Sample Fusion Network: An End-to-End Data Augmentation Network for Skeleton-based Human Action Recognition

Fanyang Meng, Hong Liu, *Member, IEEE,* Yonghseng Liang, Juanhui Tu and Mengyuan Liu

***Abstract*—Data augmentation is a widely used technique for enhancing the generalization ability of deep neural networks for skeleton-based human action recognition (HAR) tasks. Most existing data augmentation methods generate new samples by means of handcrafted transforms. However, these methods often cannot be trained and then are discarded during testing because of the lack of learnable parameters. To solve those problems, a novel type of data augmentation network called a sample fusion network (SFN) is proposed. Instead of using handcrafted transforms, an SFN generates new samples via a long short-term memory (LSTM) autoencoder (AE) network. Therefore, an SFN and an HAR network can be cascaded together to form a combined network that can be trained in an end-to-end manner. Moreover, an adaptive weighting strategy is employed to improve the complementarity between a sample and the new sample generated from it by an SFN, thus allowing the SFN to more efficiently improve the performance of the HAR network during testing. Experimental results on various datasets verify that the proposed method outperforms state-of-the-art data augmentation methods. More importantly, the proposed SFN architecture is a general framework that can be integrated with various types of networks for HAR. For example, when a baseline HAR model with 3 LSTM layers and 1 fully connected (FC) layer was used, the classification accuracy was increased from 79.53% to 90.75% on the NTU RGB+D dataset using a cross-view protocol, thus outperforming most other methods.**

***Index Terms*—Human Action Recognition, Data augmentation, Autoencoder, LSTM**

I. Introduction

Human action recognition (HAR) [1]–[7] has been a hot topic in computer vision for decades because it can be applied in various fields, e.g., human-computer interaction, game control and intelligent surveillance. Compared with other modalities, such as RGB and depth representation, the skeleton is a high-level representation of human action that is robust to variations in location and appearance. Moreover, rapid advances in imaging technology and the development of a powerful human pose estimation technique based on depth have made skeleton data easily accessible. Therefore, skeleton-based HAR has attracted substantial research attention [8]–[15].

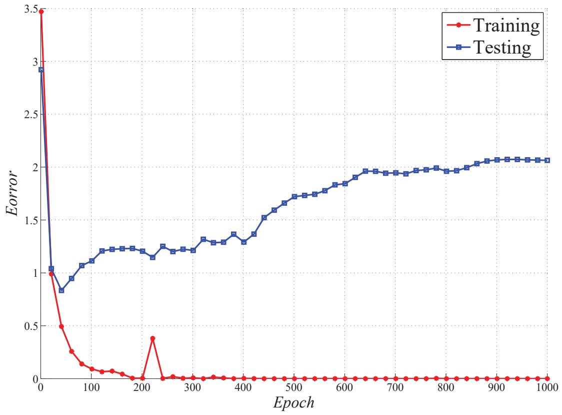


Fig. 1: The curves of the training error and testing error for the NTU RGB+D Cross-subject dataset with the baseline LSTM network, which contains only 3 LSTM layers and 1 FC layer.

Recently, deep learning methods using skeletons have been undergoing rapid development because they can automatically extract spatial-temporal relationships among joints [16]–[23]. Applications of these works have achieved outstanding performance in skeleton-based HAR. However, since skeleton data are far less abundant than RGB data, overfitting has become a very serious problem for deep learning methods, even in shallow networks (as shown in Fig. 1). This problem limits the generalization ability of deep learning methods.

To overcome such limitations, many regularization methods have been proposed [24]–[28]. These methods can be broadly categorized into three groups, namely, loss function regularization, network structure regularization and data augmentation. In contrast to the two other types of regularization methods, data augmentation [29]–[35] focuses on the data level and does not require the design of a new loss function or modification of the network structure. Because of these merits, data augmentation is widely used during the training of deep neural networks to improve their generalization ability.

However, existing data augmentation methods generate new samples by means of handcrafted transforms, the parameters of which cannot be learned. Therefore, these methods cannot be trained along with the training of an HAR network.

Fanyang Meng and Yongsheng Liang contributed equally to this paper and should be regarded as Co-first authors.

Fanyang Meng is with the key Laboratory of Machine Perception, Peking University Shenzhen Graduate School, Shenzhen, Nanshan, 518055, China, and also with the Peng Cheng Laboratory, Shenzhen, Nanshan, 518055, China. E-mail: fymeng@pkusz.edu.cn.

Hong Liu (Corresponding author) and Juanhui Tu are with the key Laboratory of Machine Perception, Peking University Shenzhen Graduate School, Shenzhen, Nanshan, 518055, China. E-mail: {hongliu,juanhuitu}@pku.edu.cn.

Yongsheng Liang is with the Harbin Institute of Technology (Shenzhen), Shenzhen, Nanshan, 518055, China. E-mail: liangys@sziit.edu.cn.

Mengyuan Liu is with the Nanyang Technological University, Singapore. E-mail: nkliuyifang@gmail.com.

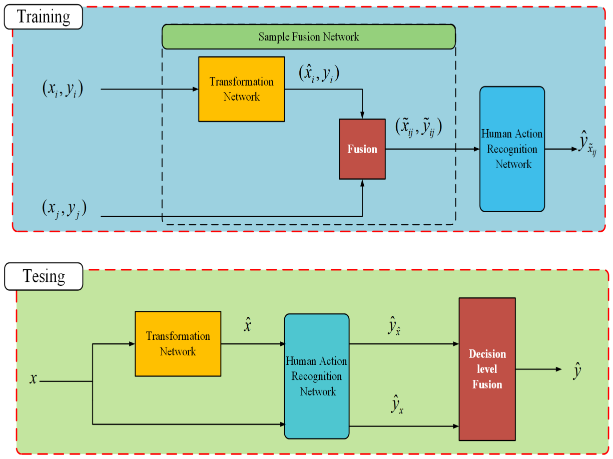


Fig. 2: A flowchart of the sample fusion network used as data augmentation for the HAR network during training and testing.

Meanwhile, since an original sample and the sample generated from it by means of a handcrafted transform are not effectively complementary, they generally cannot be adapted to the subsequent recognition networks to help improve the HAR performance during testing.

In this paper, a novel data augmentation tool called a sample fusion network (SFN) is proposed for skeleton data augmentation. The flowchart of our method is presented in Fig. 2. During training, for a given sample pair randomly selected from the training dataset, whether they are from the same class or not, sample fusion is performed through the SFN to generate a new sample. Then, the new sample is utilized to train the HAR network. During testing, for a given sample, that sample and its corresponding output from the transformation network are separately input into the HAR network, and the outputs of the HAR network are then fused as the final classification result.

An autoencoder (AE) network has also been used for data augmentation in [36], but the AE network presented in [36] is used only to transform the input into the feature space; the new samples generated in the feature space are still generated via traditional data augmentation methods. Consequently, this AE network cannot be trained in an end-to-end manner along with an HAR network during training and also cannot be utilized to improve the performance of the HAR network during testing. Therefore, that method can still be considered to generate new samples by means of handcrafted transforms.

Unlike in the case of handcrafted transforms, whose parameters are selected randomly, an SFN can be cascaded together with an HAR network during training for data augmentation. Due to their cascaded structure, the SFN and HAR network can be trained together in an end-to-end manner. Moreover, an adaptive weighting strategy is employed to improve the complementarity between a sample and the sample generated from it by the SFN. Thus, the SFN can also be utilized to improve the performance of the HAR network by means of decision-level fusion.

The key component of our method is the introduction of a neural network into the data augmentation process, which results in two advantages. First, the data augmentation method can be trained in an end-to-end manner along with an HAR network. This approach improves the effectiveness of data augmentation for HAR networks. Second, during testing, the SFN can be utilized to further improve the performance of the HAR network.

The three major contributions of this work are as follows:

- First, a unified mathematical formulation for data augmentation methods is proposed. Then, the limitations of existing data augmentation methods are analyzed.
- To address the problems with the existing data augmentation methods for skeleton-based HAR networks, an SFN is proposed. The SFN is cascaded together with an HAR network to form a combined network that can be trained in an end-to-end manner and utilized to improve the performance of the HAR network during testing.
- To further enhance the performance of the SFN, we propose an adaptive weighting strategy that is applied to the features of the SFN during training. Then, we extend the SFN to multiple samples and multiple scales to improve the diversity of the generated samples.

Furthermore, to better understand the contributions of the various aspects of our proposed method, we evaluate the impacts of the different components leveraged in our method. We find that on the NTU-CV dataset (the largest existing in-house skeleton dataset obtained under a cross-view protocol), the accuracy of the baseline model (containing only 3 long short-term memory (LSTM) layers and 1 fully connected (FC) layer) is increased from 79.53% without data augmentation to 90.75% with our method, which outperforms all state-of-the-art data augmentation methods.

The rest of this paper is organized as follows. Section II presents a review of related work. Section III introduces the problem formulation, and the optimization procedure is described in Section IV. The experimental results and the performance analysis are reported in Section V, followed by the conclusion in Section VI.

