Octave Mix: Data Augmentation Using Frequency Decomposition for Activity Recognition

TATSUHITO HASEGAWA (Member, IEEE)
Graduate School of Engineering, University of Fukui, Fukui 910-8507, Japan
e-mail: t-hase@u-fukui.ac.jp

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant-in-Aid for Early-Career Scientists under Grant 19K20420.

ABSTRACT

Research in human activity recognition (HAR) requires a huge amount of data, but it is not easy to collect such measured sensor data. Besides, there is no much application of data augmentation (DA) in HAR. This study proposes Octave Mix as a novel synthetic-style DA method for sensor-based HAR. The proposed method uses frequency decomposition to intersect low- and high-frequency waveforms. In addition, we propose a DA ensemble method and a training algorithm to ensure robustness to the original sensor data while applicable to various feature representations. To evaluate the Octave Mix method's effectiveness, we conduct experiments using four different benchmark datasets of sensor-based HAR and achieve high estimation accuracy in our results. Furthermore, we demonstrate that ensembling two DA strategies: Octave Mix with rotation and mixup with rotation, achieves higher accuracy.

INDEX TERMS

Data augmentation, human activity recognition, ensemble deep learning.

I. INTRODUCTION

The task of recognizing human activities by measuring human movement using sensors is called human activity recognition (HAR). Recognizing people's activities and providing services based on those activities requires a large amount of activity data, which is not always available.

The HAR is often implemented using machine learning [1], while the HAR based on deep learning approach is being actively studied in recent years [2]. Unlike the traditional machine learning models, deep learning generic models require a huge amount of training data to avoid overfitting. Data augmentation (DA) is generally used in image recognition to expand the data [3]. Also, in context-awareness using sensors, it is not easy to expand the labeled data. For example, Inoue et al. [4] proposed a sensor-based nursing activity recognition method. In their study, 22 nurses wearing sensors perform their nursing tasks, and other nurses as observers record their activities manually, to collect a training dataset. A total of 41 different activity class labels are annotated to the sensor data and measured over two weeks. Therefore, it takes a lot of time and effort to collect HAR training data, and effective DA methods are desirable.

In this paper, we propose Octave Mix (Fig. 1), as a novel DA method, for the HAR using sensor data. Also, we propose an ensemble method and a training algorithm by combining the existing DA methods to improve the estimation accuracy. Although there are various DA methods: simple geometric transformation style DA for single data and synthetic style for multiple data, the Octave Mix is a synthetic-style DA method. After applying a low-pass filter and a high-pass filter to two sensor data, the two are combined by intersecting low- and high-frequency data. The weighted sum of both combined data is calculated as well. Finally, the Octave Mix is used together with the existing DA methods based on geometric transformations. Thus, this study's main contribution is the proposal of the following three methods:

- **Octave Mix method**: we propose a new synthetic-style DA method for sensor data based on frequency decomposition.
- **DA ensemble method**: is a deep learning model that ensembles multiple DA methods and investigate the optimal combination of DA for HAR.
- **DA Revisited for fixed feature extractor (DAR-FFE)**: a training algorithm for the proposed ensemble model that applies pre-training to enhance the DA's effectiveness.

Additionally, the experimental results demonstrate the following important findings:

- The synthetic-style DA methods (mixup [5] and the RICAP [6]) work well for the HAR, but the Octave Mix outperforms them.
A high accuracy was achieved by combining two DA policies: Octave Mix with Rotation, and mixup with Rotation.

II. RELATED WORKS

A. SENSOR-BASED HAR

Most previous research works in HAR are based on extracting human-designed features (hand-crafted features) from sensor values and classifying the activities using machine learning algorithms: support vector machine and random forest [7]–[20].

Different papers using deep learning, end-to-end learning methods, including feature extractor as a trainable network, have been published on HAR. Many of these papers applied convolutional neural networks to sensor-based HAR tasks with model architecture based on several convolution and pooling layers before fully-connected layers [21]–[26]. A method combining multiple sensors [27], methods using recurrent layer after several convolution and pooling layers [28], [29], and more advanced methods: introducing inception, residual, and attention modules [30]–[34], have been proposed. These studies focus on the optimal model architecture for sensor-based HAR tasks without discussing data argumentation (DA).

On the other hand, some studies adopted DAs for sensor-based HAR using deep learning [35]–[39] but limited to the following DAs:

• Rotation; rotates x, y, and z axes,
• Permutation; swaps sections in time series,
• Scaling; scales waveform to amplitude direction,
• Time-warping; scales waveform to time direction,
• Magnitude-warping; multiplies smooth curve,
• Jittering; adds noise, and
• Cropping; masks a section.

