A Deep Clustering via Automatic Feature Embedded Learning for Human Activity Recognition

Ting Wang, *Student Member, IEEE,* Wing W. Y. Ng*, *Senior Member, IEEE,* Jinde Li,
Qiuxia Wu, *Member, IEEE,* Shuai Zhang, *Member, IEEE,* Chris Nugent, *Senior Member, IEEE,*
Colin Shewell, *Member, IEEE*

***Abstract*—Traditional clustering algorithms are widely used for building bag-of-words (BOW) models to aggregate spatio-temporal feature points extracted from a video for human activity recognition problems. Their performances are restricted by the computational complexity which limits the number of feature points being used. In contrast, deep clustering yields good clustering performance without the limit of the number of feature points. Therefore, this work proposes a dual stacked autoencoders features embedded clustering (DSAFEC) and a BOW construction method based on the DSAFEC (B-DSAFEC) to reduce the computational complexity and to remove the selection restriction. The DSAFEC first transforms feature points extracted from a video to a learned feature space and then probabilities of cluster assignment of feature points are predicted to build BOWs for human activity recognition. A soft clustering is used by assigning each feature point to multiple clusters yielding the largest probabilities instead of only one in hard clustering. Experimental results on three benchmark human activity datasets show that the B-DSAFEC yields better performance compared to five reference methods which are developed based on either traditional clustering methods or deep clustering methods.**

***Index Terms*—Bag-of-words (BOW), deep clustering, human activity recognition, autoencoder**

I. Introduction

HUMAN activity recognition (HAR) is an important research topic in the field of computer vision. The objective of bag-of-words (BOW) model is to aggregate various feature points of a video sequence into a fixed-length representation. The BOW is popular in HAR as it is easy to use, highly computationally efficient, and able to cope with most application contexts [1]. Particularly in HAR tasks, state-of-the-art extraction methods extract a vast amount of feature points from videos even for relatively small datasets, so that efficient feature representations are still widely used [2].

The key aspect of the BOW model in HAR is to apply a traditional clustering algorithm to build a visual vocabulary [3]. The most widely used traditional clustering algorithms in BOW include K-means clustering [4], agglomerative clustering (AGNES) [5], and spectral clustering [6]. However, these algorithms need to calculate pairwise distances or similarities between feature points, which has a high computational cost, and this cost grows rapidly as the number of feature points increases. This limitation makes traditional clustering methods unable to deal with large datasets with millions of feature points. A random undersampling of feature points can be applied to reduce the computational cost [7]–[9]. However, this may cause a loss of important information due to the random removal of feature points.

Deep clustering does not need to compute pairwise distances and/or pairwise similarities [10]. Furthermore, deep clustering has no limit on the number of feature points and can utilize all available feature points to build a BOW model. The deep clustering networks [11] [12] can be trained in an end-to-end and joint learning manner on a unified objective function. Finally, deep clustering learns improved representations that are better suited to feature point clustering [13].

Therefore, this work proposes a novel BOW model (B-DSAFEC) based on a deep clustering algorithm—dual stacked autoencoders features embedded clustering (DSAFEC) for HAR. The DSAFEC first transforms the feature points to a learned feature space to generate new representations, which are then subsequently clustered with probabilities. The B-DSAFEC is used to build BOW vectors for HAR using the probabilistic clustering generated by the DSAFEC. Major contributions of this work are:

1) The DSAFEC is a more computationally efficient clustering algorithm than traditional clustering methods and is therefore not limited by the number of feature points in the dataset.
2) The B-DSAFEC is a joint learning framework that automatically learns feature representations and performs cluster assignment simultaneously. The BOW generated by the B-DSAFEC has a good discriminative power for HAR.
3) The B-DSAFEC is robust to the selection of the BOW size and builds better BOW vectors than traditional

This work was supported by National Natural Science Foundation of China under Grant 61876066, Guangdong Province Science and Technology Plan Project (Collaborative Innovation and Platform Environment Construction) 2019A050510006, and 2020 RD Program in Key Areas of Guangdong Province (2020B010166002).

Ting Wang, Wing W. Y. Ng, and Jinde Li are with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. (e-mails: twang.scut@qq.com, 1833844770@qq.com, and wingng@ieee.org, corresponding author: Wing W. Y. Ng).

Qiuxia Wu is with School of Software Engineering, South China University of Technology, Guangzhou, China. (e-mail: qxwu@scut.edu.cn).

Shuai Zhang, Chris Nugent, and Colin Shewell are with School of Computing, Ulster University, Jordanstown, United Kingdom. (e-mails: s.zhang@ulster.ac.uk, cd.nugent@ulster.ac.uk, and cp.shewell@ulster.ac.uk).

clustering algorithms for HAR. The B-DSAFEC also yields better performances than deep clustering methods.

The rest of this paper is organized as follows. Section II reviews related works. The DSAFEC and B-DSAFEC for HAR are proposed in Section III. Experimental results and discussion will be presented in Section IV. Finally, we conclude this paper in Section V.

II. Related Works

A. Bag-of-Words in HAR

Representing local features using the BOW and its variants is successful and popular in dealing with HAR problems [14]. The scale invariant feature transform (SIFT) [15] and the dense trajectory (DT) [16] are successful feature extractors for video. The DT extracts features based on sampling trajectory and motion boundary descriptor.

Based on the DT [16], an improved dense trajectory (IDT) [17] is proposed and shows improved performance in HAR. Although the DT and the IDT are the most successful feature extraction methods for image processing, high demands in both computational complexity and storage limit their application to activity recognition. The space-time interest points (STIP) [18] has been proposed to detect the local spatio-temporal feature points from video sequences. Although the computational complexity and storage requirements of the STIP are much less than the DT and the IDT, the STIP extracts a large number of feature points and some STIP feature points are redundant (unnecessarily increasing the storage requirement of the STIP).

The multi-task information bottleneck (MTIB) clustering [19] employs the agglomerative information maximization to build the common visual vocabulary for multiple tasks. The traditional BOW models and support vector machine (SVM) are integrated into a recurrent neural network (RNN) [2] to allow feature points aggregation and action classification to be implemented simultaneously in a unified network. A discriminative embedding method based on the image-to-class distance (I2CDDE) [20] is proposed to learn compact and discriminative local feature descriptors. The genetic algorithm has also been applied to find optimal BOW representation by optimizing weights of samples, features, words in a visual vocabulary, and SVM parameters in [21]. To obtain a compact representation of video sequences, a visual word ranking method is proposed to select the significant words of the visual vocabulary and reduce the size of the BOW model [3]. An interest point pruning algorithm [22] is proposed to eliminate the large number of redundant STIP feature points and select the more discriminative visual words for building the BOW. The STIP is also employed to build the BOW in [23] where the BOW size is selected automatically via a minimization of the localized generalization error of a radial basis function neural network (RBFNN) [24]. Using action bank as feature extraction, [25] directly trains a RBFNN yielding high performance and then performs uncertainty reduction for ambiguous classes. A semi-supervised method [26] is proposed to categorize human activity using multiple visual features which discovers a common subspace shared by each type of feature and characterizes more discriminative information of each feature type.

