ResNet-Like CNN Architecture and Saliency Map for Human Activity Recognition

Zixuan Yan[1(✉)], Rabih Younes[2], and Jason Forsyth[3]

[1] Nanjing University, Nanjing, China
171180578@smail.nju.edu.cn
[2] Duke University, Durham, USA
[3] James Madison University, Harrisonburg, USA

Abstract. Human activity recognition (HAR) has been adopting deep learning to substitute well-established analysis techniques that rely on hand-crafted feature extraction and classication techniques. However, the architecture of convolutional neural network (CNN) models used in HAR tasks still mostly uses VGG-like models while more and more novel architectures keep emerging. In this work, we present a novel approach to HAR by incorporating elements of residual learning in our ResNet-like CNN model to improve existing approaches by reducing the computational complexity of the recognition task without sacrificing accuracy. Specifically, we design our ResNet-like CNN based on residual learning and achieve nearly 1% better accuracy than the state-of-the-art, with over 10 times parameter reduction. At the same time, we adopt the Saliency Map method to visualize the importance of every input channel. This enables us to conduct further work such as dimension reduction to improve computational efficiency or finding the optimal sensor node(s) position(s).

Keywords: Human activity recognition (HAR) · Convolutional neural network (CNN) · ResNet · Saliency map

1 Introduction

Human activity recognition (HAR) has recently been a research hot spot, attracting not only crowds of researchers but also plenty of funds. HAR methodologies have gotten very competent at recognizing human activities directly from raw sensor signals, which has wide applications including home behavior analysis [23], ubiquitous computing [7], health monitoring [18], etc. There are mainly two types of HAR [2]: video-based and sensor-based. We will focus on sensor-based here because of he simplicity of one-dimensional time series data and the privacy concerns it addresses.

There are mainly two types of HAR [3]: video-based and sensor-based. Video-based HAR analyzes videos of humans performing activities in front of a camera. It conforms to our intuition for recognizing human activities but suffers

S. Deng et al. (Eds.): MobiCASE 2021, LNICST 434, pp. 129–143, 2022.
https://doi.org/10.1007/978-3-030-99203-3_9

from many problems, like the complexity to process high-dimensional data, the reliance on environmental illumination, and the need for fixing the position of camera. At the same time, sensor-based HAR can process data from various sensors, such as accelerometer, gyroscope, magnetometer, Bluetooth, and other sensors. Although it requires users to wear some kind of special equipment, the simplicity of 1-dimensional time series data and the privacy concerns it addresses make it popular. There are also works on using mobile phones or smart watches as sensors, which makes it even more convenient. Therefore, in this paper, we will focus on sensor-based HAR.

In general, HAR can be treated as a typical pattern recognition problem where machine learning is very effective [2]. Conventional HAR methods adopt machine learning algorithms such as decision tree, support vector machine (SVM), naive Bayes and hidden Markov models as classifiers. However, a main problem is the heavy reliance on hand-crafted feature extraction which is constrained to the knowledge of the practitioner. Furthermore, the learnt features are always shallow and unable to generalize, which means there is no universal solution to every dataset. Due to these limitations, traditional HAR methods are restricted both in accuracy and generalization ability.

Recent years have witnessed the rise and rapid development of deep learning [13], which have achieved unparalleled performance in computer vision [12], natural language processing [1] and speech processing [9]. As a representation learning method, it can automatically learn deep features which are most useful for classification without any hand-crafted preprocessing. This advancement improves over traditional methods where the hand-crafted features are required. In [24], authors report about the existing works and future directions at the intersection of deep learning and HAR.

One of the most prominent deep learning methods are convolutional neural networks (CNN). The models are attractive in their ability to exploit spatial information inside datasets and were first used to classify images. Extending this ability to HAR, these approaches can exploit the time correlation between adjacent points in one-dimensional (1D) time series data [4,6,17,22,26,27]. However, while CNN structures in the computer vision field keep evolving, from AlexNet [12] to VGGnet [20], Inception [21], ResNet [8], DenseNet [11], MobileNet [10], and other advanced models with better performance and efficiency, the main structure of CNN in HAR field has been stuck in VGG-like model. Therefore, it is necessary to explore the possibility of applying the advancement of structures from visual field to HAR tasks.