Therefore, DAs’ discussion in HAR is limited to simple geometric transformations without considering synthetic-style DAs’ effectiveness.

B. SYNTHETIC-STYLE DA

According to Shorten and Khoshgoftaar [3], the image DAs are classified into “basic image manipulations” and “deep learning approaches,” and there is also “meta-learning” to explore optimal DAs metaphorically. The DAs used in HAR falls under “geometric transformations” or “random erasing” of “basic image manipulations.” However, only a small part of the DA methods studied in image recognition is being used in HAR.

In this study, we focus on synthetic-style DA methods that have not been applied in the previous HAR methods. Although the synthetic-style DA methods are applied in image recognition: mixup [5] and RICAP [6], the mixup [5] method combines multiple training data. For example, given two labeled data, \((x_1, y_1)\) and \((x_2, y_2)\), the method generates synthetic data \((\tilde{x}, \tilde{y})\) using (1).

\[
\tilde{x}_{\text{mixup}} = \lambda x_1 + (1 - \lambda) x_2
\]

\[
\tilde{y}_{\text{mixup}} = \lambda y_1 + (1 - \lambda) y_2
\]

(1)

where \(\lambda \sim \text{Beta}(\alpha, \alpha)\), for \(\alpha \in (0, \infty)\), and \(\lambda \in [0, 1]\). The \(\alpha\) is a hyperparameter of the mixup. From (1), it can be said that the mixup method combines two inputs and outputs by weighted average using a random weight value \(\lambda\) based on beta distribution. The important point of mixup is that not only the input \(x\) but also the output \(y\) are combined by weighted average. Data that do not exist in the training data, which are the middle of the two data, are generated with middle labels.

The RICAP [6] is a method that generate a composite image by cutting out randomly determined rectangular regions from each of the four training image data and then combine them side by side. Like the mixup, the output \(y\)
is synthesized using a weighted average, depending on the cut-out rectangular regions’ size.

C. ADVANCED DA APPROACHES

Shorten and Khoshgoftaar [3] described “deep learning approaches” that use deep learning methods to perform DA, such as a method for augmenting data by automatic data generation [40] and a method for augmenting data by style transformation [41]. Recently, meta-learning methods, such as AutoAugment [42], which uses reinforcement learning to search for the best strategy from multiple DA methods, RandAugment [43], and adversarial AutoAugment [44] that uses adversarial training, have been proposed. Also, a technique called AugMix that improved robustness by synthesizing multiple DAs was proposed in [45].

Advanced DA approaches explore combinations of the DA methods in “basic image manipulations,” so it is important to develop new methods for the “basic image manipulations.” This paper proposes a novel synthetic-style DA method with an optimal ensemble method and a training algorithm for sensor-based HAR.

III. PROPOSED METHOD

A. OUTLINE

The proposed method comprises three components: the Octave Mix, a new synthetic-style DA method; the DA Ensemble Model, which uses feature extractors trained by multiple DAs together; and the DAR-FFE, which pre-trains using DAs and additionally trains only the classifier part without DAs. Figure 2 illustrates the diagram of the proposed method with individual components described in subsequent sections.

![Figure 2. Model architecture and training procedure of the proposed method. Pre-training phase \( (E_1, E_2, C_1, C_2, DA_1, DA_2) \): train each network using two data augmentation policies (black solid line). Classifier-training phase \( (E_1, E_2, C(\text{None}) \): train a new classifier \( C \) without DA while freezing feature extractors \( (E_1, E_2) \) (orange dashed line). Prediction phase \( (E_1, E_2, C(\text{None}) \): predict activity labels (orange dashed line).](image)

B. OCTAVE MIX (OctMix)

The Octave Mix is inspired by Octave Convolution [46], which performs convolution after decomposing low- and high-frequency components and applies it to synthetic-style DA. Algorithm 1 illustrates the Octave Mix algorithm for mini-batch input. First, a low-pass filter (LPF) and a high-pass filter (HPF) are applied to the input to decompose the low-frequency component LPF\( (x) \) and the high-frequency component HPF\( (x) \). The LPF\( (x) \) and the HPF\( (x) \) with randomized order are combined. The resulting two composite waveforms are combined using a weighted sum with \( \lambda \) as the coefficient weight.