B. Deep clustering algorithms

The deep clustering (DC) is applied to address audio source separation problems by predicting implicit segmentation labels of the target spectrogram from audio in [27]. The deep embedded clustering (DEC) [11] learns embedded feature representations with stacked autoencoders and predicts cluster assignment according to distance metrics simultaneously. A joint unsupervised learning framework (JULE) [10] is proposed to extract features from images with a convolutional neural network (CNN) and perform agglomerative clustering [5] with an RNN. The deep embedded regularized clustering (DEPICT) [12] consists of a multilayer convolutional autoencoder and a softmax regression layer which learns feature transformation and cluster assignment via minimizing reconstruction errors and relative entropies. A deep convolutional autoencoder network (DCAN) and a softmax layer are combined to co-optimize the deep representation features and cluster using an integrated loss function to simultaneously minimize the reconstruction loss and the clustering loss [28]. To learn the visual features of images and videos, deep cluster [29] treats the cluster assignments generated by K-means as pseudo-labels for CNN training. A systematic taxonomy of clustering methods that utilize deep neural networks and an improved clustering method based on taxonomy is proposed in [13]. An end-to-end signal approximation objective is used to implement speaker-independent multi-speaker separation with deep clustering in [30]. A hybrid model [31] combining deep clustering and conventional networks offers improved results on the music separation problem. A unified framework [32] is used to jointly solve clustering and representation learning in an iterative manner. In this framework, a CNN is used to learn representations of images and a K-means is used to update clusters. In a multi-task network [33] that jointly learns classification and clustering, the deep clustering is treated as an auxiliary task to explore the structure of image data and assist to train a better model for the classification task.

Several deep clustering methods use unsupervised or supervised representation learning to achieve human activation recognition, such as autoencoders, clustering modules, and feature fusion [34]–[36]. An unsupervised end-to-end learning network architecture [34] is developed for clustering human activities based on raw sequences of wearable sensor data streams. An anomaly detection of human actions method [35] uses a spatio-temporal graph autoencoder (ST-GCAE) to obtain a latent vector for each action. Then, the latent vector is soft assigned to clusters using a deep clustering layer. A potential set of clusters is obtained from egocentric videos with many actions and events by combining a pre-trained CNN, a center surround model (CSM), and a K-means [36]. A new deep clustering algorithm named soft and regularized deep K-means (SR-K-means) [37] is proposed. This algorithm is a version of deep K-means and theoretically proves that maximizing the L_2 regularized mutual information via an approximate alternating direction method (ADM) is equivalent

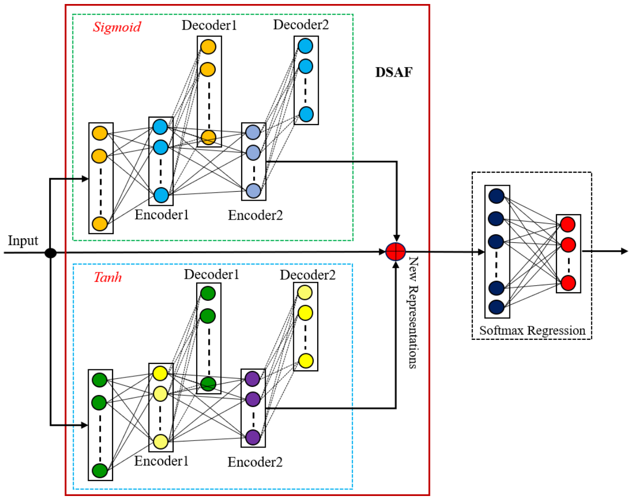


Fig. 1. The framework of the DSAFEC.

to a minimization of the SR-K-means loss. As aforementioned, deep clustering algorithms yield good clustering performances and are not limited by the number of feature points. Therefore, a unique opportunity exists to construct the BOW model for HAR using deep clustering algorithms in this work.

III. B-DSAFEC – BOW Based on The DSAFEC

In this paper, we propose an efficient deep clustering method: dual stacked autoencoders features embedded clustering (DSAFEC), and a novel BOW model based on the DSAFEC (B-DSAFEC). The DSAFEC first uses dual stacked autoencoders features to map the original inputs onto a new feature space to generate new feature representations for clustering. Cluster assignment probabilities are predicted for the new transformed feature representations. The B-DSAFEC builds BOW vectors with the probabilities predicted by the DSAFEC. Details of the DSAFEC and the B-DSAFEC are given in Section III-A and Section III-C, respectively.

A. Dual stacked autoencoders features embedded clustering (DSAFEC)

The DSAFEC uses dual stacked autoencoders features (DSAF) to learn new representations for the original input and then uses a softmax regression to predict cluster assignment probabilities for the representations. The framework of the DSAFEC is shown in Fig. 1. The DSAF concatenates the original inputs and learned features from two stacked autoencoders (SAE) [38] [39] which use the sigmoid (the upper SAE, or sigmoid SAE) and tanh (the lower SAE, or tanh SAE) functions as activation functions, respectively. In this paper, a SAE consists of an input layer and two encoding layers followed by two decoding layers.

