Robust Single Accelerometer-based Activity Recognition Using Modified Recurrence Plot

Jianjie Lu and Kai-yu Tong, Senior Member, IEEE

Abstract — Using a single 3-axis accelerometer for human activity recognition (HAR) is challenging but attractive in daily healthcare and monitoring with wearable sensors and devices. In this paper, an effective and efficient framework is proposed to address the recognition problem without any heavy preprocessing. We encode 3-axis signals as 3-channel images using a modified recurrence plot (RP) and train a tiny residual neural network (ResNet) to do image classification. The modified RP is first proposed in our paper to overcome its tendency confusion problem, which has improved our system performance significantly. We evaluate our algorithm on a new database and a public dataset. Results show that our recognition framework achieves highly competitive accuracies and good efficiencies with other state-of-the-art methods on both datasets. Moreover, our method shows stronger robustness to noise and low decimation rate through comparison experiments. Finally, we provide detailed discussion and analysis of our approach from two perspectives: the pattern analysis of encoding algorithm and the interpretation of classification model.

Index Terms — Accelerometer, activity recognition, deep learning, recurrence plot, signal recognition.

I. INTRODUCTION

DAILY monitoring and healthcare are becoming increasingly popular due to the proliferation of personal electronic devices like smartphones and wearable wrist bands. Accurate and effective daily activity recognition is an important procedure for health condition analysis. One of the most common embedded sensors is 3-axis accelerometer, which can provide useful information for activity recognition. Our paper intends to address multi-activity recognition problem with a single accelerometer.

A large amount of work has been published on this topic [1], [2]. Although differentiating walking from other activities is simple, it gets more difficult as more realistic classes of daily activities are included. Some literatures combine multiple accelerometers or other sensors like gyroscope [3] and camera [4] to enhance accuracies. However, many wearable devices only integrate a single accelerometer. Moreover, in most cases, the orientation of accelerometer is not fixed. These result in large variance and noise in the database and make the recognition challenging.

In this paper, we encode 3-channel accelerometer signals as color images using a modification of recurrence plot (RP) and train a customized residual neural network (ResNet) for recognition. Converting time series recognition problem into image classification problem has been proposed by some literature. Gramian Angular Field (GAF) and the Markov Transition Field (MTF) are presented in [5] for one channel series. Its computation efficiency might be a problem when applied in real-time motion or action recognition. They also do not give an encoding solution for 3-dimensional time series such as 3-axis accelerometer signals, where the correlation information is quite important. RP is originally applied to the analysis of dynamic systems. Though RP can be used to depict the state trajectories, it confuses the tendency of time series which is discovered in our simulation. This problem limits the performance of recognition seriously. We first modify it to improve our model’s performance. Experiments show it can characterize the curve shape in time domain and the correlation among different channel signals as well. As deep learning has been proven to be successful in image classification tasks, we design a tiny ResNet aiming not only to obtain satisfactory results in small size datasets but also to achieve high efficiency.

Our approach is compared with some state-of-the-arts on both a new database and a public dataset named ADL dataset. It shows that our method has achieved competitive accuracies and good computation efficiency. In addition, comparison experiments are also implemented under different noise level and sampling rate settings to test the robustness of different methods. At the end, we present detailed explanations and analysis on the rationale of our method.

The rest of the paper is organized as follows: Section II discusses related work on motion recognition algorithms using a single 3-axis accelerometer sensor. Section III introduces an overview of the proposed recognition framework and describes how a modification of RP is utilized to encode 3-axis signals as color images and how the tiny ResNet is designed to classify acceleration traces. In Section IV, we evaluate our system’s performance and provide detailed analysis in the experimental results. Section V further explains our algorithm. Finally, Section VI concludes the paper.

II. RELATED WORK

Human activity recognition (HAR) on data from a single 3-axis accelerometer can be treated as time series classification, which has been considerable in the past. The existing methods may be categorized into three groups according to input types.