Thus, given two labeled data \( (x_1, y_1), (x_2, y_2) \), the data \( (\tilde{x}_{\text{octmix}}, \tilde{y}_{\text{octmix}}) \) generated by this algorithm are formulated using (2).

\[
\tilde{x}_{\text{octmix}} = \lambda LPF(x_1) + (1 - \lambda) HPF(x_2),
\]

\[
\tilde{y}_{\text{octmix}} = \lambda y_1 + (1 - \lambda) y_2
\]

The Octave Mix’s main idea is to perform frequency decomposition before synthesizing \( x \) and \( y \) as well as the mixup. Fig. 1 shows an example of generating synthetic waveforms by Octave Mix using the data \( x_1 = \text{“walking”} \) and \( x_2 = \text{“jogging”} \). The waveform of \( x_1 \), a walking data, has low-amplitude and low-frequency. When LPF and HPF are applied to \( x_1 \), the waveform is decomposed into a smooth walking waveform and a noise-like vibration. Similarly, focusing on \( x_2 \), the low-frequency component’s amplitude and frequency are slightly higher than that of \( x_1 \), and the amplitude of the high-frequency component is increased generally. By intersecting and combining these two waveforms, a waveform that looks like the vibration of the high-frequency component of jogging is added to the low-frequency component of walking, and a waveform that looks like the vibration of the high-frequency component of jogging is added to the low-frequency component of jogging is generated. Finally, the two waveforms are combined by a weighted average to produce a waveform according to the weight coefficient \( \lambda \).

There are two hyperparameters in the Octave Mix: the first is \( \alpha \), which is used to determine the weight value \( \lambda \sim \text{Beta}(\alpha, \alpha) \). Following the mixup, the parameters of

---

**Algorithm 1 Octave Mix Algorithm for Mini Batch**

**Input:** A mini-batch training set \((X, Y) = ((x_i^1, y_i^1), ..., (x_i^n, y_i^n))\);

1. \( X^\text{low} \leftarrow \text{LPF}(X, f_c) \)
2. \( X^\text{high} \leftarrow \text{HPF}(X, f_c) \)
3. \( I \leftarrow \{i\}_{i=1}^n \)
4. Shuffle the indices \( I \)
5. \( \lambda \sim \text{Beta}(\alpha, \alpha) \)
6. for \( i = 1 \) to \( n \) do
7. \( j \leftarrow I_i \)
8. \( g_1 \leftarrow x_i^\text{low} + x_j^\text{high} \)
9. \( g_2 \leftarrow x_i^\text{low} + x_j^\text{high} \)
10. \( \tilde{x}_i \leftarrow \lambda g_1 + (1 - \lambda) g_2 \)
11. \( \tilde{y}_i \leftarrow \lambda y_i + (1 - \lambda) y_j \)
12. end for

**Output:** \( \tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_i), \tilde{Y} = (\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_i) \)
the beta distribution are unified as \( \alpha \). The second is the cutoff frequency of \( f_c \) of LPF and HPF. Particularly in HAR, the change in the main frequency depends on the types of activities to be recognized. Therefore, it is necessary to adjust \( f_c \) according to the task to be applied.

Defining each synthetic waveform as \( g_1, g_2 \), the Octave Mix synthesis can be transformed into the (3).

\[
g_1 = \text{LPF}(x_1) + \text{HPF}(x_2),
g_2 = \text{LPF}(x_2) + \text{HPF}(x_1),
\]

\[
\bar{x}_{\text{octmix}} = \lambda g_1 + (1 - \lambda)g_2,
\]

\[
\bar{y}_{\text{octmix}} = \lambda y_1 + (1 - \lambda)y_2
\]

The (3) is equivalent to (1) of mixup when \( g_1 = x_1, g_2 = x_2 \). This means that the Octave Mix process is the same as mixup when the cutoff frequency \( f_c \) increases. Therefore, Octave Mix is an extension of the mixup.

**C. ENSEMBLE AUGMENTATION MODEL ARCHITECTURE**

Using Octave Mix as a DA strategy, we propose an ensemble model architecture in which multiple DAs acquire multiple feature representations. Fig. 2 represent the model architecture, in which the upper path is a general deep learning model. Here, \( DA_1, DA_2 \) are different DA strategies, \( E_1, E_2 \) are feature extractors, and \( C_1, C_2, C \) are classifiers. Since the feature extractors and classifiers’ internal architecture is not restricted, various architectures such as VGG [47] and ResNet [48] are supported. For example, in the case of VGG, the convolution and pooling layer up to just before flatten is used as the feature extractor, and the remaining fully-connected layer is used as the classifier. Therefore, the proposed method is an ensemble method in which two types of feature extractors with two different DA strategies are trained separately. Although the proposed method outputs two predictions at the training phase, two feature extractors’ outputs are combined to output a single prediction result at the prediction phase using a different classifier \( C \) (orange dashed line).