A dataset X consists of N feature point $x_i \in R^{d_x}$ in $X = \{x_1, x_2, ..., x_N\}$, where d_x and N denote the length of feature point and the total number of feature points, respectively. In DSAF, the forward propagation of sigmoid SAE with input x_i as follows:

$$a_i^{se1} = \sigma(W_e^{s1} x_i + b_e^{s1}) \tag{1}$$

$$a_i^{se2} = \sigma(W_e^{s2} a_i^{se1} + b_e^{s2}) \tag{2}$$

$$a_i^{sd2} = \sigma(W_d^{s2} a_i^{se2} + b_d^{s2}) \tag{3}$$

$$a_i^{sd1} = \sigma(W_d^{s1} a_i^{sd2} + b_d^{s1}) \tag{4}$$

where σ denotes the sigmoid activation function. a_i^{se1}, W_e^{s1}, and b_e^{s1} denote the output, the weight vector, and biases of the encoding layer while a_i^{sd1}, W_d^{s1}, and b_d^{s1} denote the output, the weight vector, and biases of the decoding layer of the first autoencoder (AE) module of the sigmoid SAE, respectively. a_i^{se2}, W_e^{s2}, and b_e^{s2} denote the output, the weight vector, and biases of the encoding layer while a_i^{sd2}, W_d^{s2}, and b_d^{s2} denote the output, the weight vector, and biases of the decoding layer of the second AE module of the sigmoid SAE, respectively. Similarly, the forward propagation of the tanh SAE with input x_i is as follows:

$$a_i^{te1} = \delta(W_e^{t1} x_i + b_e^{t1}) \tag{5}$$

$$a_i^{te2} = \delta(W_e^{t2} a_i^{te1} + b_e^{t2}) \tag{6}$$

$$a_i^{td2} = \delta(W_d^{t2} a_i^{te2} + b_d^{t2}) \tag{7}$$

$$a_i^{td1} = \delta(W_d^{t1} a_i^{td2} + b_d^{t1}) \tag{8}$$

where δ denotes the tanh activation function. a_i^{te1}, W_e^{t1}, and b_e^{t1} denote the output, the weight vector, and biases of the encoding layer while a_i^{td1}, W_d^{t1}, and b_d^{t1} denote the output, the weight vector, and biases of the decoding layer of the first AE module of the tanh SAE, respectively. a_i^{te2}, W_e^{t2}, and b_e^{t2} denote the output, the weight vector, and biases of the encoding layer while a_i^{td2}, W_d^{t2}, and b_d^{t2} denote the output, the weight vector, and biases of the decoding layer of the second AE module of the tanh SAE, respectively. Then, representations learned in encoding layers of the second sigmoid and the second tanh AEs (i.e. a_i^{se2} and a_i^{te2}) are concatenated with the original input x_i to form the new representation as follows:

$$z_i = a_i^{se2} \oplus x_i \oplus a_i^{te2} \tag{9}$$

where $\oplus$ and z_i denote the concatenating operation and the corresponding feature representation of x_i, respectively.

Given N feature representations being mapped by the DSAF, $Z = \{z_1, z_2, ..., z_N\}$, where $z_i \in R^{d_z}$ and d_z denotes the length of each feature representation. The clustering task is to assign these representations into K categories. The DSAFEC uses a softmax regression function to predict the probabilistic cluster assignment P as follows:

$$p_{ik} = P(y_i = k \mid z_i, \Theta) \quad = \frac{exp(\theta_k^T z_i)}{\sum_{k'=1}^{K} exp(\theta_{k'}^T z_i)} \tag{10}$$

where P is a matrix consisting of all p_{ik} and $\Theta = [\theta_1, \theta_2, ..., \theta_K] \in R^{d_z \times K}$ denotes the softmax function parameters, and p_{ik} denotes the probability of the i^{th} feature representation (z_i) belonging to the k^{th} cluster. The detail of the procedure of network training and prediction task will be presented in Section III-B.

B. Training of the DSAFEC

The DSAFEC is an end-to-end joint learning framework [10]–[12] that learns feature transformation and cluster assignment simultaneously via a minimization of a unified objective function. The unified objective function of the DSAFEC consists of a reconstruction error (L_r) and a clustering loss (L_c) which is defined as follows:

$$L = L_c + \alpha L_r \tag{11}$$

where α denotes the regularization factor between L_c and L_r. The clustering loss L_c focuses on the enhancement of accuracy on cluster assignment probability predicted by the DSAFEC while the reconstruction loss L_r improves learned representation of the DSAFEC for clustering.

In the DSAF, the reconstruction loss is defined as follows:

$$\begin{aligned} L_r =& \frac{1}{2N} \sum_{i=1}^{N} \Big(\|a_i^{sd1} - x_i\|^2 + \|a_i^{sd2} - a_i^{se1}\|^2 \\ &+ \|a_i^{td1} - x_i\|^2 \\ &+ \|a_i^{td2} - a_i^{te1}\|^2 \Big) \end{aligned} \tag{12}$$

In contrast to [40] which trains dual AEs separately, the DSAF combines the training of four AEs in the dual SAE as a single reconstruction loss to facilitate the end-to-end training. To train the deep clustering network, we adopt an auxiliary target variable Q defined in [12]:

$$q_{ik} = \frac{p_{ik}/(\sum i' p_{i'k})^{\frac{1}{2}}}{\sum k' p_{ik'}/\,(\sum i' p_{i'k'})^{\frac{1}{2}}} \tag{13}$$

Then, we define a variable f as follows:

$$f_k = P(y = k) = \frac{1}{N} \sum_i q_{ik} \tag{14}$$

We denote the variable μ as a uniform distribution. Finally, the clustering objective function is defined as follows:

$$\begin{aligned} L_c &= KL(Q\|P) + KL(f\|\mu) \\ &= \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}} \right) + \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} f_k \log \frac{f_k}{\mu_k} \right) \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}} + q_{ik} \log \frac{f_k}{\mu_k} \end{aligned} \tag{15}$$

where KL($Q\|P$) denotes the clustering precision while KL($f\|\mu$) denotes the balanced assignment loss. KL($Q\|P$) is the kullback-leibler (KL) divergence between P and Q while KL($f\|\mu$) is designed to avoid too many feature points being allocated to a few clusters or assign outlier feature points to a cluster. In our experiments, we set $\alpha = 0.01$ and use the stochastic gradient descent (SGD) [41] with a learning rate of 0.01 to optimize parameters of the DSAFEC.

C. The DSAFEC based BOW (B-DSAFEC) and its application to HAR

Then, a BOW model (B-DSAFEC) is constructed using cluster assignment probabilities generated by the DSAFEC. When building a BOW vector, hard cluster assignment [14] is a natural choice in most situations which assigns feature point x_i to the k^{th} cluster with the highest probability. The k is deduced as follows:

$$k' = \arg\max_k \; p_{ik} \tag{16}$$

After assignment, a BOW vector $v = [v_1, v_2, ..., v_K] \in R^K$ is built for each video or image as follows:

$$v_k = \sum_{i=1}^{m'} s_{ik} p_{ik} \tag{17}$$

where s_{ik}, m', and v_k denote whether x_i is assigned to k^{th} cluster ($s_{ik} = 1$) or not ($s_{ik} = 0$), the number of local feature points extracted from the video or image, and the k^{th} element of the BOW vector v, respectively.