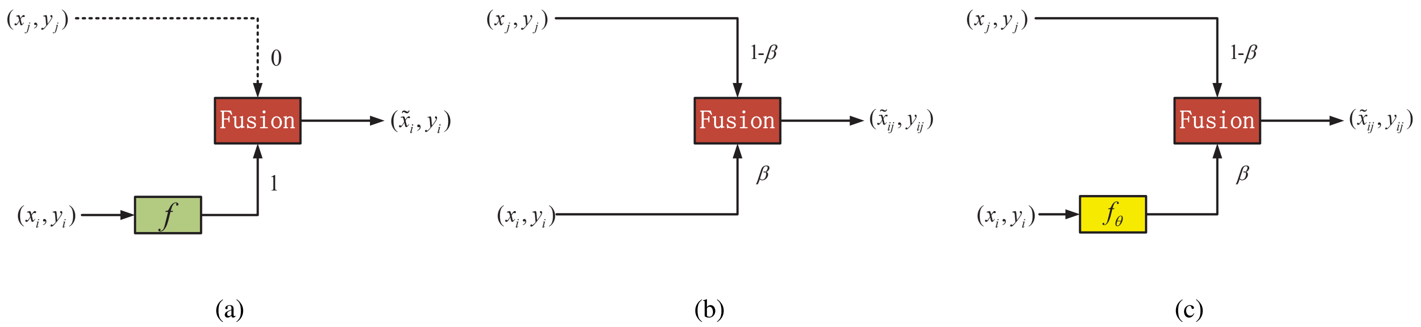
II. Related Works

Since our work addresses an attempt to increase the generalization ability of skeleton-based HAR networks by means of data augmentation, we first review related work on deep learning methods for skeleton-based HAR. Then, the existing data augmentation methods for skeleton-based HAR are briefly introduced.

A. *Skeleton-based HAR model*

According to the architecture of the neural network, deep learning methods can be broadly categorized into three groups: CNN-based methods, RNN-based methods and methods based on other architectures.

RNN-based methods: Skeleton data are used as time-series inputs to an RNN to exploit the temporal information. Du *et al.* [13] proposed an end-to-end hierarchical RNN for encoding the temporal relationships between skeleton joints and divided the skeleton joints into five main groups to extract local features. Veeriah *et al.* [37] proposed a differential gating scheme



(a)

(b)

(c)

Fig. 3: Different data augmentation strategies. (a) Single-sample-based; (b) Multiple-sample-based; (c) SFN.

for an LSTM neural network to emphasize the salient motions between successive frames. To learn the common temporal patterns of different groups of joints independently, Shahroudy *et al*. [38] proposed a part-aware LSTM human action learning model (P-LSTM). Liu *et al*. [39] introduced a spatial-temporal LSTM (ST-LSTM) network for jointly learning both the spatial and temporal relationships among joints. Song *et al*. [40] proposed an end-to-end spatial and temporal attention model that learns to selectively focus on discriminative joints of the skeleton within each frame of the input and focuses different levels of attention on the outputs of different frames.

CNN-based methods: Skeleton sequences are converted into images, thus converting the task of skeleton-based HAR into an image classification task. Therefore, the key question is how to effectively represent spatiotemporal information in the form of image properties, including color and texture. Du *et al*. [13] represented a skeleton sequence as a matrix by concatenating the joint coordinates at each instant and arranging the vector representations in chronological order. Wang *et al*. [22] proposed a method called joint trajectory maps (JTM), in which the trajectories are mapped into the hue, saturation, and value (HSV) space, to encode spatiotemporal information into multiple texture images. Li *et al*. [41] used joint distance maps (JDM) to encode the pairwise distances between the skeleton joints of single or multiple subjects into image textures. Hou *et al*. [10] drew skeleton joints with a specific pen onto three orthogonal canvases and then encoded the dynamic information in the skeleton sequences in color. Liu *et al*. [21] encoded skeletons into a series of color images and then applied visual/motion enhancement methods to the color images to enhance their local patterns. Yan *et al*. [42] proposed a generic graph-based model called a spatial-temporal graph convolutional network (ST-GCN) to automatically learn both spatial and temporal patterns from data.

Methods based on other architectures: Salakhutdinov *et al*. [43] adopted a deep Boltzmann machine (DBM) to learn low-level generic features and high-level correlations among low-level features of the skeleton. Wu and Shao [44] adopted deep belief networks (DBNs) to extract high-level features to represent humans in each frame in 3D space. Ijjina and Krishna Mohan [45] adopted a stacked AE network to learn the underlying skeleton features. Huang *et al*. [15] incorporated the Lie group structure into a deep learning architecture to learn more appropriate Lie group features of skeletons.

Generally, deep learning methods can automatically extract spatial-temporal relationships among joints; such methods have achieved outstanding performance in skeleton-based HAR. However, since skeleton data are far less abundant than RGB data, overfitting problem has become a very serious problem for deep learning methods, even in shallow networks. Thus, a more effective regularization method is required to enhance the generalization ability of deep learning methods.

B. Data Augmentation

Data augmentation aims to enlarge a training dataset by applying various transformations to the existing data. Although many data augmentation methods exist, here we focus only on data augmentation methods that can be used for skeleton data.

According to the number of samples used during data augmentation, existing data augmentation methods for skeleton data can be categorized into two main types: single-sample-based and sample-pair-based methods.

Single-sample-based methods: A single original sample is used to generate new samples. Wang *et al*. [32] proposed rotation, scaling and shear transformations as data augmentation techniques based on 3D transformations to make better use of a limited supply of training data. Ke *et al*. [33] employed cropping to increase the number of samples. Yang *et al*. [34] exploited horizontal flipping as a method of augmenting data without loss of information. Li *et al*. [35] designed various data augmentation strategies, such as random rotation in 3D coordinates, the addition of Gaussian noise and video cropping, to augment the scale of an original dataset.

Sample-pair-based methods: A sample pair is used to generate new samples. For instance, Zhang *et al*. [46] proposed a straightforward data augmentation principle called Mixup, in which trains a neural network is trained on linear combinations of pairs of samples and their labels. To impose constraints on the shape of the feature distributions, Tokozume *et al*. [47] generated between-class images by mixing two images belonging to different classes at a random ratio. Inoue *et al*. [48] designed a simple, yet surprisingly effective, data augmentation technique called SamplePairing, in which new sample is synthesized from one image by overlaying another

TABLE I: Principal notations

(x_i, y_i)	a sample and its label from the training or testing dataset
$\tilde{x}_{ij}$	a new sample generated from x_i and x_j
$\tilde{y}_{ij}$	the label of $\tilde{x}_{ij}$, generated from y_i and y_j
$f(\cdot)$	a transformation function without learnable parameters
$f_\theta(\cdot)$	a transformation function with learnable parameters θ
$\hat{x}$	a new sample generated via a transformation function
$D(\cdot)$	HAR network function
$\hat{y}_x$	the output of the HAR network with input x

image randomly chosen from the training data. However, SamplePairing is not suitable for skeleton data, and the new samples are still generated by means of handcrafted transforms. To improve the reusability and generalization ability of data augmentation methods, DeVries *et al.* [36] proposed a domain-agnostic approach to data augmentation in feature space, in which interpolation and extrapolation are performed in a feature space learned by an AE network. Compared to single-sample-based methods, methods based on sample pairs not only generate more new samples but also enhance the linear relationship between training samples.

Generally, mostly existing data augmentation methods for skeleton data have evolved from methods for image data augmentation. Therefore, these data augmentation methods can effectively enhance the spatial information contained in skeleton data. However, the temporal information contained in skeleton data, which is also very important for skeleton-based HAR, is not effectively considered. Furthermore, since the new samples are generated by means of handcrafted transforms in most of the existing data augmentation methods, these methods cannot be trained in an end-to-end manner and cannot be utilized to further improve the performance of HAR networks during testing.

III. Problem Formulation

For ease of presentation, the main notations are first summarized in Table. I. Then, a unified mathematical formulation for data augmentation methods is proposed, and the limitations of existing data augmentation methods are analyzed. Finally, a new data augmentation framework called an SFN is proposed, along with the details of its design.

A. Problem Formulation

For a given sample pair $((x_i, y_i), (x_j, y_j))$, a new sample $(\tilde{x}_{ij}, \tilde{y}_{ij})$ generated via existing data augmentation methods can be formulated as follows:

$$\begin{aligned} \tilde{x}_{ij} &= \lambda \cdot f(x_i) + (1-\lambda) \cdot x_j \\ \tilde{y}_{ij} &= \lambda \cdot y_i + (1-\lambda) \cdot y_j \end{aligned} \tag{1}$$

where λ is a fusion weight and $\lambda \in [0, 1]$.

During testing, for a given test sample x, the classification result is obtained as follows:

$$\hat{y}_x = D(x)$$

According to the definitions of the fusion weight and the transformation function that appear in Eq. (1), we can deduce that the two types of existing data augmentation methods are as follows:

- $\lambda = 1$ and $\tilde{x} = f(x)$: A new sample is generated by transformation function f. In this case, Eq. (1) is reduces to a single-sample-based data augmentation method. (as shown in Fig. 3(a)). f is artificially defined in most of the existing data augmentation methods.
- $\lambda \sim P$ and $x \equiv f(x)$: A new sample is generated through the linear fusion of the two original samples x_i and x_j. In this case, Eq. (1) corresponds to a sample-pair-based data augmentation method (as shown in Fig. 3(b)).

The existing data augmentation methods, whether based on single samples (**with no parameters to be learned**) or sample pairs (**with no transformation function to be learned**), cannot be trained along with the training of an HAR network. Moreover, these methods cannot be utilized to improve the performance of the HAR network during testing.