The first methods rely on heuristic handcrafted features, including statistics (mean, standard deviation, minimum, maxi-
The drawback is caused by the computation of similarity matrix among large datasets, which makes it fall in speed predicament although there are some speed-up DTWs [18]. In addition, convolutional neural network (CNN) can operate on raw signal directly [19], [20], [21]. It captures salient patterns at different time scales, while large size dataset is required for training. Besides, recurrent neural networks (RNN) are applied for time series analysis successfully. Reference [22] proposes a model including a CNN and a RNN where features are fused at the last layer to enhance the performance.

Unlike the former two kinds of methods, the last encodes time domain signals as images so as to take full advantage of the powerful ability of deep learning architectures in computer vision to learn features or patterns [23], [24]. The former uses polar coordinates to encode time series as two types of images, Gramian Angular Fields (GAF) and Markov Transition Fields (MTF). It requires relatively heavy computation, which makes it difficult to be applied in real-time motion or action recognition. Recurrence plot (RP) has been empirically proven to have excellent performance [14], [15]. Clustering algorithms using DTW as a distance metric are employed to find exemplars and traditional classifiers are used for signal recognition [16], [17].

Fig. 2. Recurrence plots of typical signals (top row: time series; bottom row: corresponding recurrence plots; from left to right: white noise, harmonic oscillation with two frequencies and linear drift curve).

### III. Approach

#### A. Overview

The proposed action recognition framework, as shown in Fig. 1, consists of two stages. First, we project each channel signal as a square matrix using modified RP respectively and combine them into a color image. After normalization, we design a tiny ResNet to do the recognition task end to end. Notice that our framework is simple and no heavy preprocessing, such as average filtering and features selection, is needed. Next, we will introduce the encoding algorithm and the architecture of our neural network.

#### B. Recurrence Plot

Recurrence plot (RP) is originally a visualization tool to study complex dynamic systems. It is first proposed in [25]. References [26], [27] introduce its principles and applications in detail. Recurrence is a concept for analyzing nonlinear data points on phase space trajectories of a dynamical system, whose states are typically in a rather intricate manner. RP contains typical small-scale features including dots and lines. The large-scale textures can be visually represented by homogenous, periodic, drift and disrupted. Fig. 2 depicts how RP can characterize signal features well.

We denote vectors by bold lower-case letters, e.g., \( x \), and matrices by bold upper-case letters, e.g., \( R \). The RP can be formally expressed by matrix \( R \) given a trajectory data sample \( x = [x_1, \ldots , x_N] \) as follows:

\[
R_{i,j}(\varepsilon) = \Theta(\varepsilon - ||x_i - x_j||), \quad i,j = 1, \ldots, N
\]

where \( \varepsilon \) is a threshold and determines the number of states, \( \Theta \) is the unit step function and \( || \cdot || \) is a norm. An appropriate norm must be selected to compute a recurrence matrix. The most common used norms are the \( L_1 \) norm, the \( L_2 \) norm and the \( L_\infty \) norm [28]. In our paper, we use \( L_2 \) norm because we skip the threshold step to avoid curve shape loss.
We resort RP to encoding 3-axis signals as RGB channels of images so that their correlation information can be exploited. Mathematically, given a segment of motion data $D = \{x, y, z\}$, where $D \in \mathbb{R}^{3 \times N}$. Each row component in $D$ is a vector, whose elements represent the acceleration in the $x$-, $y$-, $z$-direction correspondingly. As our focus is on the formulation of Cartesian coordinate system while for those in downhill tendency, the state difference vector falls in the third quadrant. Fig. 4 illustrates the correlation and tendency (top row: raw signals, where red, green and blue represent $x$-, $y$-, $z$- channel signals; bottom row: corresponding modified recurrence plots which are asymmetry).

Finally, accelerometer signals have been encoded as images. Note the diagonals of recurrence matrix are zero. The minimum of $M$ should be computed after repelling zero value. Fig. 4 shows that our encoding method can capture the correlation among $x$-, $y$-, $z$- channel signals by colors and their tendencies by its asymmetry as well.

