Image-Mediated Data Augmentation for Low-Resource Human Activity Recognition

Zihao Wang
Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing
16281303@bjtu.edu.cn

Youli Qu
Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing
ylqu@bjtu.edu.cn

Junru Tao
Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing
16281300@bjtu.edu.cn

Yudan Song
Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing
16281298@bjtu.edu.cn

ABSTRACT
Data augmentation is a kind of technique that widely used to solve many machine learning tasks, such as image classification, it enlarge the training dataset size virtually and avoid overfitting. Some former work has demonstrated the effectiveness of data augmentation while using simple techniques, such as cropping, rotating, and flipping the input images. However, there are few augmentation techniques for time series. In this paper, we propose a novel data augmentation technique for human activity recognition (HAR) task, whose data set is consist of time series. Our technique is named image-mediated data augmentation (IMDA), as the name suggests, we create new samples by series-to-image-to-series conversion. When the series data is converted into an image, the augmentation can be easily applied with traditional techniques for images. Then we convert this newly generated images back to the series data. So that we finished the data augmentation towards time series. Experimental result shows that IMDA improves classification accuracy from 90.50% to 91.55% on the HAR dataset. It also show that IMDA largely improved classification accuracy when the quantity of samples in the training set was quite small. The experimental result which basic on simulated low-resource settings shows that IMDA improves the classification accuracy by up to 11.67%. Therefore, our technique is more valuable for low-resource tasks.

CCS Concepts
• **Information systems → Information systems applications → Data mining → Clustering**

Keywords
Data Augmentation; Human Activity Recognition; Time Series

1. INTRODUCTION
During the image processing, the training data is augmented by such as horizontally flipping, random cropping, tilting, and altering the RGB channels of the original images [1, 2]. Since the content of the new image doesn't change, the label of the original image should be preserved. While data augmentation has become a generic technique in training deep neural networks for image processing [3, 4], it is not a common practice in training neural networks for time series classification tasks such as HAR.

In our real life, many problems can be defined as time series problems. In reality, Human Activity Recognition (HAR) is valuable in both theoretical research and actual practice. It can be used widely in health monitoring [8, 9], smart homes [10, 11], human-computer interactions [12, 13] and so on. More specifically, HAR is a process that obtain action data with sensors. It abstracts the action information and then understands and extracts the motion's characteristics, which is what activity recognition refers to.

In this paper, we propose a simple but effective approach-- image-mediated data augmentation (IMDA), which augments the training data by series-to-image-to-series conversion. When the data is converted into an image, the augmentation can be easily applied with traditional ways. With converting the newly generated images back to the series, we can get some new samples. This kind of simple data augmentation technique improves classification accuracy from 90.50% to 91.55% on the HAR dataset with a simple 4-layer Conv-LSTM network [14, 15].

To prove that our IMDA technique brings more advantages when the quantity of samples in the training data is less, we conducted the evaluations by reducing the number of samples used for training with the HAR. The results showed that IMDA makes greater improvements in accuracy when the quantity of samples are less than the full HAR dataset. When we used only 100 samples per label, thus, 700 samples in total, the classification accuracy was improved from 76.11% to 87.78% with IMDA. Based on these results, we deem that our technique is more suitable for tasks who have a limited quantity of available training datasets, that is, the low-resource case.

ICCDA 2019, March 14–17, 2019, Kahului, HI, USA

ACM ISBN 978-1-4503-6634-2/19/03…$15.00
https://doi.org/10.1145/3314545.3314565

2. IMAGE-MEDIATED DATA AUGMENTATION

2.1 Implementation
This section expounds our imaged-mediated data augmentation technique. The basic idea of our technique is simple: Since there are many methods of image augmentation, in order to achieve the augmentation towards the time series, we can convert the time series into image first, then use the existing image augmentation methods to product new samples, and then re-convert them to the time series. For time series classification problem, generally

speaking, each time series of a specific length corresponds to a label. We aim to convert each time series into an image, and then each generated image will correspond to a label.