To overcome these limitations, a new data augmentation tool called an SFN is proposed (as shown in Fig. 3(c)). The corresponding formulation is defined as follows:

$$\begin{aligned} \tilde{x}_{ij} &= \lambda \cdot f_\theta(x_i) + (1-\lambda) \cdot x_j \\ \tilde{y}_{ij} &= \lambda \cdot y_i + (1-\lambda) \cdot y_j \end{aligned} \tag{2}$$

where f_θ is a transformation function with learnable parameters θ, which can be defined by a neural network. Thus, we can easily cascade the SFN with an HAR network to form a combined network that can be trained in an end-to-end manner.

Then, during testing, the classification result for x is obtained as follows:

$$\hat{y}_x = \beta \cdot D(f_\theta(x)) + (1-\beta) \cdot D(x) \tag{3}$$

where β is a weight coefficient and $\beta \in [0, 1]$.

From Eq. (3), it can be observed that when $\beta = 0$, only the original sample is utilized during testing, and the SFN reduces to a traditional data augmentation method. When $\beta = 1$, only the sample generated via the transformation function is utilized during testing, and the SFN reduces to a preprocessing method. However, when $\beta \in (0, 1)$, the original sample and the corresponding output of the transformation function are fused at the decision level.

Eq. (2) and (3) show that compared with the existing data augmentation methods, an SFN has two advantages. First, since the transformation network is defined by a neural network, the cascaded SFN and HAR network can be easily trained in an end-to-end manner during training. Second, since a sample and its corresponding output of the transformation network usually vary at the pixel level but are equivalent in terms of classification, decision-level fusion can be applied to improve the performance of the HAR network during testing.

The above analysis shows that the SFN has three key components: the designation of the transformation network and the definitions of the fusion weight and the loss function. Therefore, we proceed to present the transformation network in Section III-B, the definition of the fusion weight in Section III-C, and the definition of the loss function in Section III-D. The optimization procedure for multiple-sample fusion is described in Section IV.

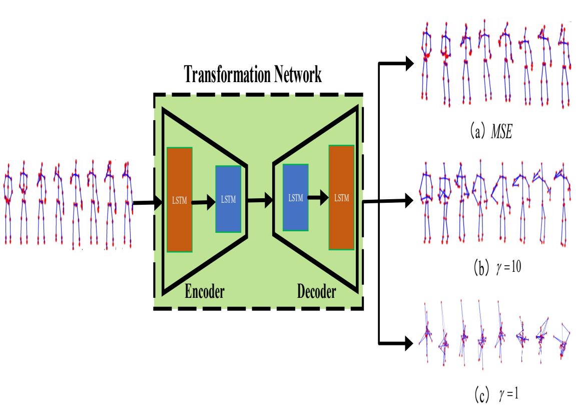


Fig. 4: Diagram of the transformation network.

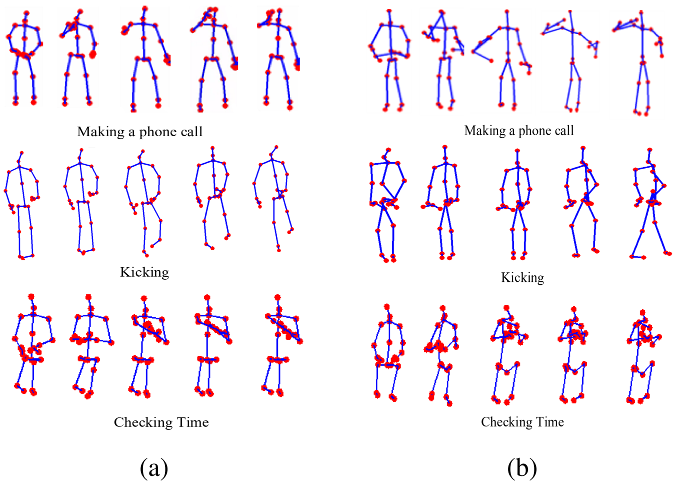


Fig. 5: Visualizations of several action sequences in the NTU RGB+D dataset: (a) the original samples and (b) the reconstructed samples output from the AE network in the SFN (γ=10 in Eq. (9)).

B. Transformation Network

To improve the performance of the HAR network during testing, the output of the transformation network should be different from the corresponding input in terms of its pixel-level representation and the same as the corresponding input in terms of classification. The AE architecture is one of the most common network structures. Because of its data compression ability, the output of an AE network is usually different from the input at the pixel level but contains most of the same useful information as the input; thus, an AE network is suitable for use as a transformation network. Meanwhile, related research [37]–[40] shows that an LSTM network, which can extract meaningful features from skeleton data with only a few layers, can serve as a powerful model for skeleton data.

Based on the above analysis, we build a transformation network with an AE network structure containing four stacked LSTM layers (as shown in Fig. 4). The first two LSTM layers serve as the encoder, and the last two layers serve as the decoder.

The loss function of the AE network in the SFN is the classification accuracy of the HAR network, rather than the mean square error (MSE) between a sample and its corresponding output from the AE network. Therefore, there is a large difference between a sample and its corresponding sample generated by the AE network in the SFN in terms of their pixelwise representations (as shown in Fig. 4 (c)). Moreover, the process of the AE network in the SFN is also different for different action sequences (as shown in Fig. 5). As a result, the samples generated by the AE network in the SFN are more diverse than those generated via most existing data augmentation methods. It is worth noting that although the AE network has only one output, the parameters of the transformation network are updated during training. Therefore, for a given sample, the output of the AE network will be different in each iteration.

C. Fusion Weight

Eq. (2) shows that the fusion weight is an important parameter that directly affects the fusion results. Therefore, the design of the parameter setting strategy is a key problem for an SFN.

First, to enable the SFN to be used for more flexible testing, the trained HAR network needs to be able to separately classify an original sample ($\beta = 0$ in Eq. (3)) and its corresponding output from the transformation network ($\beta = 1$ in Eq. (3)) separately. Therefore, the fusion weight cannot have a fixed value during training.

Second, at the beginning of training, to improve the convergence of the SFN and the HAR network, more fused samples, which are generated by fusing each sample and its corresponding output from the transformation network, are needed to train the HAR network. However, at the end of training, to ensure the classification performance of the HAR network for the original and generated samples, more original samples and samples generated by the transformation network are needed to train the SFN and HAR network separately.

Based on the above analysis and related studies [46], the fusion weight is defined as follows:

$$\begin{aligned} \lambda \in & Beta(\alpha, \alpha) \\ \alpha = & max(1 - n/N, 0.1) \end{aligned} \tag{4}$$

where α is the parameter of the beta distribution, n is the number of the current epoch and N is the total number of epochs.

Eq. (4) shows that at the beginning of training, the fusion weight approximately follows a uniform distribution (α =1.0 as shown by the red line in Fig. 6), which means that more original samples and samples generated by the transformation network are fused. At the end of training, the fusion weight should approximately follows the Bernoulli distribution (α =0.1 as shown by the green dotted in Fig. 6). In this case, more original samples and samples generated by the transformation network are utilized to train the HAR network separately. In this way, the HAR network can classify each sample and the corresponding sample generated by the SFN separately.

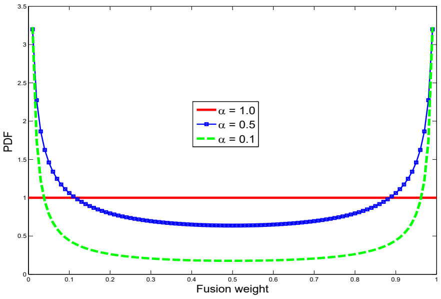
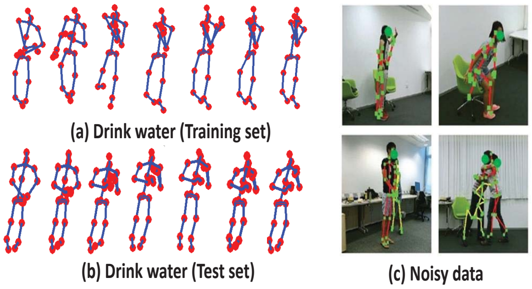


Fig. 6: The probability density function (PDF) of fusion weight on different stages of training.


(a) Drink water (Training set)

(b) Drink water (Test set)

(c) Noisy data

Fig. 7: Skeleton snapshots from the NTU RGB+D dataset [38].

D. Loss Function

For an SFN cascaded with an HAR network, the loss function is defined as follows:

$$\mathcal{L} = \lambda \cdot \mathcal{L}(y_i, \hat{y}) + (1 - \lambda) \cdot \mathcal{L}(y_j, \hat{y}) \tag{5}$$

where

$$\begin{aligned} \hat{y} &= D(\tilde{x}_{ij}) \\ \mathcal{L}(y_i, \hat{y}) &= -\sum_{c=1}^{C} y_i(c) \cdot log(\hat{y}(c)) \\ \mathcal{L}(y_j, \hat{y}) &= -\sum_{c=1}^{C} y_j(c) \cdot log(\hat{y}(c)) \end{aligned}$$

Eq. (5) shows that the loss function includes two terms. The first term, $\mathcal{L}(y_j, \hat{y})$, can be regarded as the classification error of the HAR network for the original sample. This term ensures that the HAR network can classify the original samples and that the SFN can be used during testing. The second term, $\mathcal{L}(y_i, \hat{y})$, can be regarded as the classification error of the HAR network for the samples generated by the SFN. This term not only ensures that the HAR network can classify the samples generated by the SFN but also ensures that each generated sample is the same as the corresponding original sample in terms of their classification by the HAR network.