E. Recognition Using Tiny ResNet

Considering the generalization ability and computation efficiency, we design a tiny ResNet. The exact network configurations are shown in Table I. We construct it by stacking residual blocks with bottlenecks, which is proposed to solve the problem that deeper models are more difficult to optimize and achieves decent accuracies in image classification tasks [28]. Another reason for using residual module is that the number of learning parameters tends to be less than other...
TABLE I
NETWORK ARCHITECTURE FOR CLASSIFICATION.

<table>
<thead>
<tr>
<th>Layers</th>
<th>#Kernel/Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution 3 × 3</td>
<td>32 × 3</td>
</tr>
<tr>
<td>Max Pooling</td>
<td>-</td>
</tr>
<tr>
<td>Residual Module</td>
<td>32 × 2</td>
</tr>
<tr>
<td>Residual Module</td>
<td>32 × 2</td>
</tr>
<tr>
<td>Max Pooling</td>
<td>-</td>
</tr>
<tr>
<td>Residual Module</td>
<td>32 × 2</td>
</tr>
<tr>
<td>Max Pooling</td>
<td>-</td>
</tr>
<tr>
<td>Residual Module</td>
<td>32 × 2</td>
</tr>
<tr>
<td>Max Pooling</td>
<td>-</td>
</tr>
<tr>
<td>Global Average Pooling</td>
<td>-</td>
</tr>
<tr>
<td>Fully Connected Layer</td>
<td>32 × #class</td>
</tr>
</tbody>
</table>

Fig. 5. Modified RP images of samples from each label (from left to right: walking, sitting, standing, lying and squatting) in ASTRI Motion dataset.

deep learning architectures. In other words, it might require relatively small dataset for training. No convolutional layer with kernel size greater than 3 is used, since small filters still can obtain large receptive field which reduce the inference time. All convolutional layers have 32 kernels, which is enough to extract features and contributes to very fast recognition. Batch normalization is applied after each convolutional layer. At the end of the model, global average pooling is applied for limiting the number of parameters and a fully-connected layer with softmax is used for class prediction.

Data augmentation is applied considering the size of dataset. Instead of rotating/flipping input images, we use a small range of image random cropping (0.8-1.2) and jitter on normalized pixel values (±0.01) to increase the amount of data. Since rotating/flipping in the RP image domain will distort the time domain signal, which is obviously unreasonable. In contrast, random cropping and pixel jitter are equivalent to adding a varying window and random noise in the time domain. Finally, we obtain 16 times data.

We train the network using stochastic gradient descent (SGD) with momentum 0.9, batch size 10 for 200 epochs. The learning rate is initialized at 0.001. Fig. 5 provides input image samples from each label in ASTRI Motion dataset. Each image is generated from a raw 3-axis accelerometer signal sample by applying the modification of RP. The features of data samples from different labels can be depicted by the image patterns including color, textures and their asymmetry attributes show tendencies of the signals. More visualization of these datasets can be found in our open source link https://github.com/lulujianjie/HARusingModifiedRP.

IV. EXPERIMENTS

A. Datasets

Experiments are conducted on two datasets. The ASTRI dataset is provided by Hong Kong Applied Science and Technology Research Institute (ASTRI). To evaluate the generalization ability of our model, we compare it with other state-of-the-art algorithms on a public dataset named Activities of Daily Living (ADL) from UCI Machine Learning Repository [29].

1) ASTRI Motion Dataset: This dataset is about five types of human’s hand movements. Data are collected from a single accelerometer configured on the smart wrist band created by ASTRI. In this dataset, 11 subjects of different genders and ages conduct activities including walking, sitting, standing, squatting and lying. Each subject performs several times. The wrist band can be worn on either left or right hand and has high sampling rate (52Hz). These may contribute to the orientation variant and noisy signals. The total number of samples is 1081 including 321 walking, 191 standing, 189 squatting, 193 sitting and 187 lying.