In this paper, the STIP is used as the local feature extraction method for video sequences. The radial basis function neural network (RBFNN) is used as the classifier in our experiments for its fast training speed and good generalization capacity [23] [24]. The number of nodes on the hidden layer of the RBFNN is set to be 120 as in [23].

The application of the B-DSAFEC to HAR consists of multiple steps. Given a set of video sequences, local feature points are extracted using the STIP algorithm to build a DSAFEC model for clustering. Then, based on the cluster assignment probabilities predicted by the trained DSAFEC model, a BOW vector is built for each video sequence according to Eq. (17). Finally, the generated BOW vectors of video sequences are used for training and testing a RBFNN model on HAR. The overall procedure of the B-DSAFEC for HAR is shown in Algorithm 1 and Fig. 2 presents the detailed work flowing of the B-DSAFEC.

D. B-DSAFEC with soft cluster assignment

The B-DSAFEC with hard assignment assigns a feature point to one cluster only. The B-DSAFEC can also use a soft cluster assignment [14] (B-DSAFEC-S) to assign a feature point to multiple clusters to improve its performance. The trained DSAFEC model outputs K probability values which indicate the probability of a feature point belonging to different clusters. In hard assignment, the cluster with the largest probability is assigned. In the B-DSAFEC-S, the top T largest probabilities are kept while others are change to zero. Then, cluster assignment probabilities (p_{ik}) of a feature point are then normalized to the range of $[0, 1]$. Finally, a BOW vector is built for each video sequence with the following probabilities:

$$v_k = \sum_{i=1}^{m'} p_{ik} \tag{18}$$

The value of T can be selected from powers of two which ranges between one and K. In particular, the B-DSAFEC-S reduces to B-DSAFEC when $T = 1$.

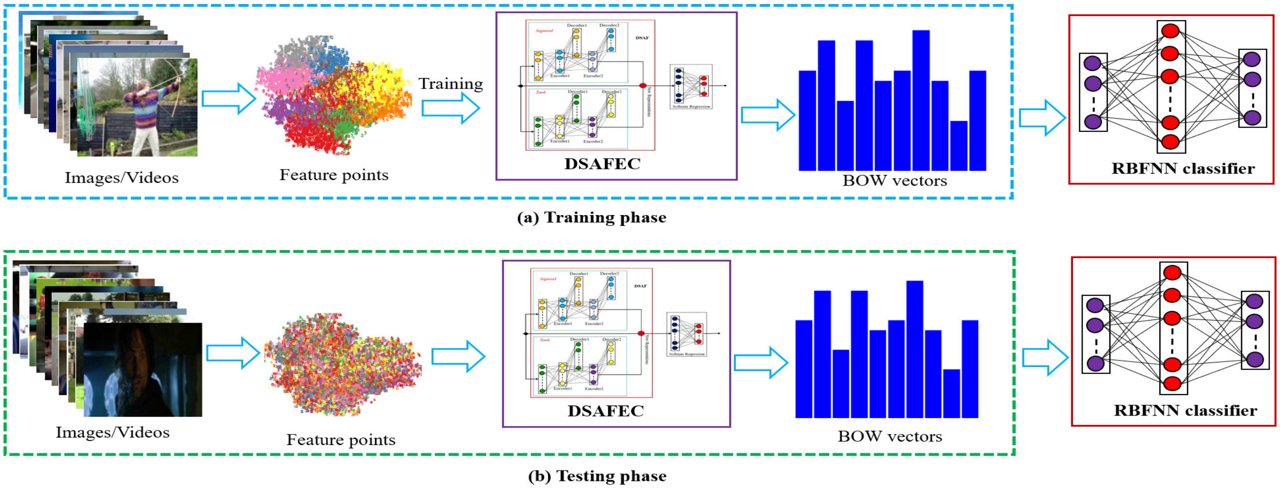


Fig. 2. The flowchart of the B-DSAFEC.

Algorithm 1 The procedure of the B-DSAFEC for HAR

```
Input: The dataset $X$
Output: The predicted activity label
    Training phase
    1: Extract local feature points from video sequences of training set using the STIP algorithm.
    2: Train the DSAFEC clustering model using STIP feature points extracted in Step 1 to form $K$ clusters.
    3: Calculate cluster assignment probabilities for STIP feature points with the trained DSAFEC in Step 2 according to the Eqs. (9) and (10).
    4: Build a BOW vector for each video sequence according to the Eq. (17) with the probabilities generated in Step 3.
    5: Train a RBFNN using the BOW vectors built in Step 4.
    Testing phase
    6: Apply the trained DSAFEC to predict cluster assignment probabilities for the STIP feature points of the test set according to the Eqs. (9) and (10).
    7: Build a BOW vector for each video sequence according to the Eq. (17) with the probabilities generated in Step 6.
    8: Predict activity labels for BOW vectors built in Step 7 with the RBFNN trained in Step 5.
```

E. The computation complexity of the DSAFEC

As aforementioned, the input dimensionality of the DSAFEC is d_x. In a stacked autoenocder, we set the dimensions of the first and the second encoding layers to be $2d_x$ and $4d_x$, respectively, as shown in Fig. 3. The dimensionality of a decoding layer is the same as its corresponding encoding layer. The outputs of 2^{nd} encoding layers in sigmoid stacked autoencoders, tanh stacked autoencoders, and the origin input are concatenated to form a new representation with dimensionality of $9d_x$. Finally, the new representation is fed into a softmax layer with dimensionality of K. The time complexity of training the first and second AE in the SAE is $O(d_x \times 2d_x \times d_x) = O(2d_x^3)$ and $O(2d_x \times 4d_x \times 2d_x) = O(16d_x^3)$. The time complexity of training the regression layer is $O(9d_xK)$. As a result, the computation complexity of the DSAFEC is about $O(2 \times (2d_x^3 + 16d_x^3) + 9d_xK) = O(36d_x^3 + 9d_xK)$.