IV. Optimization

In this section, to further enhance the diversity of generated samples by SFN, we extend the number of fused samples from a pair to multiple samples. Then, a multiscale transformation network (MSTN) is proposed to enhance the capability of the transformation network. Finally, the loss function is optimized to adapt to these improvements.

A. Multiple Samples Fusion

To further enhance the sample diversity, the number of samples used in fusion is extended from a pair to multiple samples. The multisample fusion process is formulated as follows:

$$\begin{aligned} \tilde{x}_{i_{1,..,m},j_{1,..,n}} &= \lambda \cdot f_\theta(\tilde{x}_{i_{1,..,m}}) + (1 - \lambda) \cdot \tilde{x}_{j_{1,..,n}} \\ \tilde{y}_{i_{1,..,m},j_{1,..,n}} &= \lambda \cdot \tilde{y}_{i_{1,..,m}} + (1 - \lambda) \cdot \tilde{y}_{j_{1,..,n}} \end{aligned} \tag{6}$$

where the new sample $(\tilde{x}_{i_{1,..,m},j_{1,..,n}}, \tilde{y}_{i_{1,..,m},j_{1,..,n}})$ is generated from the given sample pair $((\tilde{x}_{i_{1,..,m}}, \tilde{y}_{i_{1,..,m}}), (\tilde{x}_{j_{1,..,n}}, \tilde{y}_{j_{1,..,n}}))$, and the sample $\tilde{x}_{i_{1,..,m}}$ and its corresponding label $\tilde{y}_{i_{1,..,m}}$ are generated as follows, and similar expressions hold for $\tilde{x}_{j_{1,..,n}}$ and $\tilde{y}_{j_{1,..,n}}$:

$$\begin{aligned} \tilde{x}_{i_{1,..,m}} &= \frac{1}{\sum_{k=1}^{m} \lambda_k} \cdot \sum_{k=1}^{m} \lambda_k \cdot x_{i_k} \\ \tilde{y}_{i_{1,..,m}} &= \frac{1}{\sum_{k=1}^{m} \lambda_k} \cdot \sum_{k=1}^{m} \lambda_k \cdot y_{i_k} \end{aligned}$$

Eq. (6) shows that regardless of how many samples are used in fusion, the transformation network is used only once to reduce the computational complexity.

As the number of samples used increases, the diversity of the generated samples increases. However, the assumption that linear interpolations of feature vectors should lead to linear interpolations of the associated targets does not always hold, such as when ”standing up” and ”sitting down” are fused. Therefore, determining the number of samples to be used in fusion requires balancing the tradeoff between the diversity of the generated samples and the accuracy of the corresponding labels.

B. Multiscale Transformation Network

To further enhance the ability of the transformation network to remove useless information and enhance useful information, a multiscale transformation network (MSTN) is designed.

During training, each newly generated sample $\tilde{x}_{i_{1..m}}$ can be rewritten as follows:

$$\tilde{x}_{i_{1..m}} = \frac{1}{\sum\limits_{s=1}^{S} w_s} \cdot \sum_{s=1}^{S} w_s \cdot f_\theta^s(x_{i_{1..m}}) \tag{7}$$

where f_θ^s is a transformation network with scale s. To improve the generalization ability of the MSTN, w_s is defined in the same way as the fusion weight in Eq. (4).

During testing, to improve the performance of the HAR network, the transformation network at each scale outputs a generated sample that is then input into the HAR network to obtain a prediction result. Finally, all the prediction results are fused as follows:

$$\hat{y} = \frac{\beta}{\sqrt{S}} \cdot \sum_{s=1}^{S} D(f_\theta^s(x)) + (1 - \beta) \cdot D(x) \tag{8}$$

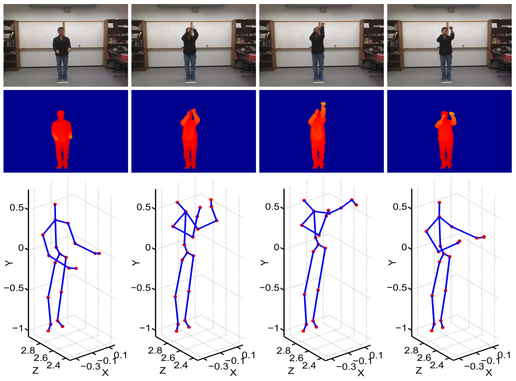


Fig. 8: Skeleton snapshots from the UTD-MHAD dataset [49].

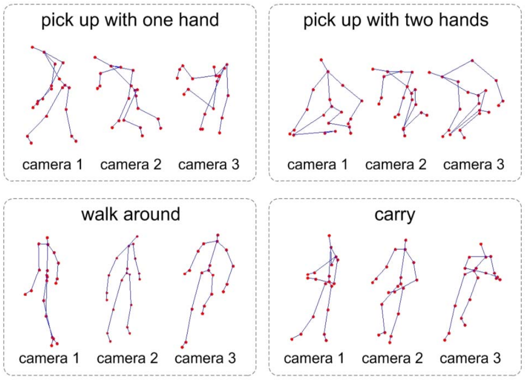


Fig. 9: Skeleton snapshots from the Northwestern-UCLA dataset [50].

As the number of scales increases, it becomes more difficult for the MSTN to converge, especially when the following HAR network is deep. Additionally, the computational complexity during testing increases. Therefore, selecting the number of scales requires balancing the tradeoff between time complexity and accuracy.

C. Loss Function Optimization

First, as the number of samples used in fusion increases, the generated samples become more diverse, but their corresponding labels become more inaccurate. Second, as the number of scales increases, more parameters must be learned in the MSTN. Therefore, it is more difficult for the MSTN to converge. Third, at the beginning of training, the HAR network is far from convergence, so a loss function based on the classification error cannot effectively guide it toward convergence.

Based on the above considerations, to reduce the effects of the number of fused samples and the number of scales used in the SFN, the loss function can be rewritten as follows:

$$\begin{aligned} \mathcal{L} &= \lambda \cdot (\mathcal{L}(y_i, \hat{y}) + \grave{\gamma} \left\|\hat{x} - x\right\|_2) + (1-\lambda) \cdot \mathcal{L}(y_j, \hat{y}) \\ \grave{\gamma} &= \gamma \cdot max(1 - n/N, 0.1) \end{aligned} \tag{9}$$

where n and N are the number of the current epoch and the total number of epochs.

Eq. (9) shows that the reconstruction error is introduced as a regularization term. At the beginning of training, since the HAR network is far from convergence, the classification error is inaccurate. In this case, a greater reconstruction error with respect to the original sample is considered to improve the convergence of the HAR network. At the end of training, since the HAR network is near convergence, more classification error is considered. This process ensures that each sample and its corresponding output from the MSTN will be complementary to each other; then, the SFN can be used during testing to improve the accuracy of the HAR network.

V. Experiments and Analysis

A. Datasets and Settings

Datasets. We evaluate the performance of our method on three benchmark skeleton datasets: NTU RGB+D [38], UTD-MHAD [49] and Northwestern-UCLA []. We first report a detailed ablation study conducted on the NTU RGB+D dataset to examine the contributions of the various components of the proposed model to its performance. Then, we compare our method with other state-of-the-art methods on all datasets.

1) **NTU RGB+D dataset** [38] (hereafter, called NTU). The NTU dataset contains 60 actions performed by 40 subjects from various viewpoints, resulting in 56,880 skeleton sequences. This dataset also contains noisy skeleton joints, which are especially challenging for recognition. Following the cross-view protocol (NTU-CV), we used all samples from camera 1 for testing and the samples from cameras 2 and 3 for training. The training and testing sets contained 37,920 and 18,960 samples, respectively. Following the cross-subject protocol (NTU-CS), we split the 40 subjects into training and testing groups. Each group contained samples of actions performed by 20 of the subjects, captured from all three different views. For this evaluation, the training and testing sets contained 40,320 and 16,560 samples, respectively.
2) **UTD-MHAD dataset** [49] (hereafter, called UTD): The UTD dataset was collected using a Microsoft Kinect

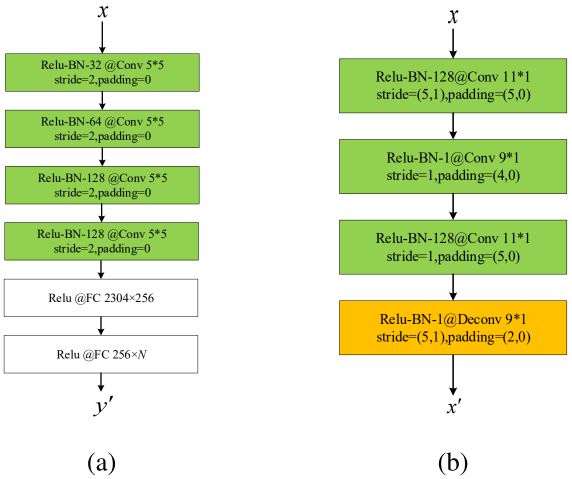


Fig. 10: Structures of CNN-based AE network and **CNN** model. (a) the **CNN** model, (b) the CNN-based AE network.