Among diverse novel CNN structures, ResNet is one of the most commonly used and is highest performing. It won the first place of ILSVRC-2015 and was awarded the best paper of CVPR in 2016 [8] and was followed by many variants. It is serving as the main structure of many visual tasks like image classification, object recognition, semantic segmentation and so on. Although it is originally designed for images, the core idea - residual learning can be easily transplanted into one-dimensional CNN designing. In this work, we have developed a novel structure based on the same idea of ResNet for HAR. We have

tested it on mainstream dataset OPPORTUNITY [3] and achieved nearly 1% better in accuracy than even the best networks to our knowledge, with over 10 times parameter reduction.

Without architecture, another essential question is that where we should put the sensor to recognize the human activity better. We creatively transplant a method called Saliency Map [19] to directly visualize the importance of every channel to the final result. Saliency Map comes from image classification field and is used to visualize which part inside the image contributes most to the classification result. The same method used here can help us find out the most important sensor without training many times or trying different combinations.

In summary, the contributions of this work are as follows:

- We put forward a novel ResNet-like CNN structure for HAR tasks with improved performance.
- The network we designed has significantly fewer parameters which is capable for more efficient computing.
- We use Saliency Map to directly visualize the importance of every input channel.

The rest of the paper is organized as follows. We discuss related work in Sect. 2. The details of our three contributions are presented in Sect. 3 (ResNet-like structure) and Sect. 4 (Saliency Map usage). The conducted experiments and results are discussed in Sect. 5. Section 6 concludes the paper and lays out potential future work.

2 Related Work

In this section, we discuss related works and contrast them with ours. Section 2.1 discusses previous works which also apply CNNs for HAR and their flaws. Section 2.2 introduces the origination of ResNet and how this residual learning idea influences our work. Section 2.3 introduces the application of Saliency Map in Computer Vision field and how it can be converted into HAR.

2.1 CNNs for HAR

As presented before, CNN has become a wide-spread tool for HAR in recent years. To our knowledge, [27] was the first work using CNN to process time series data. It treated every channel like RGB of an image but did the convolution and pooling separately, which may be considered unreasonable today. This mistake is corrected in [26], which proposed to share weights in 1D multi-channel convolution. Along with this basis, [4] did some experiments to find the optimal kernel size for HAR data. [6] did a comprehensive comparison of deep learning models for HAR, including DNN, CNN, RNN and hybrid models. Some very recent work still follows the pattern, with the changing from common filters to Lego filters [22], improving the computational efficiency for mobile applications. However, all these networks share the same main structure, only varying in number of

layers or kernels. It is necessary to explore novel architectures, like employing skip connections, which is essential in state-of-the-art CNN network designing. It is worth noting that there are some works using CNN+LSTM architectures [15,25], which have higher accuracy than our architecture. However, since CNN part and LSTM part are independent to each other, the simple structure of CNN can be easily replaced with our version and it should achieve better results.

2.2 ResNet

ResNet originated from [8] and have become one of the most wide-spread architectures for deep learning networks. The core idea is residual learning, which uses shortcut connections to ease learning complexity and strengthen gradient flow. However, the original network is for 2D pictures and cannot be directly used in HAR, so we followed the core idea of residual learning and developed our new architecture for 1D time series data. Our new architecture is based on Res Block which is designed according to residual learning concept.

2.3 Saliency Map

Saliency Map [19] comes from Image Classification field and is put forward to answer the question that which part of the picture contributes most to the final classification result. There is a similar question in HAR field that which sensor is most useful or where is the optimal position of sensor to achieve best performance with fewest sensors. Therefore, we developed a 1D version of Saliency Map to visualize the importance of every input channel, which could be used for dimensionality reduction further.

3 ResNet-Like Structure

This section has a detailed description of the basic idea of residual learning and our novel network which employed similar residual learning idea. The concept of residual learning is discussed in Sect. 3.1 and the network architecture in Sect. 3.2.

3.1 Residual Learning

Our architecture shares the same basic idea with ResNet – residual learning. According to the CNN network designing principles, the architecture of a network should be based on stacking of similar blocks, in order to simplify parameter tuning and prevent from overfitting specific dataset and leading to low generalization ability. Therefore, we will explain our network design and theory behind through blocks.