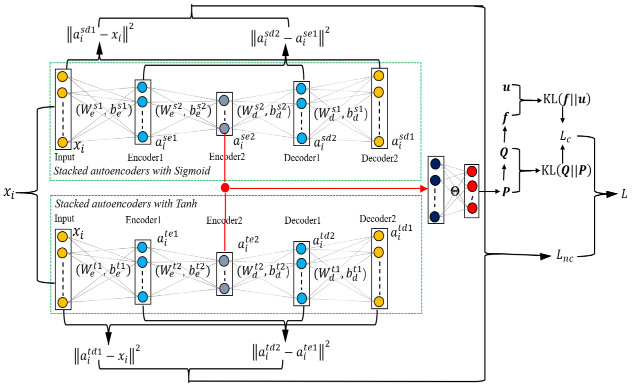


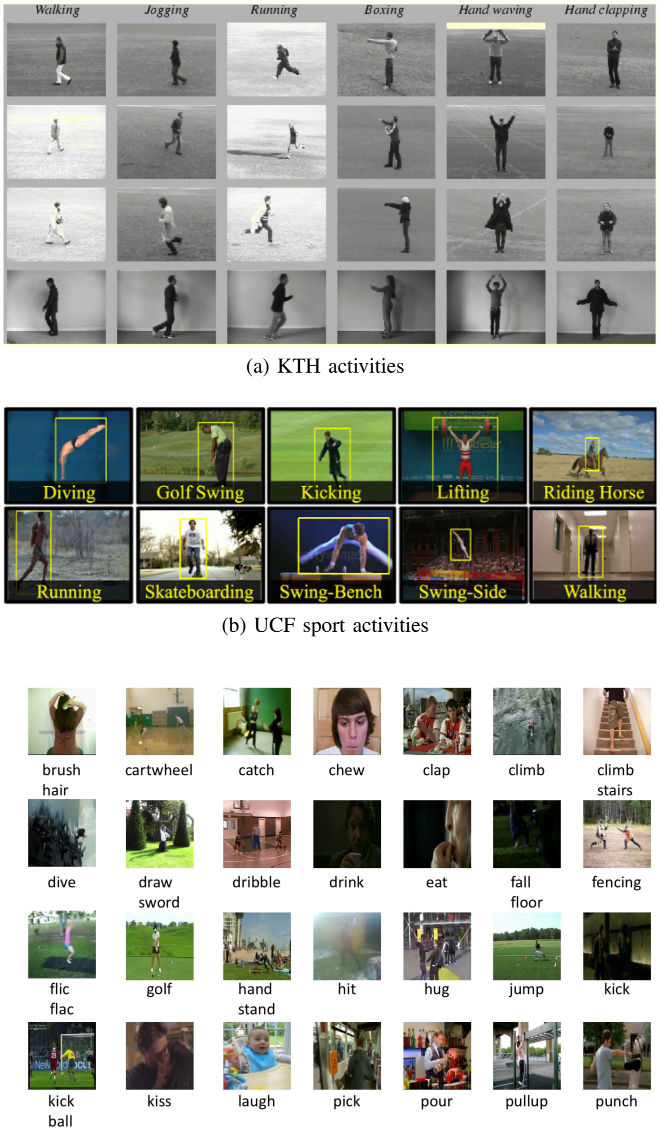
Fig. 3. The joint training of the B-DSAFEC.

IV. Experiments

A. Experimental setup

The proposed B-DSAFEC is tested on three widely used benchmarking human activity video datasets: the KTH video dataset [42], the UCF Sports video dataset [43], and the HMDB51 dataset [44]. For fair comparison to other methods, the hard cluster assignment version of B-DSAFEC is used.

The KTH activity dataset consists of six human activity classes and 600 video sequences with static backgrounds, with each class having the same number of video samples. The UCF sports datasets consists of 10 human activities and 150 video samples with a large intra-class variability. The number of instances of each activity varies within the dataset. The horizontally flipped version of each video sequence is added to double the number of instances as in [45]. The HMDB51 dataset is a large human motion database released by the Brown University in 2011. Most of the video clips are collected from various sources and movies. A small proportion of videos are collected from public databases such as YouTube, the Prelinger archive, and Google videos. The dataset contsists of 6849 clips divided into 51 action categories, each containing a minimum of 101 clips. The actions categories corresponding to a variety of human actions, such as the general facial actions, facial actions with object manipulation, general body movement, body movement with object interaction, and body movement for human interaction [44]. Fig. 4 gives some examples for these datasets.



(a) KTH activities

(b) UCF sport activities

(c) HMDB51 activities

Fig. 4. Some examples for the KTH, the UCF Sports, and the HMDB51 videos.

For these datasets, one-third of the total amount of instances are randomly selected to be the testing set while the rest of video sequences are used as the training set. This is repeated for three times with a random one-third instances selected each time and the average accuracy over all the classes is reported for the three training-testing splits.

In our experiments, both traditional clustering-based methods (the K-means [4] and the AGNES [5]) and deep clustering-based methods (the CNNKMS [32], the DEPICT [12], and the SR-K-means [37]) are compared. The CNNKMS first extracts CNN features of video frames using the ResNet50 [46] and uses K-means to build a visual vocabulary with these features. As aforementioned, the K-means and the AGNES are not suited for problems with a very large number of feature points. Therefore, to reduce the computational complexity, $100,000$ feature points are randomly sampled from the training set to construct the visual vocabulary with them. Other deep clustering-based methods are not restricted by the number of feature points, so all feature points extracted from the training set are used for training. BOW models yielded by the K-means, the AGNES, the CNNKMS, the DEPICT, and the SR-K-means are named by the B-K-means, the B-AGNES, the B-CNNKMS, the B-DEPICT, and the B-SR-K-means, respectively.

B. Experimental results and analysis

In this experiment, we compare performances of the B-DSAFEC with different reference methods equipped with different sizes of visual vocabularies (K), where $K = 2^{\eta}, \eta = 7, 8, ..., 13$.

Table I shows accuracy yielded by the B-DSAFEC and reference methods with different K on the KTH activity dataset. The proposed B-DSAFEC yields the best average performance while the B-CNNKMS yields the worst performance among all methods. This indicates that a simple combination of a CNN and a K-means clustering does not provide a good BOW model for HAR. The B-DSAFEC yields the best performance for all K except for $K = 8192$. The B-K-means yields the best accuracy of 90.66% when $K = 8192$, but it is worse than the B-DSAFEC with $K = 256$. This shows that the B-DSAFEC learns much better BOW and is able to use small K to yield a good BOW model.

TABLE I
THE ACCURACY OF REFERENCE METHODS WITH DIFFERENT SIZES OF VISUAL VOCABULARY ON THE KTH DATASET.