2) ADL Dataset: It is a public wrist-worn accelerometer dataset, which records totally 16 volunteers performing different daily activities. Data are collected by a single 3-axis accelerometer at sampling rate 32Hz. There are 689 raw data samples used in our experiments. Including 102 climbing stairs, 96 drinking water, 101 getting up bed, 100 pouring water, 96 sitting down, 95 standing up and 99 walking.

B. Experimental Settings

We compare our approach with the following seven baselines, namely RF, SVM, CNN, DTW+1-NN, DTW+Clustering, LSTM+FCN and RP+RestNet. Among them, the first six methods show the state-of-the-arts results on classification of time series data. RP+ResNet is used to evaluate the improvement of our encoding algorithm, which utilizes original RP algorithm and the proposed tiny ResNet.

1) RF [6]: A random forest (RF) classifier is exploited to detect activities based on the wrist motions. They use a total of 21 statistical features in the time and frequency domains and achieve the highest average accuracy of 89.6 % with a forest of 64 trees.

2) SVM [7]: A support vector machine (SVM) with a radial basis function kernel is used as a classification model. The features used are the same as RF.

3) CNN [21]: In this baseline, a convolutional neural network (CNN) operates on batches of raw signal samples directly by taking input as 3-channel data. ReLU and normalization are used in the first four layers.

4) DTW + 1-NN [13]: Dynamic time warping (DTW) is used to compute the distance between training samples and test samples. After getting the distance matrix of time series, k-NN is applied to perform classification. Specifically, we select 1-NN as the baseline after comparing models with different k values since it obtains the best results.
large variance within class, which is caused by independent ASTRI Motion dataset. The possible reason may lie in the satisfactory results in other actions, for example walking in accuracy on some actions like standing. But it does not obtain performance on ASTRI Motion dataset is slightly better than size. It also shows the overfitting problem that the recognition vulnerable to signals at different conditions. CNN achieves average accuracy, which is 2.5/2.0% higher than [7] under the condition that the same subjects are involved in testing set. User-independent setting refers to the test data are from subjects who are not involved in the training set. In user-mixed experiments, we split dataset as training set and testing set according to 7:3. In user-independent experiments, data of 4 (for ASTRI Motion dataset) or 5 subjects (for ADL dataset) are selected randomly for testing. Besides, we implement cross-validation experiment for 5 times and record results considering statistical significance. For better application in practice, we do not label the left or right hand, age and gender of subjects before recognition. A 5-second window is applied in our experiments since this time duration should be sufficient for subjects to finish actions.

C. Results on Recognition Accuracy and Efficiency

The quantitative comparisons of our framework and the other six baseline methods on ASTRI Motion dataset and ADL dataset are shown in Table II and Table III respectively. The best performance for each class is highlighted in bold.

From the experimental results, our method obtains the best average performance than the others on both datasets. Specifically, our method outperforms in 2.4 percentage points comparing with the second best approach [7] on ASTRI Motion dataset. On ADL dataset, our model achieves 90.9/88.7% average accuracy, which is 2.5/2.0% higher than [7] under user-mixed/user-independent settings. In comparison, methods using handcraft features and traditional machine learning models, RF and SVM, achieves decent results, though they are vulnerable to signals at different conditions. CNN achieves plain results in both datasets, though its training accuracy nearly achieves 100%. This is mainly caused by small dataset size. It also shows the overfitting problem that the recognition performance on ASTRI Motion dataset is slightly better than that on ADL with less samples. DTW+1-NN performs the best accuracy on some actions like standing. But it does not obtain satisfactory results in other actions, for example walking in ASTRI Motion dataset. The possible reason may lie in the large variance within class, which is caused by independent subjects. They differ from one person to another and even the same subject cannot perfectly replicate the same action. For some samples from different actions, they even have similar patterns in time domain. As a result, DTW does not work effectively. The same issue exists in DTW-based method even using clustering. Another problem may be from the clustering algorithm, which does not guarantee that all members of a cluster have the same label as their exemplar. Moreover, the exemplars tend to have unbalanced distribution in terms of labels, which is observed in our experiments. For instance, 166 exemplars are learned while there are only 16 exemplars from walking samples. LSTM+FCN also achieves high accuracy on both datasets as LSTM shows great success in times series analysis. After modifying RP, we achieve over 7/10% improvement in the recognition performances considerably by comparing proposed method with RP+ResNet on ASTRI/ADL dataset. Although same networks are implemented for classification, our method achieves better accuracies in four of five classes of ASTRI Motion dataset and all classes of ADL dataset than RP+ResNet.

