A GAN-based data augmentation method for human activity recognition via the caching ability

Junhao Shi, Decheng Zuo*, Zhan Zhang

Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China

shijunhao@hit.edu.cn, *zuodc@ftcl.hit.edu.cn, zz@ftcl.hit.edu.cn

No. 92, Xidazhi Street, Nangang District

Abstract: Sensor-based human activity recognition (HAR) provides vital information for an ocean of health care and entertainment applications. However, data acquisition of such studies usually requires huge time and manpower, which delays the progress of the whole project. Data augmentation is a promising way to solve this issue. In this paper, we expand the origin dataset via a GAN-based augmentation neural network. The experimental results suggest that the generated data has a certain substitution and complementary effect in terms of the real data based on the caching deployment ability. In addition, the quasi-real data also gains good performance on identifying certain activities.

Keywords: Data augmentation; Human activity recognition; GAN; Deep learning; Caching ability

1 Introduction

As a major research scope of pervasive computing, Human Activity Recognition (HAR) has been applied in various fields such as medical treatment, education and entertainment [1]. Moreover, as the popularity of smart phone in daily life, its high integration of sensors brings blossom for sensor-based HAR in many areas. The main goal of most HAR systems is to improve the recognition accuracy of simple/complex activities in different scenarios. The complicacy of both the scenarios and activities brings challenge to the HAR systems.

Most relevant research takes advantage of machine learning algorithms as classifiers. However, with the rapid development of deep learning (DL) models, a host of studies have focused on building suitable DL models for HAR. According to existing efforts in other fields, DL methods have achieved excellent results in a multitude of areas such as NLP or image processing. However, all this characteristics require large amount of data. In many studies, data collection consumes a multitude of time and human resources. Existing HAR datasets are in small scale, which is not enough to sufficiently train DL model.

In order to solve the problem of insufficient data, we introduce Generative Adversarial Nets (GAN) for data augmentation. As a special DL model, GAN is of great significance in AI fields [2]. The fundamental idea of GAN is based on two-person zero-sum game. Its structure includes a generator and a discriminator. When comes to training stage, generator and discriminator confront with each other to learn. The generator tries to generate data, which has similar distribution with origin data, to make the discriminator confused if the data is real or generated, while the discriminator judges its input if the data is real or not. The goal of training stage is to achieve a Nash equilibrium between the generator

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1002/itl2.257

and the discriminator. Data generation is the most intuitive application of GAN, which has overwhelmingly interesting applications in a host of scopes [3, 4]. However, the application of GAN in sensor-based HAR is limited. Wang et al. [5] propose a sensoryGANs, which is meanly focusing the motion data generation. They only consider 3 activities in their work, and here we enlarge the activity scope. In our work, we take advantage of this feature and propose a HAR data augmentation method, which is named HARAug-GAN.

2 Methodology

2.1 Structure of HAR system

In this section, we give an overview of HAR systems. Fig. 1 shows the basic process of HAR.