K	B-K-means	B-AGNES	B-CNNKMS	B-DEPICT	B-SR-K-means	B-DSAFEC
128	0.8207	0.7660	0.4899	0.8468	0.7121	**0.8855**
256	0.8561	0.8182	0.4566	0.8754	0.7525	**0.9074**
512	0.8699	0.8552	0.4091	0.9024	0.8737	**0.9242**
1024	0.8674	0.8552	0.3626	0.9091	0.8939	**0.9125**
2048	0.8813	0.8822	0.3212	0.9158	0.8687	**0.9360**
4096	0.8838	0.8822	0.3212	0.9259	0.8788	**0.9343**
8192	0.9066	0.8670	0.2960	**0.9461**	0.8990	0.9377
Average	0.8694	0.8466	0.3795	0.9031	0.8398	**0.9197**

Table II shows accuracy of the B-DSAFEC and reference methods with different K values on the UCF activity dataset. The B-DSAFEC yields the best performance for $K = 128$, 256, 2048, 4096, and 8192 while yields the second best performance for $K = 512$ and 1024 (only worse than the B-DEPICT). The B-DSAFEC with $K = 2048$ yields the best accuracy of 82.82% which outperforms all other methods with $K = 8192$. From both Tables I and II, the B-DSAFEC yields 3.87% and 5.50% better performances in comparison with the second-best B-DEPICT and B-CNNKMS, respectively. The B-DSAFEC yields larger advantages when K is small which indicates a better BOW model learning by the B-DSAFEC. Table III shows accuracy of the B-DSAFEC and reference methods with different K values on the HMDB51 activity dataset. The B-DSAFEC yields the best performance for $K = 128$, 256, 512, 1024, 2048, and 4096 while the B-K-means yields the best performance for $K = 8192$. The average accuracy in Table III shows that the proposed method yields a good results even for very large dataset like the HMDB51.

TABLE II
THE ACCURACY OF THE REFERENCE METHODS WITH VARYING SIZES OF VISUAL VOCABULARY ON THE UCF DATASET.

K	B-K-means	B-AGNES	B-CNNKMS	B-DEPICT	B-SR-K-means	B-DSAFEC
128	0.6357	0.6082	0.7711	0.6976	0.5979	**0.7766**
256	0.7113	0.6426	0.7794	0.7595	0.6495	**0.7835**
512	0.7766	0.6529	0.7711	**0.8110**	0.6392	0.7835
1024	0.7835	0.6873	0.7691	**0.8179**	0.7113	0.8110
2048	0.8041	0.6598	0.7526	0.8144	0.6598	**0.8282**
4096	0.8110	0.6667	0.7278	0.8007	0.7423	**0.8213**
8192	0.7973	0.6873	0.5443	0.8007	0.7010	**0.8110**
Average	0.7599	0.6578	0.7308	0.7860	0.6716	**0.8022**

TABLE III
THE ACCURACY OF THE REFERENCE METHODS WITH VARYING SIZES OF VISUAL VOCABULARY ON THE HMDB51 DATASET.

K	B-K-means	B-AGNES	B-CNNKMS	B-DEPICT	B-SR-K-means	B-DSAFEC
128	0.3235	0.3123	0.3274	0.7675	0.3889	**0.8307**
256	0.3274	0.3706	0.3503	0.8196	0.4541	**0.9013**
512	0.6189	0.4381	0.3791	0.7868	0.4576	**0.8568**
1024	0.7339	0.5159	0.4307	0.7834	0.6662	**0.8934**
2048	0.8300	0.5896	0.4549	0.8355	0.7930	**0.9045**
4096	0.9045	0.6985	0.4902	0.9067	0.8459	**0.9379**
8192	**0.9437**	0.7871	0.5209	0.8627	0.8846	0.9320
Average	0.6688	0.5303	0.4219	0.8232	0.6239	**0.8938**

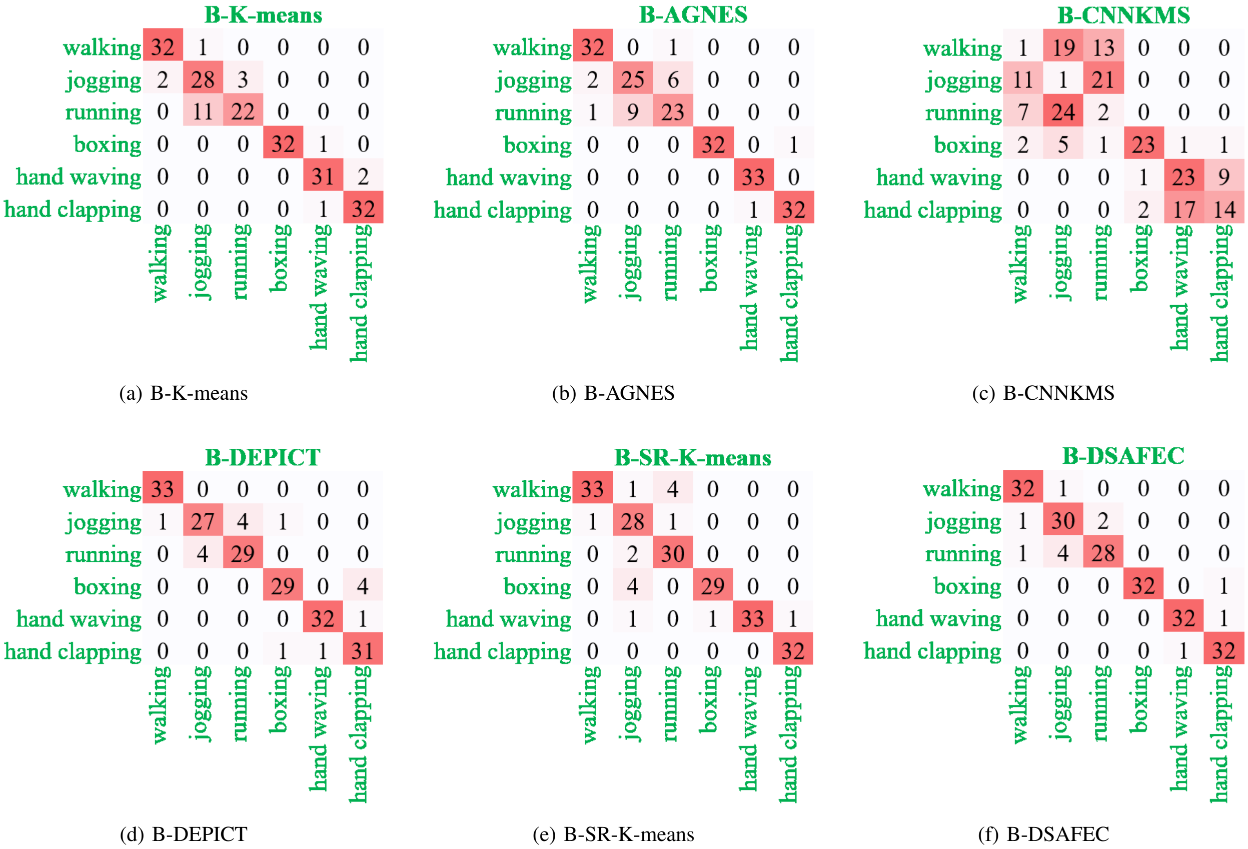
Overall, the B-DSAFEC yields the best performance over other reference methods. It demonstrates that the DSAF learns more discriminative feature representations for clustering tasks. The B-DSAFEC is robust to the selection of K and yields a promising performance even when K is extremely small (e.g. 128). Furthermore, the local representation methods may be better than deep clustering methods in small datasets like the KTH and UCF datasets for HAR. In addition, Tables I, II, and III show that the optimal K values vary for different methods and the relationship between accuracies and the value of K does not monotonically increase for all reference methods. This is because clustering performance heavily depends on the nature of the input data and the algorithm itself, and these human activity video datasets have their own characteristics, for example, UCF dataset contains a large intra-class variability. Therefore, there is no guarantee that the clustering performance will improve with the increase of the K value even if the above deep clustering-based learning methods map the input data onto a latent feature space where grouping tasks become much easier. This is also why the value of K is so difficult to be determined for most clustering methods, and the value of K is often determined via either trial-and-error or cross-validation methods.