The ensemble model is based on the idea that two different DA strategies contribute to acquiring other feature representations. Since \( DA_1 \) and \( DA_2 \) must use different approaches, we adopted two DA strategies: Octave Mix after Rotation as \( DA_1 \) and mixup after Rotation as \( DA_2 \).

**D. DA REVISITED FOR FIXED FEATURE EXTRACTOR**

He et al. [49] proposed that performing powerful DA such as mixup and AutoAugment can emphasize a gap between the original data and the augmented data. To address this issue, they proposed DA Revisited, which trained the model for \( N \) epochs on the augmented data, and then additionally trained the model for \( M \) epochs on the clean data.

Inspired by the DA Revisited, we propose a DA Revisited with fixed feature extractor (DAR-FFE), in which the \( E_1, E_2, C_1, C_2 \) in Fig. 2 is trained using augmented data and only \( C \) is trained using the original data without DA. Although it is not discussed in [49], additional M-epochs training of the feature extractor with clean data may lead to the loss of feature representations due to DA variations. We trained the feature extractors \( (E_1, E_2) \) using DA at the pre-training, and trained only the combined classifier \( C \) using clean data at the additional training with the weights-fixed feature extractors \( (\hat{E}_1, \hat{E}_2) \). Algorithm 2 shows the training procedure for the model ensemble \( K \) types of DAs.

**Algorithm 2 Training DA K-Ensemble Model by DAR-FFE**

**Input:** Training dataset \((X, Y)\); \( K \) feature extractors \( \{E_1, E_2, \ldots, E_K\} \); \( K + 1 \) classifiers \( \{C, C_1, C_2, \ldots, C_K\} \);

\( K \) DA policies \( \{DA_1, DA_2, \ldots, DA_K\} \); Pre-training and classifier-training epochs \( N \) and \( M \);

1: for \( k = 1 \) to \( K \) do
2: for epoch = 1 to \( N \) do
3: \((\tilde{X}_k, \tilde{Y}_k) \leftarrow DA_k(X, Y)\)
4: Training \( E_k, C_k \) on augmented training dataset \((\tilde{X}_k, \tilde{Y}_k)\)
5: end for
6: end for
7: \((\hat{E}_1, \hat{E}_2, \ldots, \hat{E}_K) \leftarrow \text{Fixing the weights of all feature extractors} \{E_1, E_2, \ldots, E_K\}\)
8: for epoch = 1 to \( M \) do
9: Training \( C \) using fixed feature extractors \( \{\hat{E}_1, \hat{E}_2, \ldots, \hat{E}_K\} \) on original training dataset \((X, Y)\)
10: end for

**Output:** A trained model \( \{E_1, E_2, \ldots, E_K, C\} \)

**IV. EXPERIMENTS**

**A. EXPERIMENTAL SETTINGS**

1) DATASETS

Table 1 presents the four public datasets: HASC [50], PAMAP2 [13], UCI Smartphone [14], and UniMiB SHAR [19]). Both are benchmark datasets for HAR using sensor data.

The HASC [50] is a benchmark dataset for basic HAR using smartphone sensors. It has accelerometer and gyroscope measurements labeled with six basic activities (staying, walking, jogging, skipping, going upstairs, and going downstairs). As pre-processing, we selected data with a sampling frequency of 100 Hz from BasicActivity of the corpus from 2011 until 2013 and used only the accelerometer’s raw data after trimming were used.

The PAMAP2 [13] is a benchmark dataset in which three wireless inertial measurement units were worn on the chest,
TABLE 1. Details of dataset for evaluation. Sensor type A, G, and M denotes acceleration sensor, gyroscope, and magnetic sensor, respectively, while “?” represents batch size.

| Cite | Dataset        | Number of subjects in train | Number of subjects in valid | Number of subjects in test | position | Sensor type | axes | sampling | Shape of input | Shape of output |
|------|----------------|----------------------------|-----------------------------|---------------------------|----------|-------------|------|----------|----------------|----------------|
| [50] | HASC           | 10 | 50 | 50 | free | A, A2, G, M | x, y, z | 100 Hz | (?, 256, 3) | (?, 6) |
| [13] | PAMAP2         | 4 | 2 | 2 | chest, wrist, ankle | A, A2, G, M | x, y, z | 100 Hz | (?, 256, 36) | (?, 12) |
| [14] | UCI Smartphone | 10 | 10 | 10 | belt / preferred | A, G | x, y, z | 50 Hz | (?, 128, 6) | (?, 6) |
| [19] | UniMiB SHAR    | 10 | 10 | 10 | pocket | A, G | x, y, z | 50 Hz | (?, 151, 3) | (?, 17) |

The UniMiB SHAR [19] is a benchmark dataset using smartphone sensors for HAR. This dataset collected 30 subjects’ acceleration sensor data labeled with 17 activities (standing up from sitting, standing up from laying, walking, jogging, jumping, going upstairs, going downstairs, lying down from standing, sitting down, generic falling forward, falling backward, generic falling backward, hitting an obstacle in the fall, falling with protection strategies, falling backward-sitting-chair, falling leftward, and syncope). This dataset was divided into 151 samples each. From the above, the shape of the input data is winsize = 151, channels = 3 (x, y, z).