The question is how to convert a time series into an image. The most straightforward way is to generate a curve of parameter values varying with index in rectangular coordinates. But this approach is not suitable for data augmentation, because there is only one direction represents data, while the other direction have nothing to do with data, so changes can only be applied in one direction, which is quite limited. To solve this problem, we propose to use polar coordinates.

Firstly, for a time series of length L, the data is expected to be evenly distributed in the range of $[0,2\pi]$, in other words, the data whose index is i will be drawn at the angle of $i*2\pi/L$, and the polar radius of each point is the parameter value corresponding to its index. Figure 1 illustrates drawing a sinusoidal curve in polar coordinates. The black points in rectangular coordinates correspond to those in polar coordinates. Similarly, we first normalize the index of the sequence with the range of $[0,2\pi]$. Then, taking the index as the angle and the parameter value as the polar radius, the time series will be plotted in polar coordinates in the form of scatter points. In this way, every direction of the image represents a data, so we can introduce changes in any direction, thus to improve the flexibility of conversion greatly.

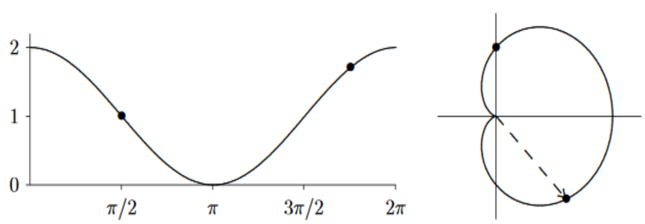

Figure 1. Sinusoidal curve in polar coordinates

On the other hand, the length of a time series is determined by both the time and the frequency of sampling. When the time is fixed, the density of the scatter points only depends on the frequency of sampling. The most extreme case is the lossless analog-to-digital conversion when the sampling frequency is high enough to be almost continuous. In this case, the image we drew is a smooth curve. Therefore, we can use existing scatter points fitting a smooth curve to simulate the real situation. Through this fitting, we can increase the finite scatter points to countless points. This smooth curve will be our data pool, which enables us to take points from it on demand. It is noteworthy that since the index is mapped to the angle in polar coordinates and the index has nothing to do with the data, we should pay attention to the points with the angle of $i*2\pi/L$,(i=0,2,3 ...L-1) on the smooth curve after augmentation.

Figure 2 and Figure 3 show the image which is converted from the data corresponding to one of the tags in the HAR dataset. HAR data set describes an action with a time series of 128 steps in nine channels. First, we normalized all the data based on Min-Max normalization, then normalized the index of the time series to be within the range of $[0,2\pi]$. At last, make the scatter points of each channel fit with a smooth curve, which different channels are drawn with different colors. Figure 2 and Figure 3 respectively shows the result in rectangular coordinates and polar coordinates. It is a process that time shifting from the last element of the time series to the first, which couldn't happen in reality and has no physical significance, as Figure 3 shows, the curve is not closed. Therefore, we should always pay attention to this point and be careful with the boundary treatment.

Since we have generated the image, we can use traditional image augmentation technology to augment the image, and then operate on the image further more to restore it to the time series, and that's how we achieve the augmentation of time series.