TABLE II: The performance of the SFN variants for the baseline model with different fusion weight strategies

P	NTU-CV				NTU-CS			
	Mixup	SFN_0	SFN_1	$\text{SFN}_{0.5}$	Mixup	SFN_0	SFN_1	$\text{SFN}_{0.5}$
0	79.53	80.28	81.23	82.23	70.02	70.50	71.15	72.15
1	84.84	83.36	82.76	85.64	76.24	75.48	74.3	77.97
2	85.8	86.27	87.16	88.79	77.70	77.86	78.71	80.50
3	85.74	86.50	87.25	88.98	77.67	77.92	78.48	80.37

sensor and a wearable inertial sensor in an indoor environment. The dataset contains 27 actions performed by 8 subjects. Each subject repeated each action 4 times, resulting in a total of 861 sequences. We used this dataset to compare the performances of methods using different data modalities. The cross-subject protocol was used for evaluation.

3) **Northwestern-UCLA dataset** [50] (hereafter, called NUCLA). The NUCLA dataset contains 1,494 sequences covering 10 action categories: picking up with one hand, picking up with two hands, dropping trash, walking around, sitting down, standing up, donning, doffing, throwing and carrying. Each action was performed one to six times by ten subjects. This dataset contains data captured from a variety of viewpoints. Following [50], we used the samples from the first two cameras as the training data and the samples from the third camera as the testing data.

Training. For training, the batch sizes for the NTU, UTD and NUCLA datasets were set to 256, 8 and 8, respectively, and the total number of iterations for the NTU, UTD and NUCLA datasets are set to 1000, 4000 and 2000. For optimization, we used Adam with the default settings in PyTorch. The learning rate was set to 10^{-3} for the first 80% iterations and 10^{-4} for the remaining 20% iterations. The implementation was based on PyTorch and was run on a system with a 8 GTX1080Ti card and 256 GB of RAM. To reduce the effects of random parameter initialization and random sampling, we repeated the training on the UTD and NUCLA datasets five times and report the average results.

B. Methods

Regularization methods.To fully and exactly evaluate the proposed method, we compare the method with 5 state-of-the-art regularization methods for the RNN model, including 1 L2 regularization method, 2 network agriculture regularization methods and 2 data augmentation methods. The compared methods are as follows:

- **L2 regularization**. A regulation method for network parameters based on L2 weight decay. The weight decay in the L2 penalty was set to 10^{-3} in the experiments.
- **Dropout** [24] 2.A regularization method for networks in which units (along with their connections) are randomly dropped from the neural network during training. The dropout probability was set to 0.5 in all experiments.
- **Zoneout** [25]. A method for regularizing RNN networks by randomly preserving hidden activations.
- **Rotation** [35]. A data augmentation method for skeleton data based on random rotation in 3D coordinates.
- **Mixup** [46]. A data augmentation method based on fusing pairs of samples and their labels.

Compared methods. To fully and exactly evaluate our proposed method, we compare our method with 20 state-of-the-art methods

- **10 LSTM-based methods**. Such as Deep RNN [38], ST-LSTM + Trust Gates [51], Geometric Features + RNN [52] ,GCA-LSTM [23], STA-LSTM [40], Pose-conditioned STA-LSTM [53], VA-LSTM [54], EnTS-LSTM [55], Zoneout [25] and IndRNN [26],
- **8 CNN-based methods**. Such as C3DJ [56], JTM [22], JDM [41], Res-TCN [57], Clips-CNN + MTLN [33], Optical Spectra + CNN [10], EVCNN [21]) and ST-GCN [42],
- **2 methods based on other types of networks**. 3DHOT-MBC [7], LieNet-3Blocks [58].

HAR models. To evaluate our proposed method, the SFN was cascaded together with three LSTM-based HAR networks separately for training and testing, including the baseline model, a CNN model, a Bi-LSTM model and a two-stream (TS) model, as described below:

- **Baseline**. We built the baseline model by stacking 3 LSTM layers followed by 1 FC layer; this design is similar to many typical HAR network designs [16], [40]. The numbers of neurons in each of the three LSTM layers was set to 100. The number of neurons in the FC layer was set equal to the number of action classes, and the exponential linear unit (ELU) function was used as the activation function.

TABLE III: Performance of SFN for baseline model with different fusion weight strategies

Strategy	NTU-CV			NTU-CS		
	SFN_0	SFN_1	$\text{SFN}_{0.5}$	SFN_0	SFN_1	$\text{SFN}_{0.5}$
$\beta = 0$	79.53	–	–	70.02	–	–
$\beta = 1$	–	81.90	–	–	74.76	–
$\beta = 0.5$	–	–	84.33	–	–	77.72
$\beta \in U(0,1)$	85.71	82.62	87.11	78.46	75.01	79.86
$\beta \in B(2,0.5)$	85.61	86.64	87.39	76.55	77.91	78.77
$\beta \in Beta(0.2,0.2)$	86.27	87.16	88.79	77.86	78.71	80.50
$\beta \in$ Eq (4)	87.79	88.23	90.19	80.59	80.35	82.90

- **CNN**. We built the CNN model by stacking 4 CNN layers followed by 2 FC layers (as shown in Fig. 10 (a)). To adapt to the CNN network structure, the skeleton sequences were converted into images using the method in [13].
- **Bi-LSTM**. To construct this model, the LSTM layers in the baseline model were replaced with bidirectional LSTM layers, and the number of neurons in each of the three bidirectional LSTM layers was set to 200.
- **TS**. The design of this model is similar to that of a two-stream convolutional network, except that two baseline models are used in place of the CNN models. The inputs to one baseline model are the original skeleton data, and the inputs to the other baseline model are the frame differences of the skeleton data.

AE network structures. To evaluate our proposed method under different network structures, in addition to the LSTM-based AE network structure, we also built a transformation network with a CNN-based AE network structure (as shown in Fig. 10 (b)).

Ablation Studies. To better understand the contributions of the different components of our method, we also implemented 3 SFN variants to perform extensive ablation studies (similar to a multiscale SFN (MSSFN)):

- **SFN**$_0$. In this case, β is set to 0 in Eq. (8), which means that only the original samples are used during testing, with no contribution from the transformation network. Thus, the SFN becomes a traditional data augmentation method.
- **SFN**$_1$. In this case, β is set to 1 in Eq. (8), which means that only the samples generated by the SFN are used during testing. Thus, the SFN becomes a preprocessing method.
- **SFN**$_{0.5}$. In this case, β is set to 0.5 in Eq. (8), which means that both the original samples and the corresponding outputs from the transformation network are used and fused at the decision level during testing.

C. Ablation Study

We examine the effectiveness of the proposed components in the SFN in this section via action recognition experiments on the NTU RGB+D dataset. For simplicity, we discuss only cases involving the baseline model in this section. If not otherwise specified, the compression ratio r and weight γ are set to 0.5 and 0.1 respectively.

TABLE IV: The accuracy of the SFN variants for the baseline model under different compression ratios r

r	NTU-CV			NTU-CS		
	SFN$_0$	SFN$_1$	SFN$_{0.5}$	SFN$_0$	SFN$_1$	SFN$_{0.5}$
0.1	87.66	87.04	89.28	80.37	78.89	81.73
0.3	87.66	88.06	89.88	80.01	80.48	82.77
0.5	87.79	88.23	90.19	80.59	80.35	82.90
0.7	87.13	87.51	89.45	79.35	79.61	82.2
0.9	87.00	86.64	89.15	79.21	79.83	82.23

TABLE V: Accuracy of MSSFN under different scale combinations

Scales	NTU-CS			
	0.3	0.5	0.7	fusion
Single Scale	82.77	82.90	82.2	–
0.3, 0.5	82.58	82.42	–	83.09
0.3, 0.7	82.36	–	82.53	83.29
0.5, 0.7	–	82.85	82.68	83.52
0.3, 0.5, 0.7	82.56	82.82	82.91	83.31

1) MultiSample Fusion: In this section, we evaluate the effect of the number of samples used in fusion. For simplicity and fairness, m and n are set to the same values in Eq. (6) ($P = m = n$), and $P = 0$ means that the two samples in the pair are the same, namely $x = x_i = x_j$. Second, the fusion weight is set to $\lambda \in Beta(0.2, 0.2)$ in accordance with the Mixup method [46]. The results are summarized in Table. II.

Table. II shows that SFN$_1$ is usually better than SFN$_0$ in terms of accuracy (except for $P = 1$) because the AE network in the SFN can remove useless information and increase the amount of useful information (as shown in Fig. 5). For either an original sample (SFN$_0$) or the corresponding output from the SFN (SFN$_1$), the HAR network can classify it effectively. Therefore, we can flexibly use the transformation network during testing. SFN$_{0.5}$ is better than the other methods in all cases, demonstrating that an original sample and the corresponding output of the SFN are strongly complementary to each other.

As the number of fused samples increases, the performances of the mixup and SFN methods substantially increase for $P = 3$. For example, on the NTU-CS dataset, the accuracy of SFN increases by 2.53% when P increases from 1 to 2 but decreases slighty by 0.13% when P increases from 2 to 3. This difference can probably be explained by the fact that the greater the number of samples used in fusion is, the more diverse the fusion samples that are generated but the more inaccurate the corresponding labels. When $P < 3$, the improvement gained from the increased sample is greater than the degradation due to the inaccuracy of the corresponding labels. However, when $P = 3$, the corresponding label become sufficiently inaccurate to overcome the improvement resulting from the higher diversity of the generated samples.