2) MODEL TRAINING

Our proposed method in Fig. 2 can be applied to any deep learning model architecture because the method adopts a general CNN model as a feature extractor and classifier. Fig. 3 illustrates the VGG architecture we adopted. Although the original VGG architecture [47] is effective for sensor-based HAR [51], we changed the flatten layer to global average pooling layer and changed the fully-connected layer of C to a single layer in this study to reduce the effect of C.

We trained the model using Adam [52] for 300 epochs with a learning rate of $\eta = 0.001$. The training converged within 300 epochs. We applied DA to the input data ($X, Y$) with a probability of 50% to obtain ($\tilde{X}, \tilde{Y}$), and then combined ($X, \tilde{Y}$) and ($\tilde{X}, Y$) as input.

3) METRICS

Gholamiangonabadi et al. [53] proposed that dataset be divided by subjects while evaluating sensor-based HAR.

Assuming that labeled sensor data of prediction-target user could not be obtained in the real use case, we performed the evaluation using subject-base-hold-out validation, which divided the dataset into training, validation, and testing by subjects. Table 1 shows the number of subjects included in each dataset. The subjects are selected by random sampling: (1) sampling of subjects, (2) dividing the dataset, and (3) training and accuracy evaluation of each method are considered as a single trial. The estimation accuracy between methods is compared using the average of the results of 10 different trials. The average f-score was used as an evaluation index because there was a bias in the number of data between labels in some datasets.

B. PARAMETER TUNING OF DA

This study aims to verify the effectiveness of synthetic-style DA methods and develop new techniques. We compared the proposed method with typical synthetic-style DA methods: mixup [5] and RICAP [6]. Since the DA method is based on simple geometric transformations, we applied Rotation, which is effective for HAR [35]. We adopted the technique to combine two waveforms back and forth in the time series direction as a RICAP procedure. Similarly, Mixup was a weighted average of two waveforms.

First, we used the HASC dataset to tune the hyperparameters of each DA method: $\alpha$, the parameter of the beta distribution in mixup, and $\beta$, the parameter of the beta distribution in RICAP. We performed the same experiments with two datasets: one as the training dataset and the other as the validation dataset.

FIGURE 3. Experimental model architecture. The feature extractor part $E$ including five ConvBlocks outputs (?, 512) shape feature map via a global average pooling layer, where the “?” denotes batch size. The classifier part $C$ is composed a single fully-connected layer that outputs (?, class) shape softmax values as one-hot vector.
Table 2 shows the results of hyperparameter tuning used to determine the weights during synthesis, was used for mixup and RICAP. The $\alpha$ plus the cutoff frequency $f_c$ were the hyperparameters tuned for the Octave Mix. The evaluation of the tuning was based on the accuracy of the validation data.

Table 2 shows the results of hyperparameter tuning. The upper five methods trained the simple model without DA (None) and with rotation DA (Rotation), mixup after rotation (Rot.& mixup), RICAP after rotation (Rot.& RICAP), and Octave Mix after rotation (Rot.& OctMix). Ours is a method that combined three components (Octave Mix, ensemble model, and DAR-FFE). First, the Rotation improved the estimation accuracy by 7.4% from None. The mixup and RICAP improved the estimation accuracy by 1.0% for $\alpha = 5.0$. Further, the Octave Mix improved the estimation accuracy by 2.0% from the Rotation with $\alpha = 0.5, f_c = 2.1$. We compared the synthetic-style DA methods mixup with RICAP, and the proposed Octave Mix further improved the accuracy. As for the hyperparameters, the effect of $\alpha$ was not so large, and $f_c$ made no much difference if greater than 1. We discuss the mixup and RICAP with $\alpha$ set to 5.0, and Octave Mix with $\alpha = 0.5, f_c = 2.1$ experiments in subsequent sections.