The mathematical model of Residual Blocks can be shown in one line of formula:

$$y_l = F(x_l) + x_l \tag{1}$$

Here y_l denotes the output feature to the l-th residual block and x_l is the input. $F(\cdot)$ denotes residual function and is always stacks of convolutional layers. Instead of directly learning the mapping $H(x)$ from input to output in traditional CNN models, the convolutional part of residual block learns $F(x) = H(x) - x$. It can be seen as a shortcut connection from input to output.

The advantages of introducing shortcut connection can be understood in three perspectives.

First, the complexity of learning is reduced, especially when the features generated by current layer is fine-tuning of last layer's, which happens mostly in deeper networks. Take identity mapping $y = x$ for example. In traditional CNN networks, you must precisely adjust the weights to achieve $H(x) = x$. While in residual blocks, the only thing you need to do is setting all weights to zero. It means $F(x) = 0$ and $H(x) = x$.

Second, it strengthens the gradient flow. One of the most serious difficulty to train very deep CNNs is that gradient may vanish or explode during flowing from deep layers to shallow ones. However, because of the connection between deep layers and shallow ones established by shortcuts, the gradient can be directly transferred without any loss, which avoids gradient vanishing or explosion and accelerates learning process.

Third, it introduces multiscale receptive field. Now the input to every residual block combines features from different layers' output, which extracts features of different complexity from different length of input signals. This kind of multiscale learning corresponds to biological nature of human cognition.

3.2 Network Architecture

Based on the idea of residual learning discussed above, we construct a 1D CNN network shown in Fig. 1a. The whole architecture stacks residual blocks with different number of kernels. The basic structure of residual block is shown in Fig. 1b. It consists of two convolutional layers with batch-norm and relu activation, and adds a shortcut connection from the input to the output of the second layer. We have used the pre-activation version recommended in [14] which means stacking layers in relu-bn-conv style. The only variable of each residual block is the number of kernels.

The whole network starts with a first convolutional layer with 16 kernels. Then there are three stacked residual blocks with 16 kernels and a max pooling layer. Next, two stacked residual blocks with 32 kernels and another max pooling layer followed. After each pooling we have 1×1 convolutional layer to transform channels numbers for the convenience of shortcut connection because it is more difficult to realize if the channel number of input and output differs. Finally, we use global average pooling layer to unify the feature maps among all channels and a softmax layer to get the probability for each class. All the convolutional kernels are 1×3 in order to achieve more complicated model in small receptive field.

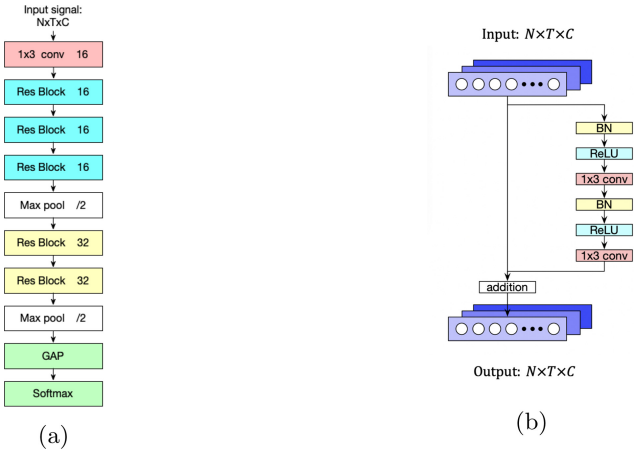

Fig. 1. (a) The structure of the whole network. N, T, C respectively represents number of input samples, time points and channels. (b) Detailed design inside every Res Block. The basic design follows [8]. The BN layer is prepositioned for better performance.

4 Saliency Map

A question well-worth researching in HAR is: where is the optimal position to place sensors? Answering this question could help us in i) abandoning irrelevant channels in order to achieve higher accuracy, ii) reducing the dimensions of the input signals and improve the efficiency of both training and inferring, and iii) reducing the obtrusiveness and hassle of wearing too many sensors which could affect normal daily life.