TABLE VI: Accuracy of the SFN under different γ on the NTU dataset

γ	NTU-CV			NTU-CS		
	SFN$_0$	SFN$_1$	SFN$_{0.5}$	SFN$_0$	SFN$_1$	SFN$_{0.5}$
0.01	87.55	87.11	89.4	79.82	79.83	82.33
0.1	87.79	88.23	90.19	80.59	80.35	82.90
1	87.64	87.9	89.86	80.12	80.34	82.72
10	87.76	88.36	89.76	79.46	80.4	82.08
100	87.37	88.1	88.5	79.1	80.85	80.91

TABLE VII: Accuracy of the SFN under different AE networks and HAR networks on the NTU dataset

Network		NTU-CV					NTU-CS				
HAR	AE	None	Mixup	SFN$_0$	SFN$_1$	SFN$_{0.5}$	None	Mixup	SFN$_0$	SFN$_1$	SFN$_{0.5}$
CNN	CNN	80.60	82.35	82.81	83.13	83.45	69.33	65.77	70.09	70.18	71.89
	LSTM			78.23	80.59	85.93			65.29	66.96	74.84
LSTM	CNN	79.53	85.8	88.05	88.14	88.73	70.02	77.70	76.53	76.76	79.20
	LSTM			87.79	88.23	90.19			80.59	80.35	82.90

Based on these observations, to achieve an appropriate tradeoff between complexity and performance, P was set to 2 for all the sample fusion methods in the following experiments.

2) Fusion Weight: In this work, we evaluate the performance of the SFN variants with different weight setting strategies. The results are summarized in Table. III.

Table. III shows that the random strategies are better than the fixed strategies, probably because the random strategies can generate more diverse fusion samples and thus improve the generalization ability of the HAR network. The fusion weight strategy based on a beta distribution is better than those based on other random distributions because when a sample is fused with more information about the other sample ($\beta \in U(0,1)$), more diverse fusion samples are generated but the corresponding labels are more inaccurate, and vice versa ($\beta \in B(2,0.5)$). The beta-distribution-based strategy can achieve an efficient tradeoff between sample diversity and label accuracy. However, our strategy, as defined in Eq. (4), is better than the other strategies. For example, for NTU-CS, the accuracy of SFN$_{0.5}$ increases from 80.50% with the beta-distribution-based strategy to 82.90% with our strategy. Therefore, Eq. (4) is found to be the most suitable strategy for our method, confirming that the previous analysis is correct.

Based on these observations, we used Eq. (4) as the fusion weight strategy for the SFN in the following experiments.

3) Compression Ratio of AE: In this section, we evaluate the performance of the SFN variants under different AE network compression ratios. The results are shown in Table. IV.

Table. IV shows that as the compression ratio r increases, the accuracy of the SFN initially increases and then decreases. For example, the SFN achieves the best accuracy at $r = 0.5$ for the NTU-CV dataset in most cases. These results are likely explained by the fact that as the compression ratio decreases, the AE network can remove more useless information from the samples, but some discriminative information is also inevitably lost. As a result, the AE network will be underfit in this case. By contrast, as the compression ratio increases, the AE network can retain more discriminative information about the samples, but more useless information is also retained. As a result, the AE network will be overfit in this case.

TABLE VIII: Performance of regularization methods under different skeleton datasets

Method	NTU-CV	NTU-CS	NUCLA	UTD
None	79.53	70.02	61.13	63.16
L2	78.79	69.57	56.91	63.5
Dropout	79.6	70.81	62.17	68.23
Zoneout	85.78	78.24	61.02	67.67
Rotation	82.36	72.23	75.65	78.6
Mixup	85.8	77.7	72.04	82.51
SFN$_0$	87.79	80.59	78.21	84.95
MSSFN$_0$	88.21	80.31	81.30	87.44
SFN$_{0.5}$	90.19	82.90	79.57	87.75
MSSFN$_{0.5}$	90.75	83.52	82.61	88.67

To analyze the effect of multiple scales, we evaluate the effect of MSSFNs with different scale combinations. The results are summarized in Table. V. As the number of scales increases, the MSSFN becomes better than SFN. For example, on the NTU-CS dataset, the performance increases from 82.77% for a single-scale SFN to 83.52% for the MSSFN with r = {0.5, 0.7}. The 3-scale MSSFN performs slightly worse than the 2-scale MSSFNs do. This performance difference can be explained by the fact that, on the one hand, as the number of scales increases, the convergence of the MSSFN worsens, but on the other hand, the representation ability of the baseline model is more limited. We believe that the performance can be further improved by increasing the number of epochs or the representation ability of the HAR network.

Based on these observations, r was set to 0.5 for SFN and to 0.5 and 0.7 for the MSSFN in the following experiments.

4) Reconstruction Error Constraints: To analyze the effect of the reconstruction error constraints, we evaluate the performance of the SFN variants with different weights. The results are shown in Table. VI.

Table. VI shows that in most cases, the SFN achieves

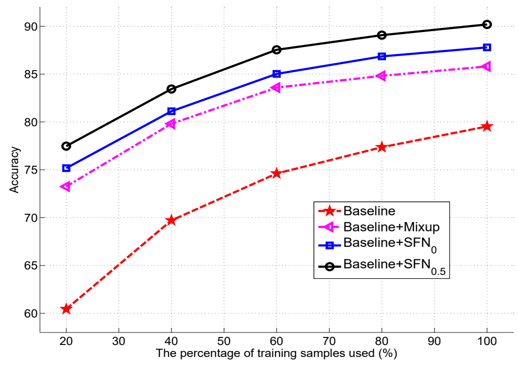


Fig. 11: Performance of the SFN for NTU-CV dataset under different percentages of training samples used.

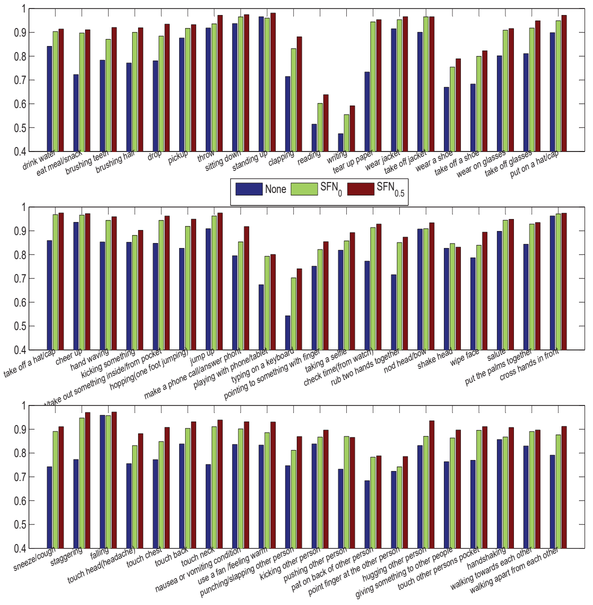


Fig. 12: Accuracy for each action in the NTU-CV dataset with the baseline HAR model.

the best performance with $\gamma = 0.1$ probably because a small improves the convergence of the SFN by means of the reconstruction error constraints, especially when the labels generated through sample fusion are inaccurate. However, as γ increases, the output of the transformation network in the SFN becomes more similar to the input sample at the pixel level rather than at the classification level. In this case, the output not only cannot guide the HAR network to learn more useful information during training, but also cannot ensure complementarity during testing.

Based on these observations, γ was set to 0.1 in the following experiments.

5) Different Numbers of Training Samples: To evaluate the performance of the SFN variants when different numbers of training samples are used, the percentage of training samples used was varied from 20% to 100% in increments of 20%. The results are shown in Fig. 11.

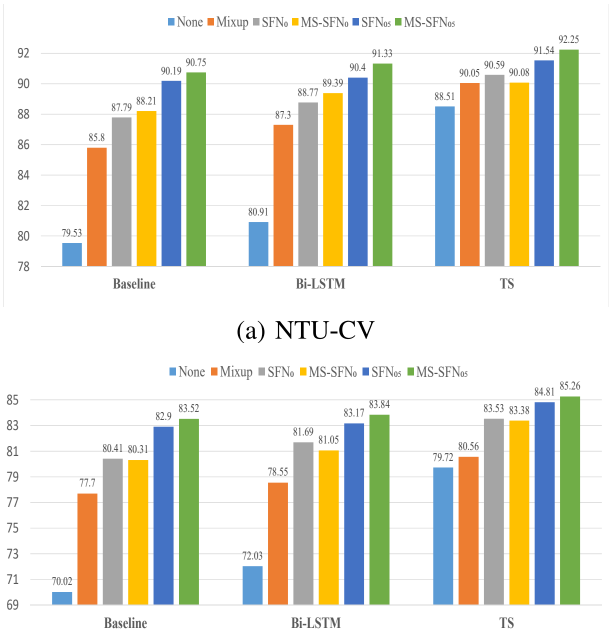
Fig. 11 shows that the SFN effectively improves the performance of the baseline model when different numbers of training samples are used, especially when the number of training samples used is small. For example, when only 20% of the training samples were used, the baseline model achieved an accuracy of only 60.44%, whereas the baseline+SFN$_{0.5}$ can achieve an accuracy of 77.59%, an increase of 17.15%. As the number of training samples used increases, the performance improvement achieved through data augmentation gradually decreases. For example, the baseline+SFN$_{0.5}$ model achieved an accuracy of only 90.75%, an increase of only 11.22%, when all training samples were used. This finding can be explained by the fact that as the number of training samples used increases, the generalization ability of the baseline model also increases, which decreases the improvement space available through data augmentation.