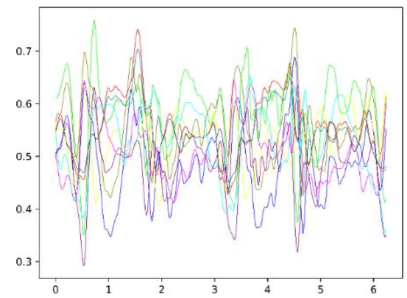

Figure 2. Conversion result in rectangular coordinates

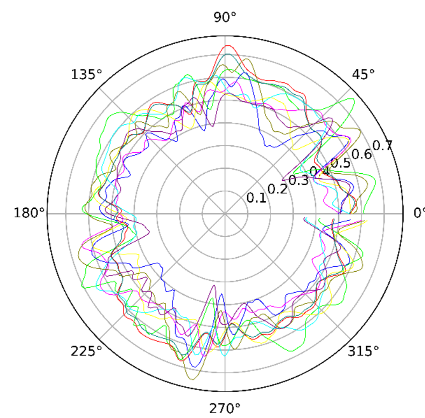

Figure 3. Conversion result in polar coordinates

But the process of generating images and processing them pays too much, it consumes a lot of computing resources. In order to reduce the computational cost, we are thinking of skipping the process of image generation to simplify this conversion process. Without generating images, we can still achieve the same effect by merely processing data.

We still regard a time series as a series of scatter points, whose index is distributed in the range of $[0,2\pi]$, and deem the parameter values as the polar radius. Then execute the following steps:

First, normalize the time series of length L based on Min-Max normalization, we can get a series:

$$\rho: \qquad \rho_1, \rho_2, \rho_3, \dots, \rho_k, \dots, \rho_L$$

with angles:

$$\theta^0: \qquad 0, \frac{2\pi}{L}, \frac{2*2\pi}{L}, \frac{3*2\pi}{L}, \dots, \frac{(k-1)*2\pi}{L}, \dots, \frac{(L-1)*2\pi}{L}$$

The series above can be represented in rectangular coordinates by the following two sequences:

$$\begin{aligned} \text{X:} \quad & \rho_1 * \cos(0), \rho_2 * \cos\left(\frac{2\pi}{L}\right), \rho_3 \\ & * \cos\left(\frac{2*2\pi}{L}\right), \dots, \rho_k \\ & * \cos\left(\frac{(k-1)2\pi}{L}\right), \dots, \rho_L * \cos\left(\frac{(L-1)2\pi}{L}\right) \\ \text{Y:} \quad & \rho_1 * \sin(0), \rho_2 * \sin\left(\frac{2\pi}{L}\right), \rho_3 \\ & * \sin\left(\frac{2*2\pi}{L}\right), \dots, \rho_k \\ & * \sin\left(\frac{(k-1)2\pi}{L}\right), \dots, \rho_L * \sin\left(\frac{(L-1)2\pi}{L}\right) \end{aligned}$$

So far, a series with length L can be uniquely represented by two other series with length L. Then we can apply change f to the transformed series. After adding changes, we will have two new transformed series X', Y':

$$(X', Y') = f((X, Y))$$

The change f can be distorting, rotating, noise and so on. Then restore the new X', Y' with polar coordinates:

$$\rho' = \sqrt{X'^2 + Y'^2} \quad \theta' = \arcsin\left(\frac{Y'}{\rho'}\right)$$

It is notable that the series θ' is not the final result. Since after the change f, the angle of each scatter point has also changed. $\theta' \neq \theta^0$. But the angle corresponds to the index so that the angle should be fixed. To simplify the processing, we simply link scatter plots in order and then we can get a line chart. Afterwards we obtain a series of points whose angles are θ^0 in the line chart. With obtaining the polar radius of these points, we can get the correct result of the new series.

It's good to know that θ' is still an incremental series, so for every θ_i^0, we only need to find two adjacent points satisfying the condition:

$$\theta'_k < \theta_i^0 < \theta'_{k+1}$$

Then the point with the angle of θ_i^0 in the line chart must be on the straight line between the point K and the point $\mathrm{K} + 1$.

The linear equation determined by these two points is

$$\begin{aligned} (\rho_{k+1} \cos\theta'_{k+1} - \rho_k \cos\theta'_k) * (\rho\sin\theta - \rho_k sin\theta'_k) \\ = (\rho_{k+1} \sin\theta'_{k+1} \\ - \rho_k \sin\theta'_k) * (\rho\cos\theta - \rho_k cos\theta'_k) \end{aligned}$$

Let $\theta = \theta_i^0$, the polar radius ρ''_i corresponding to θ_i^0 on the line chart can be solved.