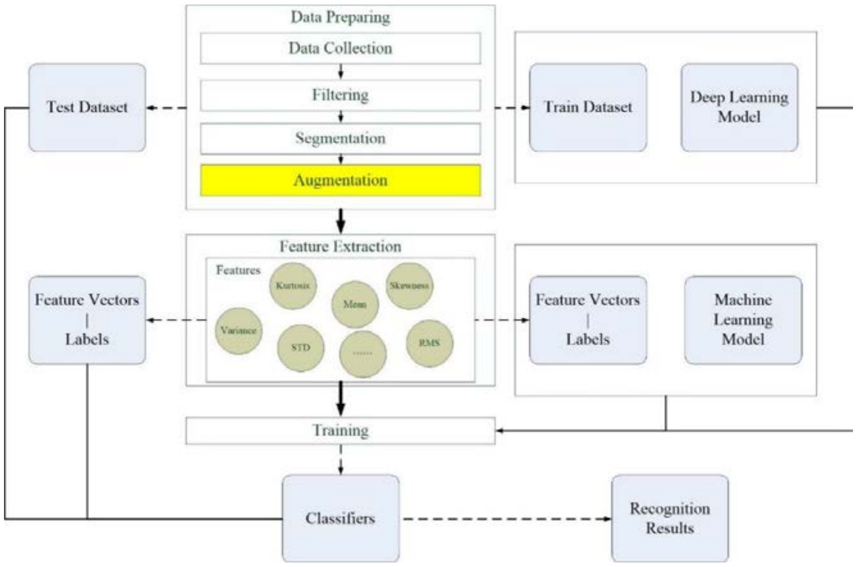


Fig 1 Basic process of HAR

In the *Data Preparing* stage, we use hardware platform, which is basically smartphone, to collect motion data from subjects. Signal noise may appear during this phase. We smooth the origin data through the filter. Here, a Butterworth low-pass filter is utilized to remove the data noise. After filtering, sliding window algorithm is utilized for data segmentation to a valid size of 125*3, which is enough for each single activity. Then we divide the data into two parts, one for train and the other for test. Due to that the whole data volume may be less enough for training DL model, data augmentation is used for expand the data set. In the feature extraction stage, we extract statistical features (e.g. mean, variance, etc.) to train the machine learning model. However, deep learning can be directly trained with the data input. After training, the model can be used for recognition.

2.2 Data Augmentation

The purpose of data augmentation is to generate “real” data. It is supposed to have same distribution as origin data but somehow with different details, which makes the classifiers more robust and generalized.

The data augmentation in our work is mainly realized through GAN, which consists of two parts: generator and discriminator. The generator produces fake data based on true data input. The generated data is distinguished if it is real through discriminator. During the GAN training, the generator constantly updates its parameters according to the feedback of the discrimination results, which helps generated data to be more real. After several epochs of training, the two networks finally achieve a balanced result: the generator produces fake data which is similar to the real activity data, while the discriminator is confused by the generated data and the overall recognition accuracy is around 50%.

We build the HAR-Aug GAN which refers to the structure of DCGAN [6]. The HAR-Aug GAN is composed of two parts: generator and discriminator. The origin data size gets expanded into 60*60*25 through 4 deconvolution layers. Then 3 convolution layers and 1 reshape layer changes the data back to 3*125, then the fake data is produced by generator. The generated data needs to be classified by discriminator. According to the formal version of DCGAN, the model structure of discriminator should be deeper than we design. However, given that our activity data is low-dimensional, we simplify our discriminator, which consists of only 2 convolution layers and 2 full connected (Dense) layers. The classification ability of discriminator will affect the data generated by generator. If we utilize a more complex structure, the classification result could be extremely perfect for generator, which may cause generator to give up generating appropriate data.

We elaborate the operations in the net. We here use X to present input data of the CNN. After convolution, the element in the kth feature map $E_{p,q}^{k}$ can be denoted as follows:

$$E_{p,q}^{k} = F(b^{k} + \sum_{\alpha=1}^{i} \sum_{\beta=1}^{j} \omega_{\alpha,\beta}^{k} X_{\alpha,\beta}^{p*(m-2)+q}) \tag{1}$$

Where p and q represent the location of $E_{p,q}^{k}$ in the feature map, b^{k} is the bias term of the feature map. Convolution kernel size is expressed as $i*j$. $\omega_{\alpha,\beta}^{k}$ represents the weight of the convolution kernel and filter index α and β. F is the Rectified Linear Unit (ReLu) activation function.

$$F(x) = max(0, x) \tag{2}$$

The deconvolution is almost operating in the same way as convolution, except that deconvolution layer takes the stride parameter for expanding the input data with embedding 0, and uses transposed kernel for convolution.