We also compare the efficiency. Experiments are conducted on a platform, which has an Intel i5-7500, 8G RAM and a GTX1060 6G GPU. We use fast version of DTW algorithm, since original DTW requires heavy computation cost. The prediction time of 100 samples are listed in Table IV. Each instance is in 5-second. Our framework takes about 86 seconds (CPU) or 7 seconds (GPU) to inference 500-second signal segments, which is competent in real-time recognition task.

D. Comparison Experiments on Robustness

The environment in which wearable devices are used is often complex, especially with much interference, such as power frequency noise. Recognition algorithms need to be robust to noise. In addition, in order to enhance recognition efficiency, a common practical method is to reduce the number of data points in a signal segment. The fewer data points in the segment used to calculate the feature, the faster the algorithm recognizes the signal class. Note that the decimation is not downsampling at the hardware level, but to use partial data points obtained by hardware sampling when calculating features. It is a software level downsampling for efficiency.

Therefore, we carry out experiments for comparing different methods under varying noise and sampling rate settings. Specifically, we assume that the noise is subject to a Gaussian distribution, and the recognition accuracy is tested by adding noise of different variances, i.e. \( \sigma = 0.05, 0.1, 0.2 \) to the original signal. Then we also test the performance of algorithms by operating downsampling with various rates, i.e. \( M = 0.8, 0.7, 0.6 \) on the original signals.

Fig. 6 shows the results under different noise levels on ASTRI Motion dataset and ADL dataset respectively. It can be seen that as the noise intensity increases, the performance of algorithms gradually deteriorates. By comparison within the bar cluster, the accuracy of our recognition framework reduces marginally, showing stronger robustness to noise. Our encoding algorithm essentially converts phase changes in time domain into images. Each pixel on the image actually represents the magnitude and direction of the phase.
change of the signal. Hence, we can apply unique image data augmentation, jitter on normalized pixel values, to train the network. An empirical analysis on the importance of data augmentation is provided in next part. In contrast, traditional recognition algorithms based on handcraft features, e.g. RF and SVM, achieve worse performance when adding noise, though they have obtained decent accuracies without noise. Another interesting observation is that they seems to be more vulnerable than those methods using deep learning techniques including CNN, LSTM+FCN and our method.

Fig. 7 depicts the comparison with different decimation rates on both datasets. Less data points in the segment cause lower recognition accuracy. When the decimation rate is reduced to 0.6, the performance of each algorithm has been greatly weakened, which is more pronounced on the ADL dataset (32 Hz). It is worth noting that our algorithm has less performance degradation than other algorithms, because our hierarchical model can fuse features at different scales. Besides, our another data augmentation method is random crop, which is equivalent to adding varying windows in time domain. All of these may contribute to a stronger robustness of our method.

### E. Ablation Study

We provide an ablation study to evaluate and analyze the influence of data augmentation. Experiments are implemented on ASTRI dataset. Table V reports the comparison results of our method without and with data augmentation under user-mixed/user-independent settings. Overall, data augmentation can improve recognition accuracy. In particular, as shown in
the first row of Table V, the performance of our model increased by about 1 percentage point using data augmentation. By horizontally comparing the data from the second row to the fourth row, it shows that the accuracy of the model can be increased by up to 5 percent by data augmentation in the case of Gaussian noise. This also demonstrates the importance of data augmentation against noise. In contrast, the improvement of Gaussian noise. This also demonstrates the importance of the fourth row, it shows that the accuracy of the model can be increased by about 1 percentage point using data augmentation.