### TABLE 3. Results comparison on accuracy estimation for the four datasets (accuracy and average f-score for test set [%]).

| DA          | HASC | PAMAP | UCI Smartphone | UniMiB SHAR |
|-------------|------|-------|----------------|-------------|
|             | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score |
| None        | 68.1(±4.0) | 68.1(±4.1) | 52.8(±12.1) | 48.2(±11.5) | 80.3(±13.5) | 78.7(±17.4) | 60.3(±3.3) | 51.1(±3.5) |
| Rotation(Rot.) | 76.6(±1.9) | 76.9(±1.9) | 61.8(±11.4) | 59.8(±10.2) | 88.2(±4.6) | 88.3(±4.9) | 68.1(±2.9) | 57.5(±2.6) |
| Rot.&mixup  | 77.7(±2.9) | 77.6(±3.0) | 66.1(±9.8)  | 64.6(±10.4) | 89.7(±3.2) | 90.1(±2.9) | 72.0(±2.6) | 61.9(±2.6) |
| Rot.&RICAP  | 77.8(±2.1) | 77.9(±2.0) | 63.3(±10.1) | 60.7(±9.8)  | 90.0(±3.1) | 90.5(±2.9) | 69.2(±2.0) | 58.4(±2.9) |
| Rot.&OctMix | 79.3(±4.5) | 79.5(±4.4) | 62.0(±11.0) | 59.0(±10.8) | 89.7(±2.4) | 90.1(±2.3) | 67.9(±2.6) | 58.7(±2.3) |
| Ours        | 80.9(±1.5) | 81.0(±1.5) | 66.5(±10.6) | 64.7(±10.5) | 90.8(±3.2) | 91.2(±3.0) | 75.1(±1.9) | 65.4(±2.4) |

### TABLE 4. Results of ablation study using HASC dataset (accuracy and average f-score for test set [%]).

| Methods       | Accuracy | F-score | Note                  |
|---------------|----------|---------|-----------------------|
| (1) Rotation  | 68.1(±4.0) | 68.1(±4.1) | Without DA |
| (2) Rot. & OctMix | 76.6(±1.9) | 76.9(±1.9) | With Rotation |
| (3) Rot. & RICAP | 79.3(±1.4) | 79.5(±1.4) | With Rot. & OctMix |
| (4) Rot. & OctMix | 76.1(±2.6) | 76.2(±2.7) | Ensemble Rot. & RICAP and Rot. & Mixup |
| (5) Octave Mix | 73.3(±1.9) | 77.6(±1.9) | DAR-FFE from (2) |
| (6) Octave Mix | 74.7(±2.8) | 74.8(±3.3) | Ensemble Rot. & OctMix and Rot. & Mixup |
| (7) Octave Mix | 79.9(±1.9) | 80.1(±1.9) | DAR-FFE from (3) |
| (8) Octave Mix | 80.0(±2.0) | 80.2(±2.0) | DAR-FFE from (4) |
| (9) Octave Mix | 80.9(±1.5) | 81.0(±1.5) | DAR-FFE from (6) |
parameters were the same as the simple methods, this method improved the accuracy by 3.2% in the f-score from Rotation alone and by 0.6% from Rot.&OctMix. This was equivalent to the accuracy, where only the classifier was additionally trained after ensembling RICAP and Mixup. The proposed method further improved the accuracy by about 1%.

E. EFFECTS ON THE AMOUNT OF TRAINING DATA

Fig. 4 shows the change in the average f-score when the number of subjects in the training data is changed. The proposed method demonstrates effective results regardless of the amount of training data. Additionally, the difference between Ours and Rot.&OctMix was relatively small when the number of subjects is small (less than 10 persons). The difference became more significant as the number of subjects increases. In other words, Octave Mix’s effect was high when the amount of data is small, and the impact of ensemble became higher as the number of subjects increases.

![Figure 4. Effect of changing the number of subjects in the training set (average f-score for test set [%]).](image)

F. COMPARISON K-ENSEMBLE

Table 5 shows our proposed method’s experimental results with different DA combinations. The method is an ensemble method of Octave Mix and Mixup (b) in the table. The proposed method’s estimation accuracy is higher than that of the other combinations: (a) and (c). Besides, it could increase the number of DA combinations to improve accuracy. We evaluated the method combining three DA policies (d), but there was almost no change in accuracy starting from (b). Combining two DA policies were sufficient for the DA variation adopted in this study. Combining any other DA methods to acquire a variety of feature representation could also improve the accuracy by increasing the number of combinations.