If there is no K in θ' so that $\theta'_k < \theta_i^0 < \theta'_{k+1}$, then θ_i^0 falls into the angle range of no physical significance, which is caused by the non-closure of the line chart. And that is inevitable. Therefore, we can only try some approximations: Let $\rho''_i = \rho'_i$.

Following the rules above, processing all the points in the series in loop, we can finally obtain a new series ρ''. Adjust the parameter values in change f randomly, we will obtain a great deal of new series ρ''. This method skips the process generating and processing images, and carry out the augmentation by merely numerical calculation, which significantly reduces the calculation cost of the augmentation process.

2.2 Rationality Analysis

Since not all the images augmentation methods are suitable for the time series augmentation. It's necessary to analyze which kinds of changes f are reasonable for the time series classification problem, and which kinds are unreasonable. Because the change f only changes the each point's position in the image, in other words, it only changes the shape of the image, and has nothing to do with its color. Therefore, the processing of color in the image augmentation technology is not in our consideration, we only concern about the change of the point's location.

2.2.1 Rotating and Flipping

Flipping is unreasonable for the time series classification problem, because flipping will disturb the order of the series points in the polar coordinates, that is, it will disrupt the series time order, which would destroy the physical significance of the series. Therefore, flipping is not suitable for the augmentation towards time series.

In contrast, rotating can be regard as the horizontal shift of the time axis. To a large extent, the time order is maintained. So, rotating is suitable for the series augmentation.

For the image classification problem, the core features of an image are often concentrated in one or a few small areas, such as the nose of cats, the wings of butterflies. But an image will also contain a lot of unserviceable information, such as the sky and the clouds and so on. With the Attention Model, we can accurately capture which parts of the image contribute the most to the classification. The time series classification problem has no exception, the core features of a sequence are often concentrated in a subsequence, not scattered throughout the full sequence.

For example, for a sequence (1, 2, 3, 4, 5, 6, 7, 8), the new sequence after rotating will be (5, 6, 7, 8, 1, 2, 3, 4). Because it's no physical meaning to make a sequence end to end, we can regard the new sequence as two subsequences (5, 6, 7, 8) and (1, 2, 3, 4). Each of them carries a part of the information of the original sequence. As mentioned above, the core feature of a sequence is carried on the sub-sequence. Now we assume that (2, 3, 4) is the core sub-sequence in this sequence. After the rotation, we can see that (2, 3, 4) still exists in the two sub-sequences above, which is a part of (1, 2, 3, 4). Even if after rotation the sequence becomes (3, 4, 5, 6, 7, 8, 1, 2), which makes the sub-sequence (2, 3, 4) no longer exist, the sub-sequence (3, 4) can still contribute to the classification. And even in the worst case, the core sub-sequence is cut from the middle, it can retain at least half of the contribution. Not to mention in most cases, only a small portion of the core sub-sequence would be cut, and most of them are even not cutting. What's more, rotating randomly can change the position of the core features in the sequence, which could effectively avoid overfitting[5] and makes it more accurately to find and locate features in time series classification problems.

2.2.2 Rescaling and Tilting

Rescaling and tilting have changed the physical meaning of the time series in a sense. For example, when measuring and sampling the speed of an object in a direction. If the object is moving in a uniform linear motion, the samples will be a constant sequence, and the corresponding image will be a circle. But if we process this sequence with rescaling and tilting, the perfect circle will become to an ellipse, which means that the movement on this direction is no longer uniform, but accelerates. We assume that the motion corresponding to the new sequence has a maximum acceleration of a, and if we want to classify the "original uniform linear motion" and "uniform acceleration linear motion with acceleration of $a/10$", such kind of changes is not advisable. But if the object of classification is a uniform acceleration linear motion with an acceleration of $10 * a$, we believe that after adding such kind of changes, they can still be effectively classified.