There have been many works trying to find the answer by wearing sensors on all potential positions, like the head, chest, upper arm, wrist, waist, thigh, leg and ankle [5,16]. Then they trained different networks with all combinations of sensors and compare the performances. The combination of sensors for model with the best performance is considered as the optimal position.

However, this approach has obvious problems. First, it requires an extensive research to train many different networks, which is time-consuming and cumbersome. The number of combinations will also explode when the possible positions increase. Second, the gaps between the best model and others are too small to judge. In the experiment of [5], the thigh was considered to be the best place with 99% accuracy. However, the model trained with sensors on chest and side waist also achieved 98.5% and 98.34% respectively. It is hard to tell if this gap is just because of noise or sensors on thigh is really better than other places.

Therefore, we use a method called *Saliency Map*. It is widely used in the image classification field to visualize which part of input image contributes most to the classification result. The core idea is to compute the gradient of classification unit to input signals, as Eq. 2 shows.

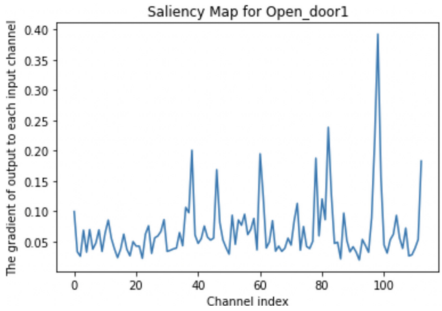

Fig. 2. Saliency Map for class Opendoor1. We computed Saliency Map for every sample in this class and averaged them all in order to eliminate the effect of noise. Each channel is associated with one sensor.

$$\omega = \frac{\partial S_c}{\partial I} \tag{2}$$

where S_c denotes the classification score for class c and I is the input signal. ω has the same dimensionality with input signal. It can be seen as the extent that how the scores will change along with input signal. From another perspective, it can also be considered as weights applied on inputs. If the weight is large, even small changes in input signal will result in big difference so we can say the result is sensitive to these channels and vice versa.

In practice, the dimensionality of the saliency map for a single sample is $T \times C$. T denotes the length of time series and C is number of channels. We add up the matrix along vertical axis because time length can not be treated separately. In the end, we obtain an array whose length is the same as the number of channels. If plotted, we can see peaks and valleys, like the ones in Fig. 2.

It is worth noting that a single sample has randomness, so the Saliency Map generated will be affected by noise. Therefore, a stable Saliency Map should be the average of abundant Saliency Maps generated by samples of the same class.

The values attached to each channel can be seen as the representation of importance. Therefore, we can consider channels with top values as the most important sensors. We have conducted more experiments in Sect. 5.5 to find out how convincing these top channels are and what kind of benefits it could bring.

5 Experiment and Result

This section includes the introduction of the dataset we used and several tests we conducted to prove the practicality of our theory.

5.1 Dataset

The OPPORTUNITY Activity Recognition Dataset is one of the most popular HAR datasets [3]. It originated from OPPORTUNITY challenge in 2010 and is still used as benchmark test of many state-of-the-art structures. Overall, it contains recordings of four subjects in a daily living scenario performing morning activities, with sensor-rich environment. We are focusing on Task B2 which contains 18 classes. During the recordings, each subject performed a session five times with activities of daily living (ADL) and one drill session. During each ADL session, subjects perform the activities without any restriction, by following a loose description of the overall actions to perform (i.e., checking ingredients and utensils in the kitchen, preparing and drinking a coffee, preparing and eating a sandwich, cleaning up). During the drill sessions, subjects performed 20 repetitions of a predefined sorted set of 17 activities. The dataset contains about 6 hours of recordings in total.

We follow the mainstream training and testing setting to make it possible to compare our work with famous and recent ones. It means training our network on all five ADL sessions and drill session for the first subject and on ADL1, ADL2 and drill sessions for Subjects 2 and 3, reporting classification performance on a testing set composed of ADL4 and ADL5 for Subjects 2 and 3. ADL3 sessions for Subjects 2 and 3 are left for validation.