V. CONCLUSION

This paper proposed a novel DA Octave Mix method for sensor-based HAR, a method for ensembling the DAs, and additional training on the original data (DAR-FFE). The method combines multiple input data and improves the conventional mixup method using frequency decomposition. The experiments demonstrate that the three components of our proposed method: Octave Mix, ensemble model, and DAR-FFE, could improve the estimation accuracy of the HAR. We hope to apply the three components of our approach to various problems and problem settings in different future fields.

| REFERENCES |
|---|
| [1] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Commun. Surveys Tuts., vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013. |
| [2] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deeplearning for sensor-based activity recognition: A survey,” Pattern Recognit. Lett., vol. 119, no. 3, pp. 3–11, Mar. 2019. |
| [3] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” J. Big Data, vol. 6, no. 1, pp. 1–48, Jul. 2019. |
| [4] S. Inoue, N. Ueda, Y. Nohara, and N. Nakashima, “Recognizing and understanding nursing activities for a whole day with a big dataset,” J. Inf. Process., vol. 24, no. 6, pp. 853–866, Nov. 2016. |
| [5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in Proc. ICLR, Apr. 2018, pp. 1–15. |
| [6] R. Takahashi, T. Matsubara, and K. Uehara, “RICAP: Random image cropping and patching data augmentation for deep CNNs,” Proc. Mach. Learn. Res., vol. 95, pp. 786–798, Apr. 2018. |
| [7] K. Kiani, C. J. Snijders, and E. S. Gelsema, “Recognition of daily life motor activity classes using an artificial neural network,” Arch. Phys. Med. Rehabil., vol. 79, no. 2, pp. 147–154, Feb. 1998. |
| [8] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in Proc. PerCom, 2004, pp. 1–17. |
| [9] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in Proc. IAAI, 2005, pp. 1541–1546. |
| [10] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Crestata, and J. D. R. Millán, “Collecting complex activity datasets in highly rich networked sensor environments,” in Proc. INSS, 2010, pp. 233–240. |
| [11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SIGKDD Explor. Newslett., vol. 12, no. 2, pp. 74–82, Dec. 2010. |
| [12] P. Zappi, D. Roggen, E. Farella, G. Tröster, and L. Benini, “Network-level power-performance trade-off in wearable activity recognition: A dynamic sensor selection approach,” ACM Trans. Embedded Comput. Syst., vol. 11, no. 3, pp. 1–30, 2012. |
| [13] A. Reiss and D. Stricker, “Creating and benchmarking a new dataset for physical activity monitoring,” in Proc. PETRA, 2012, pp. 40:1–40:8. [Online]. Available: http://doi.acm.org/10.1145/2413097.2413148 |
| [14] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in Proc. ESANN, Apr. 2013, pp. 437–442. |
| [15] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” ACM Comput. Surv., vol. 46, no. 3, pp. 1–33, Jan. 2014. |
| [16] D. Banos, C. Villalonga, R. Garcia, A. Saiz, M. Damas, J. A. Holgado, S. Lee, H. Pomaers, and I. Rojas, “Design, implementation and validation of a novel open framework for agile development of mobile health applications,” Biomed. Eng. OnLine, vol. 14, no. 2, pp. 1–20, Aug. 2015. |

VOLUME 9, 2021

53685
M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, “Multi-modal convolutional neural networks for multimodal wearable activity recognition,” in Proc. MobileHC, 2015, pp. 9–14.

M. Shaob, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, “Complex human activity recognition using smartphone and wrist-worn motion sensors,” Sensors, vol. 16, no. 4, pp. 1–24, 2016. [Online]. Available: http://www.mdpi.com/1424-8202/16/4/426

D. Micucci, M. Mobilo, and P. Napoletano, “UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones,” Appl. Sci., vol. 7, no. 10, p. 1101, Oct. 2017. [Online]. Available: http://www.mdpi.com/2076-3417/10/10/1101

R.-A. Voicu, C. Dobrec, L. Bajenaru, and R.-I. Ciobanu, “Human physical activity recognition using smartphone sensors,” Sensors, vol. 19, no. 3, p. 458, Jan. 2019. [Online]. Available: http://www.mdpi.com/1424-8202/19/3/458

M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, “Convolutional neural networks for human activity recognition using mobile sensors,” in Proc. MobiCASE, Nov. 2014, pp. 197–205.

Y. Chen and Y. Xue, “A deep learning approach to human activity recognition based on single accelerometer,” in Proc. IEEE SMC, Oct. 2015, pp. 1488–1492.

J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in Proc. IJCAI, Jul. 2015, pp. 3995–4001.

S. Ha, J.-M. Yun, and S. Choi, “Multi-modal convolutional neural networks for activity recognition,” in Proc. IEEE SMC, Oct. 2015, pp. 3017–3022.

H. Gjoreski, B. Bizjak, M. Gjoreski, and M. Gams, “Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer,” in Proc. IJCAI, Jul. 2016, pp. 1–7.

J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gaßmann, J. Klucken, and H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams, “Comparing deep and feature learning methods for human activity recognition,” in Proc. IEEE SMC, 2015, pp. 85–93, Jan. 2017.