The Dense layer is a fully connected layer and its output can be expressed as follow:

$$v_{\gamma} = S(\sum_{\gamma=1}^{u} \boldsymbol{W} x_{\gamma} + b) \tag{3}$$

Where u represents the dimension of the input data, x_{γ} is the γth input and v_{γ} the γth output. $\boldsymbol{W}$ is the weight vector. b is the bias value. S is the Tanh activation function which can be expressed as follow:

$$S(x) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{4}$$

2.3 Dataset, Feature Extraction and Machine learning classifier

In our work, we use a public HAR dataset as original input. This dataset [7] covers 19 activities. Each activity is performed by eight subjects (4 female, 4 male, between the ages of 20 and 30) for 5 minutes. There are 5 device locations for data acquisition including torso, right arm, left arm and right leg, left leg. In our work, we only focus on single sensor system, so we select the left leg data, which has high identification for different activities according to previous results in another work of ours, as the source data. After filtering and segmentation, each data block size is 125*3 because we only utilize the accelerometer data. We use A1-A19 to represent the activities in the dataset, including sitting, standing, lying, et al. More information can be found in Reference [7]. The data are collected via 9-axis inertial sensing module (3D Accelerometer, 3D gyroscope, and 3D magnetometer).

Meanwhile, we take the support vector machine (SVM) as representative of machine learning algorithm for comparison. We use 9 commonly used statistical features, including mean, variance, standard deviation, information entropy, root mean square, skewness, kurtosis, energy and zero-crossing rate to form 27-dimension feature vectors to train SVM.

3 Experiment Results

In this part, we introduce the process of GAN training. We generate the activity data separately for each activity. The expected data is similar with the origin data while with some difference. This feature can be beneficial to the training of the model. We train the GAN for 200 epochs. Fig 2 shows the loss curves of 2 activities. The pattern of other loss curves is pretty similar, and we choose not show them here. In order to test whether the generated data can make supplement and replacement for the real data, we use the generated data and the real data for activity recognition experiment in this part. During the data generation, the client frequently reads and writes to the server. We implement the caching ability via the deployment of three CDN servers [8] to accelerate the data reading and writing. Using CDN technology can make it easy for clients to download and use new data to update the model, which accelerates the efficiency of model iteration.

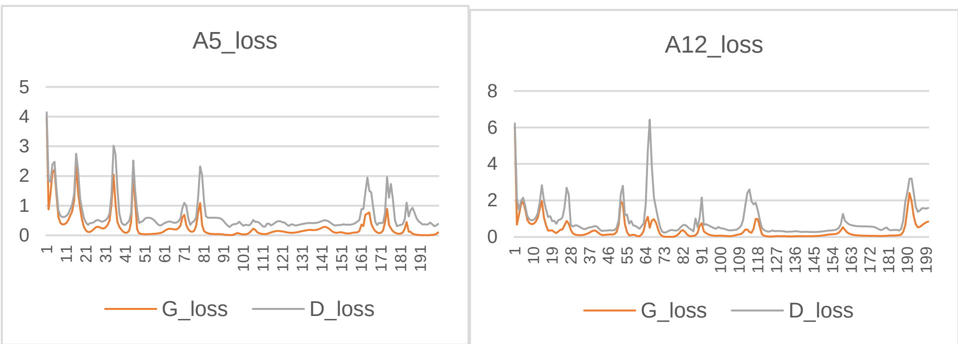


Fig 2 Training loss of 2 presentative activities (A5, A12)

First, we separate the true dataset into 10 parts and utilize 10-fold cross validation. We take DL and SVM as the classifiers in our experiment. The structure of our DL model is depict as Fig. 3.

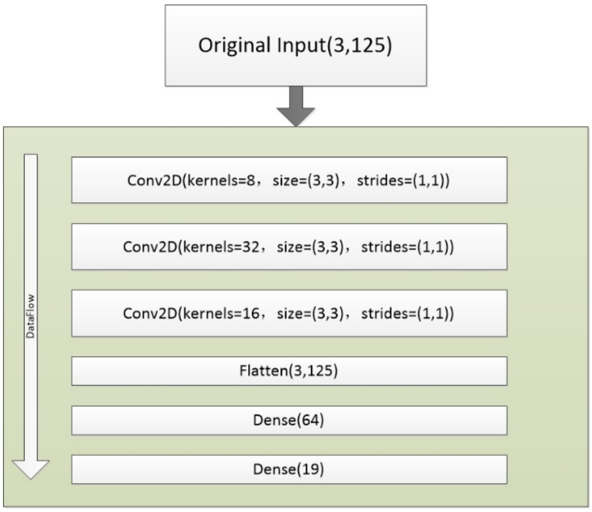


Fig 3 Structure of deep learning model as classifier

The recognition result is given in Fig 4. As Fig 4 shows, both classifiers are well trained using true data. Still, they perform less outstanding on certain activities. For instance, the accuracy of DL on A07 and A10 is inferior to 70%, while SVM on A08 and A09 is unable to reach 60%. We believe this is mainly due to these activities are easily confused with others.

