in this work.

### References