TABLE IX: Results of all skeleton-based methods on the NTU RGB+D dataset

Method	NTU-CS	NTU-CV
Lie Group [58]	61.37	66.95
Deep RNN [38]	59.29	64.09
ST-LSTM + Trust Gates [51]	69.20	77.70
Geometric Feature + RNN [52]	70.26	82.39
GCA-LSTM [23]	74.40	82.80
STA-LSTM [40]	73.4	81.2
PC-STA-LSTM [53]	77.10	84.50
VA-LSTM [54]	79.40	87.60
Zoneout [25]	78.24	85.78
EnTS-LSTM [55]	74.60	81.25
IndRNN (6 layers) [26]	81.80	87.97
JTM + CNN [22]	73.40	75.20
JDM + CNN [41]	76.20	82.30
Res-TCN [57]	74.30	83.10
SkeletonNet [59]	75.94	81.16
Clips-CNN + MTLN [33]	79.57	84.83
EVCNN [21]	80.03	87.21
ST-GCN [42]	81.5	88.3
Baseline	70.02	79.53
Baseline+SFN$_{0\|0.5}$	80.59|82.90	87.79|90.19
Baseline+MSSFN$_{0\|0.5}$	80.31|83.52	88.23|90.75
TS	79.72	88.51
TS+SFN$_{0\|0.5}$	83.53|84.81	90.59|91.54
TS+MSSFN$_{0\|0.5}$	83.38|85.26	90.08|92.25

6) Comparisons with Different Network Structures: To analyze the effect of the network structures, we evaluate the performance of the SFN variants with different combinations of CNN and LSTM network structures in the AE and HAR network models. The results are shown in Table. VII.

Table. VII shows that the LSTM structure is a more powerful network structure than the CNN structure for analyzing skeleton data. For example, on the NTU-CS dataset, the purely CNN-based network structure achieved an accuracy of only 71.89%, whereas the purely LSTM-based network structure achieved an accuracy of 82.9%, an increase of 11.01%. This finding can probably be explained by the fact that the key property of the convolution operator is the ability to leverage the spatially local correlations found in natural images, whereas there usually are no spatially local correlations in images converted from skeleton data. For the CNN-based HAR network, SFN$_1$ is superior to SFN$_0$ in term of accuracy, indicating that the AE network can learn a better transformation for converting a skeleton sequence into an image that is more suitable for the CNN structure. Note, however, that no matter how poorly SFN$_0$ and SFN$_1$, SFN$_{0.5}$ is still better than all other methods. This finding demonstrates that SFN$_0$ and SFN$_1$ are strongly complementary to each other. The CNN-based network structure is obviously worse than the LSTM-based network structure in terms of accuracy on the NTU-CS dataset. This finding can probably be explained by the fact that



(a) NTU-CV

(b) NTU-CS

Fig. 13: Performance of the SFN under different HAR networks.

the temporal information is more important than the spatial information when the cross-subject protocol is applied.

Based on these observations, LSTM-based network structures were used in the following experiments.

D. Comparisons with State-of-the-Arts Regularization Methods

In this section, we compare our algorithm with other regularization methods in terms of performance. For simplicity, we discuss only cases involving the baseline model. The results are shown in Table. VIII.

Table. VIII shows that compared to the other regularization methods, data augmentation more effectively improves the performance of the baseline model on different datasets. For example, on the NUCLA dataset, the accuracy of the baseline model is improved by at least 10% via data augmentation, whereas the accuracy is only slightly improved or even degraded with the other regulation methods. The performance of data augmentation based on single-sample transformation depends on the dataset. For example, the rotation method performs better than mixup on the NUCLA dataset but worse than mixup on the other datasets. The performance of sample fusion is clearly better than that of single-sample transformation because sample fusion generates samples with greater diversity. Our method outperforms the other regularization methods on different datasets.

Fig. 12 shows the accuracy for each action in the NTU-CV dataset. The SFN effectively improves the accuracy for most of the actions. For example, the accuracy increases to 19.72% for the action ”typing on a keyboard”. Note that the accuracy is clearly worse for ”reading” and ”writing” than for the other actions because these two actions involve similar temporal relationships, so an LSTM network using only temporal information cannot distinguish them.

TABLE X: Results of all skeleton-based methods on the UTD-MHAD dataset

Method	Accuracy%
Cov3DJ [56]	85.58
Deep RNN [38]	66.10
JTM + CNN [22]	85.81
Optical Spectra + CNN [10]	86.97
3DHOT-MBC [7]	84.40
JDM +CNN [41]	88.10
Baseline	63.16
Baseline+SFN$_{0\vert 0.5}$	84.95 $\vert$ 87.75
Baseline+MSSFN$_{0\vert 0.5}$	87.44 $\vert$ 88.67
TS	70.51
TS+SFN$_{0\vert 0.5}$	91.16 $\vert$ 91.89
TS+MSSFN$_{0\vert 0.5}$	92.13 $\vert$ 92.33

E. HAR Network

We evaluate the performance of the SFN variants when combined with three different HAR networks to further validate our method. The results are shown in Fig. 13.

Fig. 13 shows that both the SFN and MSSFN methods outperform mixup in terms of accuracy, even when the transformation network is not used during testing (SFN$_0$ and MSSFN$_0$). For example, on NTU-CS, the baseline model achieved an accuracy of only 70.02%, whereas baseline+MSSFN$_{0.5}$ achieved an accuracy of 83.52%, an increase of 13.5%. Even when the transformation network was not used during testing, the accuracy of the baseline+MSSFN$_{0.5}$ model was 80.31%, an increase of 10.29%. As the complexity of the HAR network increases, the performance with an SFN or MSSFN also increases. Therefore, our method can effectively improve the generalization ability of an HAR network.

Note that MSSFN$_0$ performs slightly worse than SFN$_0$ in terms of accuracy in some cases because an MSSFN has more parameters to be learned. Therefore, it is more difficult for an MSSFN to converge. We believe that this problem can be alleviated by increasing the number of training epochs.

F. Comparisons with State-of-the-arts methods

In this section, we compare our method with other state-of-the-art methods on the NTU RGB+D, UTD-MHAD and Northwestern-UCLA datasets.

1) NTU RGB+D dataset: Table. IX presents the results of the various methods on the NTU dataset and shows that the proposed method greatly improves the performance of both the baseline model and the TS model. For example, even when the transformation network was not used during testing, the accuracy of the baseline model increased by 10.29% with our method, reaching 80.31%, better than the results of most of

TABLE XI: Results of all skeleton-based methods on the Northwestern-UCLA dataset.

Method	Accuray%
Lie group [58] (reported by [21])	74.20
Actionlet ensemble [60]	76.60
HBRNN (reported by [21])	78.52
Enhanced visualization [21]	86.09
EnTS-LSTM [55]	89.22
Baseline	63.16
Baseline+SFN$_{0\|0.5}$	78.21 | 79.57
Baseline+MSSFN$_{0\|0.5}$	81.30 | 82.61
TS	65.00
TS+SFN$_{0\|0.5}$	86.30 | 87.61
TS+MSSFN$_{0\|0.5}$	86.96 | 88.91

the existing methods. This finding illustrates the effectiveness of our method in improving the generalization ability of an HAR network for large-scale data.

2) UTD dataset: Table. X presents the results of the various methods on the UTD dataset. Again, our method greatly improves the performance of the baseline model and the TS model. For example, the baseline model alone achieved an accuracy of only 63.16%, while baseline+MSSFN$_{0.5}$ achieved an accuracy of 88.67%, an increase of 25.51%. Even when the transformation network was not used during testing, the baseline model achieved an accuracy of 87.44% with our method, an increase of 24.28%.

3) NUCLA dataset: Table. XI presents the results of the various methods on the NUCLA dataset. Again, the proposed method greatly improves the performance of the baseline model and the TS model. For example, the TS model alone achieved an accuracy of only 65.00%, while TS+MSSFN$_{0.5}$ achieved an accuracy of 88.91%, an increase of 23.91%.

Although our method achieves considerable improvement, TS+MSSFN$_{0.5}$ still performs worse than EnTS-LSTM in terms of accuracy, probably because the NUCLA dataset contains fewer classes (10 classes) than the other datasets. Therefore, data augmentation methods based on sample fusion cannot produce more diverse samples.

VI. Conclusion

In this paper, a data augmentation method called an SFN, which can be trained in an end-to-end manner along with an HAR network, is proposed for skeleton-based action recognition. During training, for a given sample pair, one sample is input into a transformation network to obtain a preprocessed sample that is enhanced compared with the original sample. Then, an adaptive fusion strategy is applied to fuse the preprocessed sample and the other sample from the sample pair. Subsequently, the fused sample is sent to the skeleton-based HAR network, and the transformation and HAR networks are trained in an end-to-end manner. During testing, the samples preprocessed by the transformation network and the original samples are input into the HAR network, and the outputs of the HAR network are fused at the decision level. Our method improves the performance of the baseline model by nearly 12% on the NTU RGB+D dataset, which is the largest available dataset for skeleton-based recognition. This result verifies the efficacy of our method compared with other state-of-the-art data augmentation methods and other HAR networks.

In future work, other network architectures, for example, a DenseNet architecture, will be incorporated into SFNs to increase the effectiveness of HAR. We can also expand the SFN approach to other modalities, e.g., RGB and depth, and can enhance its performance by using a deeper AE network.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant U1613209 and Grant 61871154, and in part by Scientific Research Project of Shenzhen City (JCYJ20170306164738129), Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (No.ZDSYS201703031405467), and the project ”The Verification Platform of Multi-tier Coverage Communication Networks for oceans (PCL2018KP002)”, and Shenzhen Science and Technology Projection (JCYJ20160608151239996).