C. Further analysis of experimental results

To further analyze the performance of the proposed B-DSAFEC and reference methods on each action class of each dataset, four other performance indices are employed in this study, including confusion matrix, AUC, F_1-score, and G-mean [40]. Multi-class HAR problems are inheritantly imbalanced because the classification of each action versus all other classes is imbalanced. Furthermore, different action classes may have different number of instances. Overall accuracy may not be enough to reflect the performance of a method for HAR problems. Therefore, these metrics are needed to further analyze performances of different methods. In a confusion matrix, row and column stand for true labels and predicted labels, respectively. The value in each entry denotes the number of samples in the corresponding label class in the row being predicted as the corresponding class in the column. AUC measures the area under receiver operating characteristic curve. The F_1-score is the harmonic mean of the precision and recall of the positive class. In the experiment, each class in a dataset takes turn to be the positive class while remaining classes are used as negative classes. The average value over all classes is then recorded. The G-mean measures the geometric average precision of the positive class and the negative class. The average values of AUCs, F_1-scores, and G-means over all classes in a dataset are represented as $mAUC$, mF_1, and mG, respectively. In this experiment, we choose $K = 4096$, which is widely used in BOW models for HAR [3] [7] [21] [45].

Confusion matrices of the B-DSAFEC and reference methods on the KTH dataset are depicted in Fig. 5. The $mAUC$, mF_1, and mG of these methods on the KTH dataset are listed in Table IV. As shown in Fig. 5, the B-DSAFEC and reference methods show the worst performances on the $jogging$ and $running$ classes. The same situations appear between $walking/jogging$ and $handwaving/handclapping$. It is reasonable because these actions are similar in the real world. With these difficult HAR tasks, the proposed method outperforms other methods with less incorrect predictions on these classes. The B-CNNKMS performs even worse than the B-K-means. It indicates that simple concatenation of CNN with K-means may hamper the performance and dedicated representation learning methods for clustering should be employed, for examples B-DSAFEC and B-DEPICT. Table IV shows that the B-DSAFEC yields the best $mAUC$, mF_1, and mG on the KTH dataset over reference methods.

Confusion matrices of the B-DSAFEC and reference methods on the UCF dataset are depicted in Fig. 6 while $mAUC$, mF_1, and mG of these methods on the UCF dataset are



B-K-means

	walking	jogging	running	boxing	hand waving	hand clapping
walking	32	1	0	0	0	0
jogging	2	28	3	0	0	0
running	0	11	22	0	0	0
boxing	0	0	0	32	1	0
hand waving	0	0	0	0	31	2
hand clapping	0	0	0	0	1	32

(a) B-K-means

B-AGNES

	walking	jogging	running	boxing	hand waving	hand clapping
walking	32	0	1	0	0	0
jogging	2	25	6	0	0	0
running	1	9	23	0	0	0
boxing	0	0	0	32	0	1
hand waving	0	0	0	0	33	0
hand clapping	0	0	0	0	1	32

(b) B-AGNES

B-CNNKMS

	walking	jogging	running	boxing	hand waving	hand clapping
walking	1	19	13	0	0	0
jogging	11	1	21	0	0	0
running	7	24	2	0	0	0
boxing	2	5	1	23	1	1
hand waving	0	0	0	1	23	9
hand clapping	0	0	0	2	17	14

(c) B-CNNKMS

B-DEPICT

	walking	jogging	running	boxing	hand waving	hand clapping
walking	33	0	0	0	0	0
jogging	1	27	4	1	0	0
running	0	4	29	0	0	0
boxing	0	0	0	29	0	4
hand waving	0	0	0	0	32	1
hand clapping	0	0	0	1	1	31

(d) B-DEPICT

B-SR-K-means

	walking	jogging	running	boxing	hand waving	hand clapping
walking	33	1	4	0	0	0
jogging	1	28	1	0	0	0
running	0	2	30	0	0	0
boxing	0	4	0	29	0	0
hand waving	0	1	0	1	33	1
hand clapping	0	0	0	0	0	32

(e) B-SR-K-means

B-DSAFEC

	walking	jogging	running	boxing	hand waving	hand clapping
walking	32	1	0	0	0	0
jogging	1	30	2	0	0	0
running	1	4	28	0	0	0
boxing	0	0	0	32	0	1
hand waving	0	0	0	0	32	1
hand clapping	0	0	0	0	1	32

(f) B-DSAFEC

Fig. 5. The visualization results of confusion matrices of reference methods on the KTH dataset.