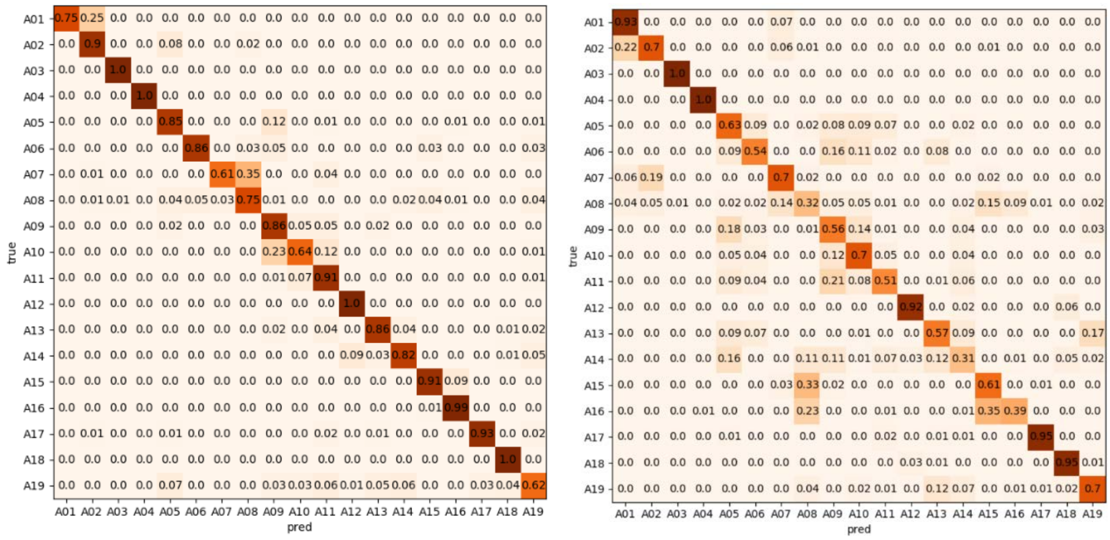


Fig 4 Recognition result with True data as train set (Left:DL Right:SVM)

We then replace the training set with same amount of generated data, without changing the test set to conduct the experiment. We desire to observe the fitting degree of the generated data to real data through this test. The result is given in Fig 5.