References

[1] L. Chen, H. Wei, and J. M. Ferryman, “A Survey of Human Motion Analysis Using Depth Imagery,” *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1995–2006, 2013.

[2] J. Aggarwal and L. Xia, “Human activity recognition from 3D data: A review,” *Pattern Recognition Letters*, vol. 48, pp. 70–80, oct 2014.

[3] S. Escalera, V. Athitsos, and I. Guyon, “Challenges in Multi-modal Gesture Recognition,” in *Gesture Recognition*. Springer, 2017, pp. 1–60.

[4] L. Lo Presti and M. La Cascia, “3D skeleton-based human action classification: A survey,” *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[5] F. Han, B. Reily, W. Hoff, and H. Zhang, “Space-time representation of people based on 3D skeletal data: A review,” *Computer Vision and Image Understanding*, vol. 158, pp. 85–105, 2017.

[6] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, “RGB-D-based human motion recognition with deep learning: A survey,” *Computer Vision and Image Understanding*, may 2018.

[7] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, “Action Recognition Using 3D Histograms of Texture and A Multi-Class Boosting Classifier,” *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4648–4660, 2017.

[8] J. Weng, C. Weng, and J. Yuan, “Spatio-Temporal Naive-Bayes Nearest-Neighbor (ST-NBNN) for Skeleton-Based Action Recognition,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 445–454.

[9] R. Vemulapalli and R. Chellappa, “Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 4471–4479.

[10] Y. Hou, Z. Li, P. Wang, and W. Li, “Skeleton Optical Spectra-Based Action Recognition Using Convolutional Neural Networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 807–811, mar 2018.

[11] C. Li, Y. Hou, P. Wang, and W. Li, “Joint Distance Maps Based Action Recognition With Convolutional Neural Networks,” *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, may 2017.

[12] J. Liu, A. Shahroudy, D. Xu, A. Kot Chichung, and G. Wang, “Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017.

[13] Y. Du, Y. Fu, and L. Wang, “Skeleton based action recognition with convolutional neural network,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, nov 2015, pp. 579–583.

[14] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A New Representation of Skeleton Sequences for 3D Action Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 4570–4579.

[15] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1243–1252, 2017.

[16] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. Phoenix, Arizona: AAAI Press, 2016, pp. 3697–3703.

[17] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-Based Human Action Recognition With Global Context-Aware Attention LSTM Networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, apr 2018.

[18] B. B. Amor, J. Su, and A. Srivastava, "Action Recognition Using Rate-Invariant Analysis of Skeletal Shape Trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 1–13, 2016.

[19] Yong Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June. IEEE, jun 2015, pp. 1110–1118.

[20] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton Optical Spectra Based Action Recognition Using Convolutional Neural Networks," *Ieee Tcsvt*, vol. PP, no. 99, p. 1, 2016.

[21] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, 2017.

[22] P. Wang, Z. Li, Y. Hou, and W. Li, "Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks," in *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*. New York, New York, USA: ACM Press, 2016, pp. 102–106.

[23] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global Context-Aware Attention LSTM Networks for 3D Action Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 3671–3680.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[25] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations," *arXiv*, 2016.

[26] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN," in *Conference: Computer Vision and Pattern Recognition (CVPR 2018)*, 2018.

[27] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 2684–2691, 2017.

[28] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement: Journal of the International Measurement Confederation*, vol. 89, pp. 171–178, 2016.

[29] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart Augmentation Learning an Optimal Data Augmentation Strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017.

[30] S. Dutta, B. Tripp, and G. Taylor, "Convolutional Neural Networks Regularized by Correlated Noise," *arXiv*, apr 2018.

[31] Z.-H. Feng, J. Kittler, W. Christmas, P. Huber, and X.-J. Wu, "Dynamic Attention-Controlled Cascaded Shape Regression Exploiting Training Data Augmentation and Fuzzy-Set Sample Weighting," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 3681–3690.

[32] H. Wang and L. Wang, "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 3633–3642.

[33] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A New Representation of Skeleton Sequences for 3D Action Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 4570–4579.

[34] W. Yang, T. Lyons, H. Ni, C. Schmid, L. Jin, and J. Chang, "Leveraging the Path Signature for Skeleton-based Human Action Recognition," *arXiv*, jul 2017.

[35] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, jul 2017, pp. 601–604.

[36] T. DeVries and G. W. Taylor, "Dataset Augmentation in Feature Space," 2017. [Online]. Available: http://arxiv.org/abs/1702.05538

[37] P. Wang, Z. Li, Y. Hou, and W. Li, "Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks," in *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*. New York, New York, USA: ACM Press, 2016, pp. 102–106.

[38] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 1010–1019.

[39] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition." Cham: Springer International Publishing, 2016, pp. 816–833.

[40] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 4263–4270.

[41] C. Li, Y. Hou, P. Wang, and W. Li, "Joint Distance Maps Based Action Recognition with Convolutional Neural Networks," *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[42] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," *arXiv*, jan 2018.

[43] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1958–1971, 2013.

[44] D. Wu and L. Shao, "Leveraging Hierarchical Parametric Networks for Skeletal Joints Based Action Segmentation and Recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014, pp. 724–731.

[45] E. P. Ijjina and C. Krishna Mohan, "Classification of human actions using pose-based features and stacked auto encoder," *Pattern Recognition Letters*, vol. 83, pp. 268–277, 2016.

[46] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," *arXiv*, 2017.

[47] Y. Tokozume, Y. Ushiku, and T. Harada, "Between-class Learning for Image Classification," *arXiv preprint arXiv:1711.10284*, 2017.

[48] H. Inoue, "Data Augmentation by Pairing Samples for Images Classification," *arXiv preprint arXiv:1801.02929*, 2018.

[49] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2015-Decem, pp. 168–172, 2015.

[50] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-View Action Modeling, Learning, and Recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014, pp. 2649–2656.

[51] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," *arXiv*, vol. 9907 LNCS, pp. 816–833, 2016.

[52] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer LSTM networks," in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, 2017, pp. 148–157.

[53] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned Spatio-Temporal Attention for Human Action Recognition," *arXiv*, mar 2017.

[54] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition from Skeleton Data," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017, pp. 2136–2145.

[55] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 1012–1020, 2017.

[56] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2466–2472, 2013.

[57] T. S. Kim and A. Reiter, "Interpretable 3D Human Action Analysis with Temporal Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jul 2017, pp. 1623–1631.

[58] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 588–595, 2014.

[59] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "SkeletonNet: Mining Deep Part Features for 3-D Action Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 731–735, 2017.

[60] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3D human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 914–927, 2014.

Fanyang Meng received Ph.D. degree in 2016 under the supervision of Prof.Xia Li from the School of EE&CS, Shenzhen University, China. He currently serves as a Post-doctoral fellow supervised by Prof. Yongsheng Liang and Prof. Hong Liu.

His research interests include computer vision, human action recognition and abnormal detection using RGB, depth and skeleton data.

Hong Liu received the Ph.D. degree in mechanical electronics and automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by National High-level Talents Special Support Plan since 2013. He is also the Director of Open Lab on Human Robot Interaction, PKU.

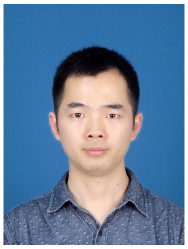
He research fields include computer vision and robotics, image processing, and pattern recognition. Liu has published more than 150 papers and gained Chinese National Aero-space Award, Wu Wenjun Award on Artificial Intelligence, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors in PKU. He is an IEEE member, vice president of Chinese Association for Artificial Intelligent (CAAI), and vice chair of Intelligent Robotics Society of CAAI. He has served as keynote speakers, co-chairs, session chairs, or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC and IIHMSP, recently also serves as reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. On PAMI.

YongSheng Liang received the Ph.D. degree in communication and information system from Harbin Institute of Technology, Harbin, China, in 1999. From 2002 to 2005, he was a Research Assistant in the Harbin Institute of Technology. Since 2017, he has been an Professor in school of electronic and information engineering, Harbin Institute of Technology (Shenzhen).

His research interests include video coding and transfering, machine learning and applications. He has gained the Wu Wenjun Artificial Intelligence Science and Technology Award for Excellence, and the Second Prize in Science and Technology of Guangdong Province.

Juanhui Tu is currently pursuing a master's degree under the supervision of Prof. Hong Liu from the School of EE&CS, Peking University (PKU), China. She is proficient in machine learning and deep learning theoretical knowledge.

Her research interests include human action recognition using skeleton and RGB data. Related methods have been published in ICME, ICASSP and ICIP.



Mengyuan Liu received Ph.D. degree in 2017 under the supervision of Prof. Hong Liu from the School of EE&CS, Peking University (PKU), China. He currently serves as a research fellow supervised by Prof. Junsong Yuan and Prof. Kai-Kuang Ma in the School of Electrical and Electronic Engineering of Nanyang Technological University (NTU), Singapore.

His research interests include human action recognition and abnormal detection using RGB, depth and skeleton data. Related methods have been published in T-CSVT, T-MM, PR, CVPR, AAAI and IJCAI. He has been invited to be a Technical Program Committee (TPC) member for ACM MM 2018 and 2019. He also serves as a reviewer for many international journals and conferences such as T-II, T-IP, T-CSVT, CVIU, ACM MM and WACV.