In terms of the sensor setting, we follow the OPPORTUNITY challenge guidelines, taking into account only the on-body sensors. This includes 5 commercial RS485-networked XSense inertial measurement units (IMU) included in a custom-made motion jacket, 2 commercial InertiaCube3 inertial sensors located on each foot and 12 Bluetooth acceleration sensors on the limbs. Each IMU is composed of a 3D accelerometer, a 3D gyroscope and a 3D magnetic sensor, offering multimodal sensor information. Each sensor axis is treated as an individual channel, yielding an input space with a dimension of 113 channels. Since raw data is continuous 1-dimensional time series signal with multiple channels which is unable to be trained on, we have applied the data segmentation method recommended in [3]. The sample rate of these sensors is 30 Hz and recommended time window is 500 ms. It means each segmentation consists of 15 time points. In conclusion, the raw data is turned into training data with $N \times 15 \times 113$ dimensionality.

One main problem of the OPPORTUNITY dataset is its unbalanced distribution between different classes. For example, NULL class dominates the whole dataset with more than 80% but Open Drawer 1 - the minimum class, only occupies 1%. It has some risky disadvantages. The whole model will be very easily overfitting on small classes but is not showing any evidences since NULL class dominates the most. We will try to discuss and ease this unbalanced feature in Sect. 5.4.

5.2 Experiment Parameters

The whole network is trained on single MacBook Pro with 8 Intel Core i9 processors and AMD Radeon Pro 5000M. Our training uses mini-batch gradient

Table 1. Comparison of NF, parameters, and FLOPs of our ResNet-like structure and four baseline networks on OPPORTUNITY dataset. The metrics with a "*" mean they are not reported in the original paper and we do the computation by ourselves.

	NF	Parameters	FLOPs
Yang et al. 2015	85.1	0.912M*	1.85M*
Ordonez and Roggen, 2016	88.3	7.44M	54.58M*
Tang et al. 2020 baseline	86.1	3.2M	41.9M
Tang et al. 2020 simplified	84.5	0.42M	13.78M
Our ResNet-like structure	**89.2**	**0.024M**	**0.25M**

descent with 0.9 momentum and 256 batch size. The learning rate is set as 0.001 initially and divided by 10 when plateau. We have tested many other hyperparameters and ended up choosing this group that worked best. Since the selection of parameters is not the main topic of this work, we don't do more demonstration and discussion.

5.3 Performance

The results of our ResNet-like structure and four baseline networks on OPPORTUNITY dataset are shown in Table 1. We have chosen two most famous work [6,26] and one most recent method [22]. Following [6], normalized F-measure(NF) is used to evaluate the performance.

From the comparison, we can see that our ResNet-like structure performs better than all four baselines with a significant reduction in parameters and FLOPs. Actually, there has already been a trend to explore more efficient networks while keeping the performance stable, reducing parameters by more than ten times. However, there is still a significant gap in computational complexity between our network and very recent work. We attribute this significant reduction to several factors. First, we found that most of the previous works treated the data format as 2-dimensional map and only did convolutions on time axis, leaving channels unchanged. It means keeping 113 channels throughout the whole network, which is obviously a burden. Like in [26], the first layer does 1×5 unpadded convolution to the input signals, changing it from $N \times 113 \times 30$ into $N \times 113 \times 26 \times 50$. However, in our network, we treat channels of sensors and 'channels' of kernels the same, which is very different from the mentioned method. Our first layer transforms input data from $N \times 15 \times 113$ into $N \times 15 \times 16$, which can be seen as the rearrangement of different sensors' signals. The different ways to treat two kinds of channels lead to the significant reduction of parameter numbers. Second, we got rid of the fully connected (FC) layers and use global average pooling [14] instead. FCs is notorious about the explosion of parameters and we find that a network without them for HAR, like in many other fields, can still achieve equivalent performance. Finally, the residual block makes learning easier as explained in Sect. 3.1, so that we can use very few kernels compared with

Table 2. Comparison between the same network trained with different setting of training dataset.

	Part with NULL	Part without NULL	Full with NULL	Full without NULL
Yang et al. 2015	89.2	77.3	93.7	90.1

previous works while achieving same performance. We use 16 and 32 kernels here which is significantly reduced from 128 [6], 60 [15] and 64 [22].