Therefore, although rescaling and tilting will change the physical meaning of the sequence to some extent, it is still a reasonable change for the augmentation when the magnitude of rescaling and tilting is slight. It is because that when the magnitude of the rescaling and tilting is sufficiently small, the deformation of them is small relative to the variation magnitude of the feature sequence, thus does not affect the features contained in the time series, that is, the features of the time series can be carried in this deformation. By randomly rescaling and tilting during the training, the overfitting can be avoided. Therefore, rescaling and tilting are effective approaches to the augmentation, while the magnitude of the rescaling and tilting should be reasonably determined according to the data.

2.2.3 Noise

For the image classification problem, noise can be added at any random position in the image, but since we are drawing the image ourselves with our determined rule, it makes no sense to add noise at random locations. Different from the image classification problem, for the time series classification problem, the noise should be added to the x, y coordinates of some scatter points directly. To make parts of points be shifted randomly within a small range. But this kind of operation takes great risks, because it may change the chronological order of some points, which is not allowed. Therefore, it is more prudent to make some scatters be shifted randomly in the direction of the polar diameter. The magnitude of the added noise, which is the range of random displacements, is more limited than rescaling and tilting. As the noise is added only to one or a few points, a small magnitude of displacement can also cause large fluctuations, which may be detrimental to the classification. While noise with small magnitude can enhance the tolerance of the classification model to the noise, which may be conductive to the classification.

3. Evaluation

In this section, we investigate the effects of imaged-mediated data augmentation using the HAR dataset. As well as the full dataset with 7,352 samples (HAR-7352), we tested a shrunk dataset with only 700 samples (HAR-700).

Instead of generating images, we directly use numerical calculation to augment the time series by using the method mentioned above. Then compare the classification results before and after the augmentation. We used a network has three convolutional layers and a bidirectional LSTM layer followed by two fully connected layers with dropout [6, 7, 16] as shown in Figure 7. We still trained the network using Adam [17] as the optimizer with the batch size of 20. This time, we used data augmentation by rotating, rescaling, tilting and adding noise.

We first tested on the shrunk dataset (HAR-700). Figure 8 and Figure 9 illustrate HAR-700's validation accuracy and training accuracy. We tested every transform separately except Rescaling and Tilting. We merged Rescaling and Tilting into Distorting because they are almost the same thing. What's more, rescaling is allowed in the range of [0.9, 1.1]; the tilt angle of tilting is allowed in the range of [-5, 5]; the angle of rotating is allowed in the range of [0, 180]; Noise is allowed to be added to one-eighth of the elements in the series, and the range of random displacement (noise) is allowed in the range of [0.95, 1.05].From Figure 8, we can see that all the distorting, rotating and adding noise, did improve the validation accuracy a lot base on the baseline. And it performs best when we apply distorting, rotating and adding noise at the same time. The classification accuracy was improved from 76.11% to 87.78% after applying all these three above at the same time. From Figure 9, we can see that for most of the first 50 epochs of a training, the training accuracy with augmentation is slightly lower than the training accuracy without augmentation. This implies that learning augmentation helps with generalizing the classifier and avoiding overfitting.

(input tensor)	128×9
Convolution 3 &ReLU	128×16
Max Pooling 2	64×16
Convolution 3 &ReLU	64×32
Max Pooling 2	32×32
Convolution 3 &ReLU	32×64
Max Pooling 2	16×64
Bidirectional LSTM	32
Full Connect & ReLU	64
DropOut w/ 20% drop rate	
Full Connect	7
Sigmoid	