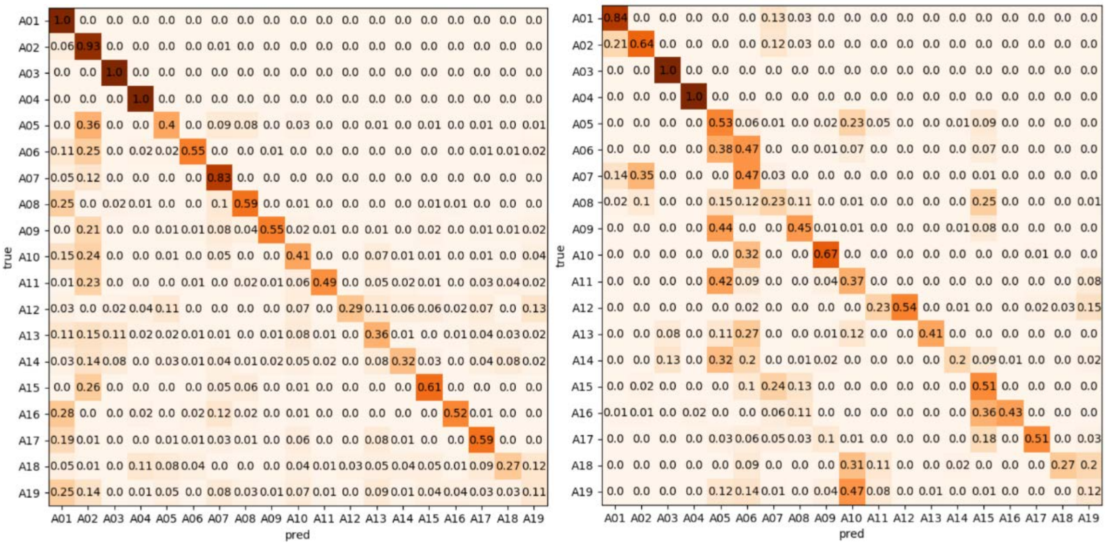


Fig 5 Recognition result with Generated data as train set (Left:DL Right:SVM)

According to the results, the generated data as a train set alone is unable to yield a superior performance. However, some activity can be partially fitted by our generated data like A01, A04, and A07.Finally, we use a slight part of real data mixed with generated data at a ratio of 1:10 as the training set. The experiment result is shown in Fig 6.

The result in Fig 6 is promising considering that overall recognition accuracy is close to Fig 4, which means that the generated data and the real data are somehow complementary. Accuracy on certain activities is promoted such as DL on A07, which has a 20% accuracy promotion. However, there are also activities suffering negative effects brought by mix train set such as DL on A11.According to the overall results, SVM is greatly improved by the mix set, the overall result is promoted from 68.33% to 81.72%, while slight promotion appears on DL method by about 2% (from 85.64% to 87.88%). We believe that the true data brings useful details to the model.

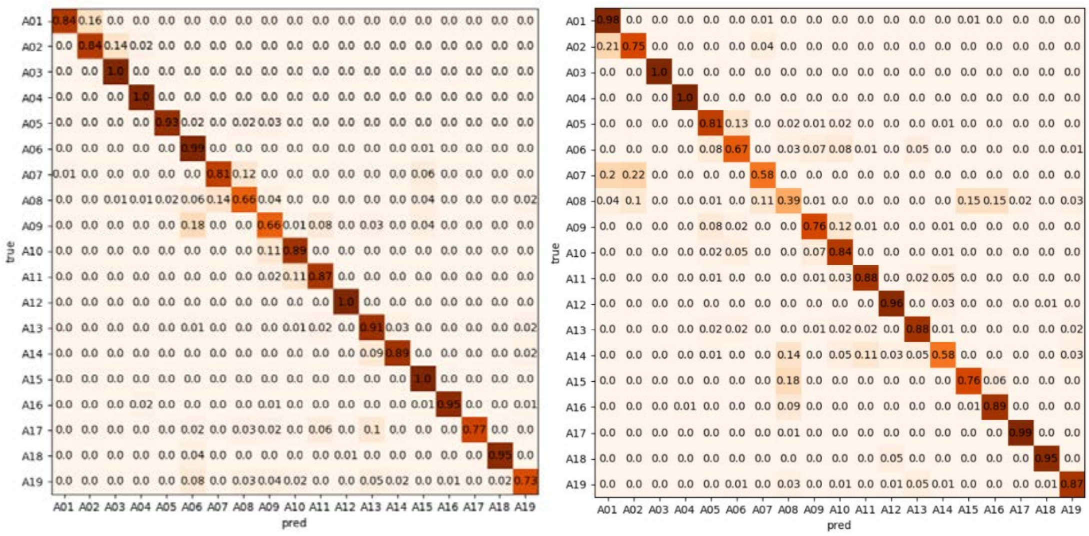


Fig 6 Recognition result with Mix data as train set (Left:DL Right:SVM)

4 Conclusions

In this paper we augment a public dataset through a GAN-based neuro network. Various experiment results suggest that the quasi-real data generated by our HARAug-GAN can supplement or even replace the origin real data. However, the quasi-real data alone cannot achieve outstanding performance. The mix train set consist of real data and generated data is more robust in our work.

Acknowledgment

This work is supported by the National Science Foundation of China (Grant No. 61370085). In addition, we thanks Qingyi Zhang for the experiment deployment on CDN servers.