5.4 Discussion of Unbalance

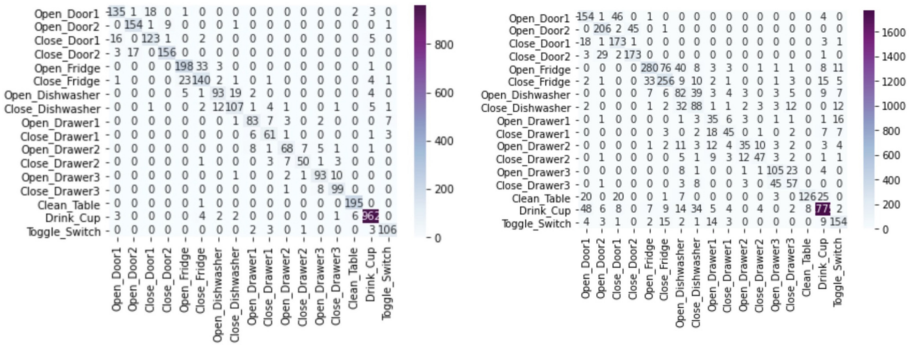

Fig. 3. Left: Confusion matrix of the network trained with full data. Right: Confusion matrix of the network trained with partial data

Besides comparing our network with state-of-the-art works in a mainstream training setting for OPPORTUNITY, we have conducted several more experiments with different setting. We remove NULL class to see more clearly what happens inside small classes. We also have tried training on the whole dataset with and without NULL class, applying train-val-test split by 8-1-1, to see if the increasing data will affect the performance of same model. The results are shown in Table 2. The confusion matrix for part without NULL and full without NULL is shown in Fig. 3 for more detailed discussion.

From the results, we can see that both in part dataset and full dataset, if NULL class is discarded, the performance drops at different level. This provides efficient evidence for the problem mentioned before: the unbalanced distribution of different classes in OPPORTUNITY. After discarding the dominant NULL class, the problem of lacking data for some small classes emerges. That is why the performance drops. We can also see that the network trained with full dataset has a relative small margin. That is because full dataset has more data which eases the overfitting problem, especially for some small classes.

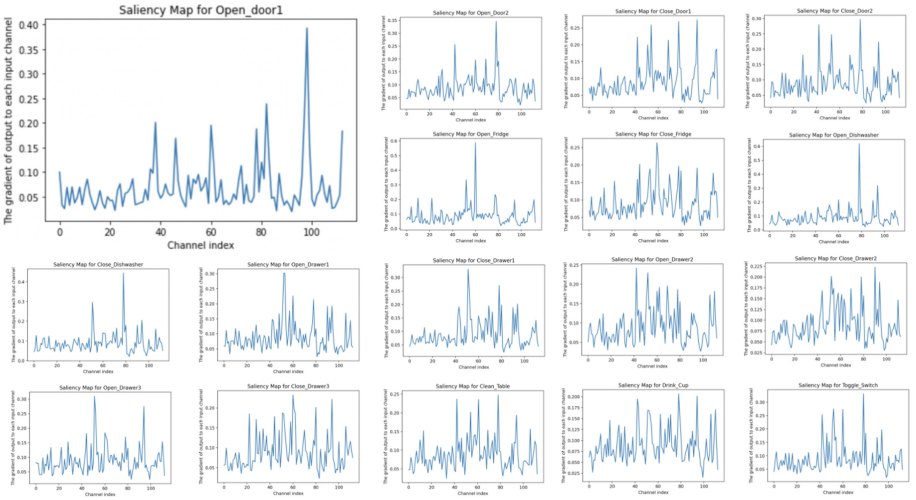

Fig. 4. Saliency Maps for all 17 activities in OPPORTUNITY, Open door1's is zoomed in for detailed observation and explanation.

When we conducted a parallel comparison between full and part dataset, the performance is improved by 4.5% when with NULL and 13% when without. It means obviously that providing more data can greatly enhance the network, especially when there is strong overfitting in original network. It also provides the evidence that current network structure is complex enough and what we need is to collect more data for improvement.

When it comes to the confusion matrix, we found that network trained with part dataset does not perform very well. Some samples will be misclassified as any class which means the network has not learnt an appropriate feature representation for the correct class. At the same time, the network trained with full dataset performs very well. The misclassification mainly happens in block diagonal matrix, which means the correct class and output class has just nuance. It means the network has learnt deep features but these features are not strong enough to classify between these very close classes. The possible improvement method should be more complicated network structure or new technical idea.