Figure 4. Design of classification network used for HAR-700 and HAR-7352

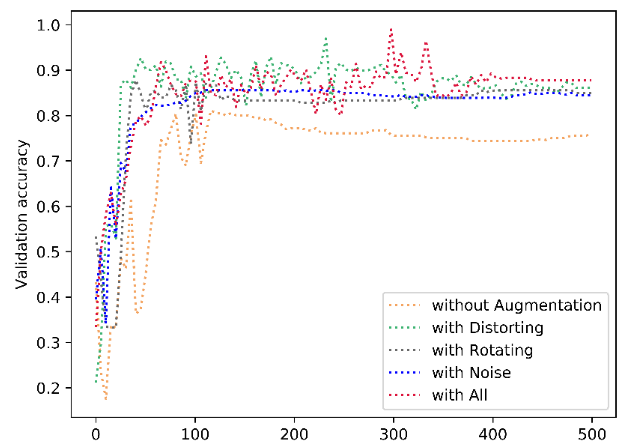

Figure 5. Changes in validation accuracy for HAR-700 dataset with and without augmentation

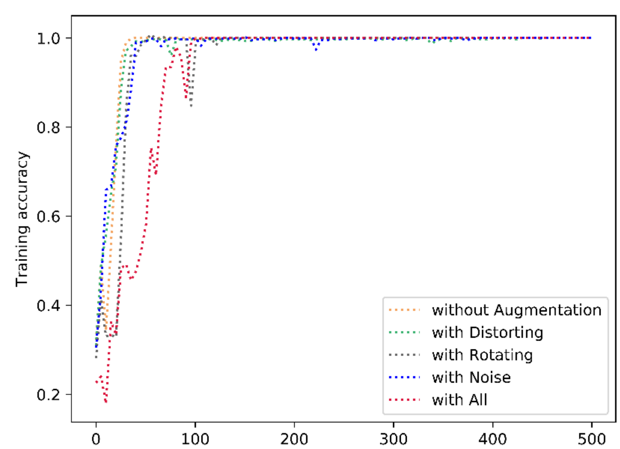

Figure 6. Changes in training accuracy for HAR-700 dataset with and without augmentation

Finally, we tested on the full dataset (HAR-7352). Figure 10 and Figure 11 illustrate the HAR-7352's validation accuracy and the training accuracy. From Figure 11 we can see that the training accuracy with IMDA is always lower than the training accuracy without IMDA. What's more, after applying IMDA, when training accuracy became high enough, it would fluctuate significantly. From Figure 10 we can see that during the whole 1000 epochs, validation accuracy that with IMDA is always higher than the validation accuracy without IMDA. And in the end, based on the whole dataset, the augmentation improves the classification accuracy from 90.50% to 91.55%.

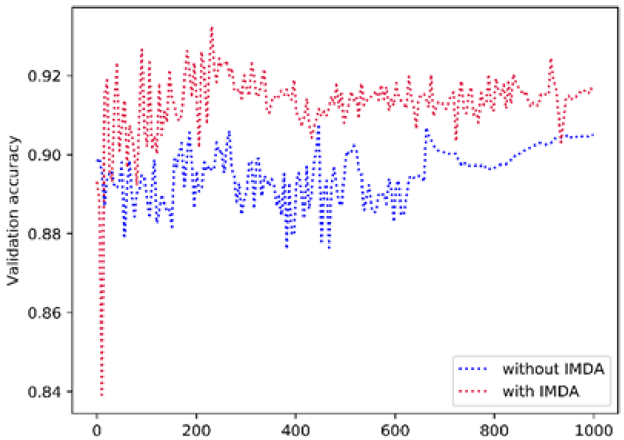

Figure 7. Changes in validation accuracy for HAR-7352 dataset with and without IMDA