Z. Yang, O. I. Raymond, C. Zhang, Y. Wen, and J. Long, “DFTerNet: Toward 2-bit dynamic precision networks for accurate human activity recognition,” IEEE Access, vol. 6, pp. 56750–56674, 2018.

F. J. Ordóñez and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” Sensors, vol. 16, no. 1, pp. 1–25, Jan. 2016.

F. Li, K. Shirahama, M. A. Nisar, L. Köping, and M. Grzegorzek, “Comparison of feature learning methods for human activity recognition using wrist accelerometer,” in Proc. IJCAI, Jul. 2016, pp. 1–7.

Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, “Deep residual bidir-LSTM for human activity recognition using wearable sensors,” Math. Problems Eng., vol. 2018, Dec. 2018, Art. no. 7316954. [Online]. Available: https://www.hindawi.com/journals/mpe/2018/7316954/cit/

M. Dong, J. Han, Y. He, and X. Jing, “HAR-Net: Fusing deep representation and hand-crafted features for human activity recognition,” in Proc. ICSINC, Apr. 2019, pp. 32–40.

J. Long, W. Sun, Z. Yang, and O. I. Raymond, “Asymmetric residual neural network for accurate human activity recognition,” Information, vol. 10, no. 6, pp. 1–19, Jun. 2019.

C. Xu, D. Chai, H. Je, X. Zhang, and S. Duan, “InnoHAR: A deep neural network for complex human activity recognition,” IEEE Access, vol. 7, pp. 9893–9902, Jan. 2019.

K. Wang, J. He, and L. Zhang, “Attention-based convolutional neural network for weakly labeled human activities’ recognition with wearable sensors,” IEEE Sensors J., vol. 19, no. 17, pp. 7598–7604, Sep. 2019.

T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, “Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks,” in Proc. ICMI, Nov. 2017, pp. 216–220, doi: 10.1145/3136755.3136817.

O. S. Eyoob and D. S. Han, “Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM network neural,” Sensors, vol. 18, no. 9, pp. 1–26, Aug. 2018.

G. Kalouros, E. I. Zacharakis, and V. Megaloukonomou, “Improving CNN-based activity recognition by data augmentation and transfer learning,” in Proc. INDIN, vol. 1, Jul. 2019, pp. 1387–1394.

A. Z. M. Faridee, M. A. A. H. Khan, N. Pathak, and N. Roy, “AugToAct: Scaling complex human activity recognition with few labels,” in Proc. MobiQuitous, Nov. 2019, pp. 162–171, doi: 10.1145/3360774.3360831.

K. M. Rashid and J. Louis, “Times-series data augmentation and deep learning for construction equipment activity recognition,” Adv. Eng. Informat., vol. 42, Oct. 2019, Art. no. 100944. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474043619300886

S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici, “DOPING: Generative data augmentation for unsupervised anomaly detection with GAN,” in Proc. ICDM, Nov. 2018, pp. 1122–1127.

P. T. G. Jackson, A. Atapour-Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, “Style augmentation: Data augmentation via style randomization,” in Proc. CVPRW, Jun. 2019, pp. 83–92.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in Proc. CVPR, Jun. 2019, pp. 113–123.

E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in Proc. CVPRW, Jun. 2020, pp. 3008–3017.

X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, “Adversarial data augmentation,” in Proc. ICLR, Apr. 2020, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=ByxUfLYSKs

D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “Augmix: A simple method to improve robustness and uncertainty under data shift,” in Proc. ICLR, Apr. 2020, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=r1gmxHFvB

Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, Y. Shuicheng, and J. Feng, “Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolutions,” in Proc. ICCV, Oct. 2019, pp. 3434–3443.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, May 2015, pp. 1–14.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. CVPR, Jun. 2016, pp. 770–778.

Z. He, L. Xie, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Data augmentation revisited: Rethinking the distribution gap between clean and augmented data,” pp. 1–10, Nov. 2019, arXiv:1909.09148. [Online]. Available: http://arxiv.org/abs/1909.09148

N. Kawaguchi, N. Nishio, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara, and Y. Sumi, “HASC challenge: Gathering large scale human activity corpus for the real-world activity understandings,” in Proc. AH, Mar. 2011, pp. 1–5.

T. Hasegawa and M. Koshino, “Representation learning by convolutional neural network for smartphone sensor based activity recognition,” in Proc. CIIS, Nov. 2019, pp. 99–104.

D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, May 2015, pp. 1–15.

D. Gholamianpanah, N. Kiselov, and K. Groeller, “Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection,” IEEE Access, vol. 8, pp. 133982–133994, Jul. 2020.