5.5 Saliency Map Usage

As discussed above, the Saliency Map for each activity class in OPPORTUNITY appears to be a curve with 113 points after processing. Each point is corresponding to one channel of a sensor and the value is considered to be the importance of this channel to the final result.

During experiments, we computed Saliency Map for every activity class and showed it in Fig. 4. First of all, let's take a look from big picture. Each map has clear peaks and valleys, which means we can easily extract channels with

Table 3. The top 10 and bottom 10 indexes from the saliency map for class 1 - open door1 and their corresponding sensors. The correspondence and positions of the sensors or other information can be found in [3].

Top 10 maximum		Bottom 10 minimum	
Index	Corresponding sensor	Index	Corresponding sensor
62	RLA mag X	15	LH acc Y
40	BACK acc Z	112	R-Shoe AngVelNavFrame Y
56	RLA acc X	97	L-Shoe AngVelNavFrame Y
55	RUA mag Z	108	R-Shoe AngVelNavFrame X
54	RUA mag Y	113	R-Shoe AngVelNavFrame Z
53	RUA mag X	75	LLA acc Y
44	BACK mag X	94	L-Shoe AngVelNavFrame X
84	L-Shoe Eu Y	96	L-Shoe AngVelNavFrame Y
99	R-Shoe Eu X	77	LLA gyro X
100	R-Shoe Eu Y	92	R-Shoe AngVelNavFrame X

Table 4. The same network structure with same learning schedule trained with different setting of input channels

How many channels used	Performance (NF)	Parameters in total
All 113	91.6	24594
Top 20	88.1	19650
Bottom 20	57.6	19650

high or low importance. Another interesting fact is that similar activities have similar patterns, particularly obvious are the map for Open Dishwasher and Close Dishwasher. It makes sense because in similar activities, such as opening or closing something, high importance should be given to same sensors, which is exactly what we have observed.

Then we take a step closer to analyze the information given by every single map. Because of the limited page length, we can not give each map a detailed explanation. Therefore, we mainly made detailed explanation for Activity 1: Open Door1, and other maps can be further explored in the same way.

To check if the values of each point has any realistic meaning, we cut out the top 10 channels with most importance and bottom 10 with least. The index numbers and corresponding sensors are shown in Table 3.

From Table 3, we can see that top 10 sensors mainly consist of sensors on right hand (RUA RLA) and on back (BACK), especially magnetic. It is reasonable since the key to opening the door is the movement of the right hand and the rotation of the waist. It doesn't matter what the left hand is doing or the state of the lower body. At the same time, there are some sensors about left

hand in bottom 10 sensors which means signals related to left hands have little effect on classification, which is the same result as above. Besides, nearly all AngVelNavFrame sensors are on bottom 10 which means this kind of sensor is all useless for classification. AngVelNavFrame, according to its description in [3], represents orientation of the sensor with respect to a world coordinate system in quaternions. It is a useless feature beyond any doubts, because nobody cares which direction I'm heading when I open a door.

In order to further verify our theory, we average maps for every activity to generate the final Saliency Map for this whole dataset. Then we extract top 20 channels with highest values and bottom 20 channels as well. We have trained three different networks separately with all 113 channels, top 20 channels and bottom channels.

The results are shown in Table 4.

According to results, the performance of the whole model just drops by 3% when we choose top 20 channels as inputs instead of whole 113, while the parameters are reduced by 20%. It means that we must have discarded most useless channels and only the channels with strong ability to classify between different classes remain, significantly improving the efficiency of learning useful features.

At the same time, we observe that network trained with 20 top and 20 bottom have same network structure and same number of parameters but end up with totally different performance (88.1% and 57.6%). It provides the evidence of the assumption that channels with larger values on Saliency Map have stronger classification ability again.

It is worth noting that even the network trained with 20 bottom channels has stronger classification ability compared to empty model, which means there are still some information inside the channels which are discarded.

In conclusion, the values of Saliency Map can indeed reflect the importance of the corresponding sensor. Depending on the results of Saliency Map towards whole dataset, we can get direct feedback about which channels are most important. Therefore, dimensional reduction can be done by reserving channels with highest values and dropping the others. The optimal positions or sensors could also be found by directly comparing the values inside the Saliency Map.