References

[1] Akshay Uttama Nambi S.N., et al, CoachMe: Activity Recognition using Wearable Devices for Human Augmentation, International Conference on Embedded Wireless Systems and Networks, 2017: 174-179.

[2] Yu T., et al, CR-GAN: Learning Complete Representations for Multi-view Generation, 27th International Joint Conference on Artificial Intelligence, 2018, 942-948.

[3] Zongyong Cui, et al., Image Data Augmentation for SAR Sensor Via Generative Adversarial Nets, IEEE Access PP.99,2019, 1-1.

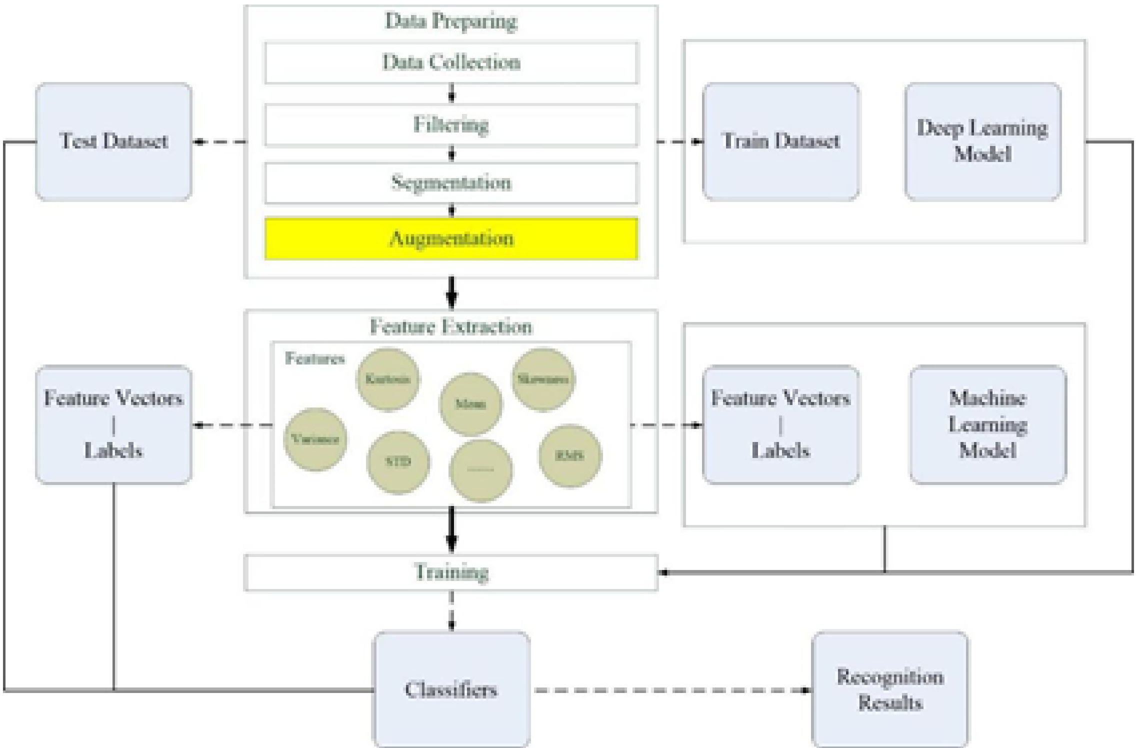
[4] Ramachandra, Gautam, GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks, ArXiv 2017.

[5] Wang J., et al.,SensoryGANs: An Effective Generative Adversarial Framework for Sensor-based Human Activity Recognition, 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-8.

[6] Lu D., et al, Mobile Online Activity Recognition System Based on Smartphone Sensors, International Conference on Advances in Information and Communication Technology, 2016: 357-366.

[7] Altun K., et al, Comparative study on classifying human activities with miniature inertial and magnetic sensors, Pattern Recognition, 2010, 43(10): 3605-3620.

[8] Cui S., et al, Multi-CDN: Towards Privacy in Content Delivery Networks, IEEE Transactions on Dependable & Secure Computing, 2020, 17(5): 984-999.



ITL2_257_Graphical Abstract Image.png