**B. ResNet Interpretation**

We do visualization to identify which part of the time series contributes to the recognition results. The proposed classification network essentially maps a color RP image to class scores. For ASTRI Motion dataset, our model outputs five values for each motion class. The weights of fully connected layers will learn to assign high scores to the correct class. Before flowing into the classification layers, data (feature map) is in 2-dimension. Therefore, we can visualize them to see which part of inputs contribute to the classification results. A method named class activation map (CAM) is proposed in [30] to produce heatmaps for localizing the most indicative areas of the input. We use $h_i$ to denote the $i$-th feature map and $w_{c,i}$ denote the weights in the fully connected layer for $h_i$ leading to class $c$. There are totally 32 feature maps as illustrated in Table I. The CAM for class $c$ is formulated as follows:

$$H_c = \sum_{i=0}^{31} w_{c,i} \cdot h_i$$

where $h_i$ is a $14 \times 14$ matrix and $w_{c,i}$ is a scalar.

We follow this approach and slightly modify it. Since we observe that when weight $w_{c,i}$ is less than zero, the corresponding $h_i$ has a negative effect on the class scores. Therefore, we only visualize the features maps that have positive weights. We name this method as positive class activation map (PCAM), which has the formulation as (9).

$$H_c = \sum_{i=0}^{31} max(0, w_{c,i}) \cdot h_i$$

Fig. 8 shows the PCAMs of five classes in ASTRI Motion dataset. For walking, the ResNet seems to capture the recurrent textures, i.e., periodicity in time series data. For squatting, the ResNet sketches the slight quasi-periodicity and the phase state change over a period of time. Interestingly, the PCAM of lying shows a clear highlight vertical lines, which depicts the model mainly distinguish this motion by the transition between two stationary states. The PCAMs of sitting and standing are

**A. Pattern Analysis of Modified RP**

Temporal features can be characterized by the textures or patterns after encoding accelerometer signals as images use modified RP. The following presents various patterns and the correspond meanings.

1) Homogeneity in RP shows that there is random noise existing in the time span. See the first column in Fig. 2.
2) Cyclical recurrent textures indicate the quasi-periodicity in time domain. See the second column in Fig. 2.
3) Fading from the diagonal line to corners reveals the time series data has a tendency. See the last column in Fig. 2. However, the trend cannot be distinguished as going up or down, which is the tendency confusion problem.
4) Horizontal or vertical lines depict there is a sharp change of phase state existing in the data. According to equation (2), lines occurs if the state difference vector $\delta^{m} - \delta^{n}$ remains the same given $\delta$. In this case, the norm of $\delta^{m}$ should be much bigger or smaller than the others. If we convert this state space into time domain space, Horizontal or vertical lines reveal that a significant relationship between a certain piece of time series data and the entire data example.
5) Colors in RP characterize the signal correlation among different channels. See Fig. 4. Asymmetry is an important property of the proposed modified RP, which reveals the tendency direction. See Fig 4, equations (5) and (6).
asymmetric, which illustrate the tendency plays a key role in the classification decision. Particularly, there are also highlight area existing in the edge of the PCAM of sitting. It reveals the significance of state relationship between the initial period and the whole duration in distinguishing this class.

VI. CONCLUSION

In conclusion, we propose an effective and efficient activity recognition framework using a single accelerometer. Our method encodes accelerometer signals as color images using a modified RP, which is first proposed to solve its tendency confusion problem. After obtaining images of time series, we use unique data augmentation and train a customized ResNet for recognition end-to-end. Finally, we test and evaluate our method on two datasets. Experimental results have shown that our method achieves highly competitive recognition accuracies, satisfactory efficiencies and strong robustness when compared to the other state-of-the-art methods. It shows good potential for real-time 3-axis accelerometer signals recognition in small size datasets.

REFERENCES