6 Conclusion and Future Work

In this paper, we proposed a new ResNet-like CNN structure for HAR tasks. This novel network takes advantage of residual learning and achieves state-of-the-art performance with significant parameters and computational complexity reduction. The improvement can be attributed to three factors: i) a different processing method towards sensor channel and kernel channel, ii) removing fc layers, and iii) making kernel numbers smaller using residual learning. Another contribution of our work is the application of Saliency Map in HAR model. Using this method, we can visualize the importance of every input channel, which is corresponding to actual sensors wore on different positions. Based on the Saliency Map, we can visualize the importance of each sensor and conduct further work

like dimension reduction to improve computational efficiency or find the optimal position with the largest value.

In our experiments, we demonstrated the performance of the proposed ResNet-like structure and compared it to other state-of-the-art works to prove its advantages. We also conducted a contrast test to show that the usage of Saliency Map can really benefit the network. Therefore, we believe that a ResNet-like structure can serve as a competitive structure of feature learning and classification for HAR problems and Saliency Map will serve as another useful tool.

For future work, there are two main ways to proceed. First, we will keep exploring the application of more advanced CNN network architecture on HAR tasks, like DenseNet, MobileNet, and others. Second, since the usefulness of Saliency Map is verified, a more complicated system should be designed to automatically realize the dimensional reduction work, like reserving the top 20 channels. We would also like to develop our novel network to specific applications of Physical Therapy to test the robustness on actual application scenarios.

References

1. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: Artificial Intelligence and Statistics, pp. 127–135. PMLR (2012)
2. Bulling, A., Blanke, U., Schiele, B.: A tutorial on human activity recognition using body-worn inertial sensors. ACM Comput. Surv. (CSUR) **46**(3), 1–33 (2014)
3. Chavarriaga, R., et al.: The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recogn. Lett. **34**(15), 2033–2042 (2013)
4. Chen, Y., Xue, Y.: A deep learning approach to human activity recognition based on single accelerometer. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1488–1492. IEEE (2015)
5. Cleland, I., et al.: Optimal placement of accelerometers for the detection of everyday activities. Sensors (Basel) **13**(7), 9183–200 (2013). https://doi.org/10.3390/s130709183, https://www.ncbi.nlm.nih.gov/pubmed/23867744
6. Hammerla, N.Y., Halloran, S., Plötz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880 (2016)
7. Harrison, C., Tan, D., Morris, D.: Skinput: appropriating the body as an input surface. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 453–462 (2010)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
10. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
14. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
15. Ordóñez, F.J., Roggen, D.: Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
16. Pannurat, N., Thiemjarus, S., Nantajeewarawat, E., Anantavrasilp, I.: Analysis of optimal sensor positions for activity classification and application on a different data collection scenario. Sensors (Basel) **17**(4) (2017). https://doi.org/10.3390/s17040774, https://www.ncbi.nlm.nih.gov/pubmed/28379208
17. Pourbabaee, B., Roshtkhari, M.J., Khorasani, K.: Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. IEEE Trans. Syst. Man Cybern. Syst. **48**(12), 2095–2104 (2018)
18. Qin, J., Liu, L., Zhang, Z., Wang, Y., Shao, L.: Compressive sequential learning for action similarity labeling. IEEE Trans. Image Process. **25**(2), 756–769 (2015)
19. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2014)
21. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
22. Tang, Y., Teng, Q., Zhang, L., Min, F., He, J.: Efficient convolutional neural networks with smaller filters for human activity recognition using wearable sensors. arXiv preprint arXiv:2005.03948 (2020)
23. Vepakomma, P., De, D., Das, S.K., Bhansali, S.: A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 1–6. IEEE (2015)
24. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep learning for sensor-based activity recognition: a survey. Pattern Recogn. Lett. **119**, 3–11 (2019)
25. Xia, K., Huang, J., Wang, H.: LSTM-CNN architecture for human activity recognition. IEEE Access **8**, 56855–56866 (2020)
26. Yang, J., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
27. Zeng, M., et al.: Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International Conference on Mobile Computing, Applications and Services, pp. 197–205. IEEE (2014)