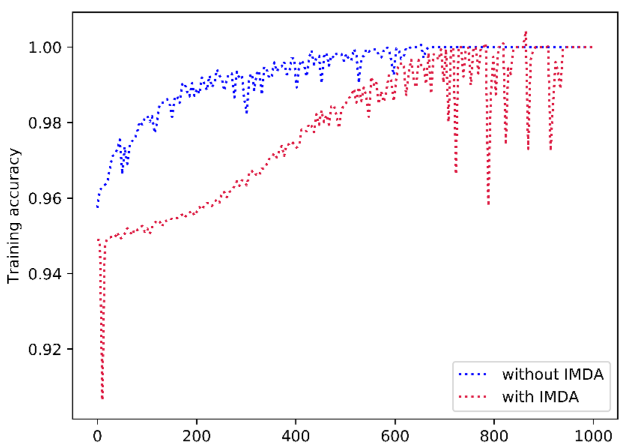

Figure 8. Changes in training accuracy for HAR-7352 dataset with and without IMDA

Based on these results, we can see that IMDA does a better job while dealing with HAR-700, which is regard as a kind of low-resource case. When the data set is large enough, data augmentation will not make a significant difference, but it can still introduce positive effect.

4. CONCLUSION

This paper presents IMDA--our new technique about data augmentation towards time series. IMDA is easy to implement and gives significant improvements in classification accuracy on HAR dataset by avoiding overfitting, especially when the quantity of available samples is limited, that is, the low-resource case. Therefore, IMDA is valuable for tasks with a limited quantity of samples.

IMDA also has its limitations in practice. It cannot solve all the time series classification problems, such as the binary time series classification. In further work, we will try to find a method that would be more general for time series augmentation.

5. ACKNOWLEDGMENTS

The Corresponding Author is Youli Qu. E-mail: ylqu@bjtu.edu.cn

6. REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Annual Conference on Neural Information Processing Systems (NIPS), pp. 1106-1114, 2012.

[2] C. Ken, S. Karen, V. Andrea, and Z. Andrew. 2014. Return of the devil in the details: Delving deep into convolutional nets. In Proceedings of the British Machine Vision Conference. BMVA Press. https://arxiv.org/pdf/1405.3531

[3] Y. Xu, R. Jia, L. Mou, G. Li, Y. Chen, Y. Lu, and Z. Jin. Improved relation classification by deep recurrent neural networks with data augmentation. CoRR, abs/1601.03651, 2016.

[4] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: when to warp? CoRR, abs/1609.08764, 2016.

[5] B. Wang and D. Klabjan. Regularization for unsupervised deep neural nets. CoRR, abs/1608.04426, 2016.

[6] Y. Kubo, G. Tucker, and S. Wiesler. Compacting Neural Network Classifiers via Dropout Training. ArXiv e-prints, Nov. 2016.

[7] Y. Gal and Z. Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. ArXiv e-prints, Dec. 2015.

[8] H. Kalantarian, C. Sideris, Mortazavi B, et al. Dynamic computation offloading for low-power wearable health monitoring systems, IEEE Transactions on Biomedical Engineering. 64(3) (2017) 621–628.

[9] Pantelopoulos A, Bourbakis N G. A survey on wearable sensor-based systems for health monitoring and prognosis, IEEE Transactions on Systems Man & Cybernetics Part C Applications & Reviews. 40(1) (2010) 1–12.

[10] Ahmed S H, Kim D. Named data networking-based smart home, ICT Express. 2(3) (2016) 130–134.

[11] Kumar, Shiu. Ubiquitous smart home system using android application, arXiv preprint, arXiv: 1402.2114, 2014.

[12] Rautaray S, Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey, Artificial Intelligence Review. 43(1) (2015) 1–54.

[13] Yeo H S, Lee B G, Lim H. Hand tracking and gesture recognition system for human–computer interaction using low-cost hardware, Multimedia Tools and Applications. 74(8) (2015) 2687–2715.

[14] Yang J B, Nguyen M N, San P P, et al. Deep convolutional neural networks on multichannel time series for human activity recognition, in: International Conference on Artificial Intelligence. AAAI Press, 2015.

[15] Ordóñez F J, Roggen D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition, Sensors. 16(1) (2016) 115–140.

[16] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.

[17] Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.