TABLE IV
PERFORMANCE OF DIFFERENT METHODS ON THE KTH DATASET.

methods	$mAUC$	mF_1	mG
B-K-means	0.9364	0.8934	0.9336
B-AGNES	0.9364	0.8929	0.9337
B-CNNKMS	0.5939	0.3352	0.4625
B-DEPICT	0.9485	0.9137	0.9474
B-SR-K-means	0.8879	0.7671	0.8870
B-DSAFEC	**0.9636**	**0.9393**	**0.9630**

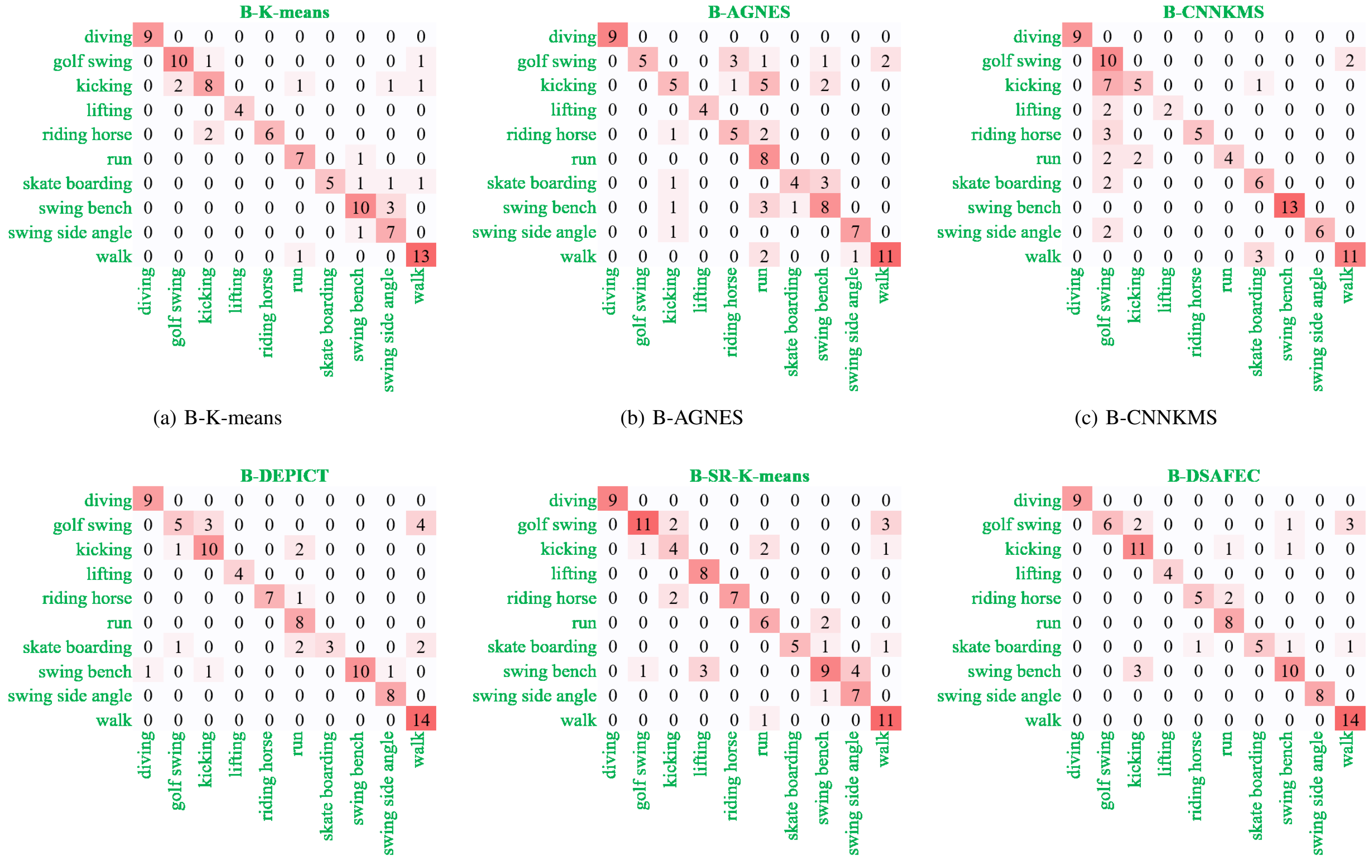
listed in Table V. As shown in Fig. 6, the proposed method and reference methods yield worse performances on actions of kicking and run. The B-CNNKMS misclassifies a lot of instances in different action classes to be golf swing because they have some common hand actions. The B-AGNES has a similar deffiiciency. Table V shows that the B-DSAFEC yields the best $mAUC$, mF_1, and mG on the UCF activity dataset over reference methods. Especially, due to the fact that there are 51 categories in the HMDB51 dataset, the confusion matrix cannot be clearly presented. Therefore, the confusion matrix of the HMDB51 is omitted in this section. From Table VI, the B-DSAFEC yields the best $mAUC$, mF_1, and mG on the HMDB51 dataset over reference methods. The performances of all reference methods in some actions are not prominent, such as $punch$, run, $smile$, and $swordexercise$. It is reasonable because these actions are relatively complicated. For example, everyone has different running postures and speed. Even the same person in different scenes can not guarantee to perform the same action in the exact same way. Thus, complex actions reduce the performance of models.

TABLE V
PERFORMANCE OF DIFFERENT METHODS ON THE UCF DATASET.

methods	$mAUC$	mF_1	mG
B-K-means	0.9031	0.8286	0.8970
B-AGNES	0.8423	0.7080	0.8212
B-CNNKMS	0.8412	0.7441	0.8216
B-DEPICT	0.8992	0.8089	0.8842
B-SR-K-means	0.8750	0.8235	0.8613
B-DSAFEC	**0.9083**	**0.8369**	**0.8990**

D. The performance of B-DSAFEC with soft cluster assignment

As mentioned in Section III-D, this experiment explores the influence of the soft cluster assignment and the hard cluster assignment strategies on the performance of the model for HAR. The outputs of the B-DSAFEC are probabilities of a STIP feature point belonging to different clusters. So, this outputs can be directly used as the probability values for the soft cluster assignment. Fig. 7 shows the accuracy of the B-DSAFEC-S on the KTH, the UCF, and the HMDB51 datasets



(a) B-K-means (b) B-AGNES (c) B-CNNKMS

(d) B-DEPICT (e) B-SR-K-means (f) B-DSAFEC

Fig. 6. The visualization of the results of the confusion matrices of reference methods on the UCF dataset.

TABLE VI
PERFORMANCE OF DIFFERENT METHODS ON THE HMDB51 DATASET.

methods	$mAUC$	mF_1	mG
B-K-means	0.9276	0.7123	0.9256
B-AGNES	0.7133	0.4905	0.6561
B-CNNKMS	0.7573	0.4324	0.7234
B-DEPICT	0.8940	0.6667	0.8890
B-SR-K-means	0.8967	0.7500	0.8914
B-DSAFEC	**0.9290**	**0.7536**	**0.9602**

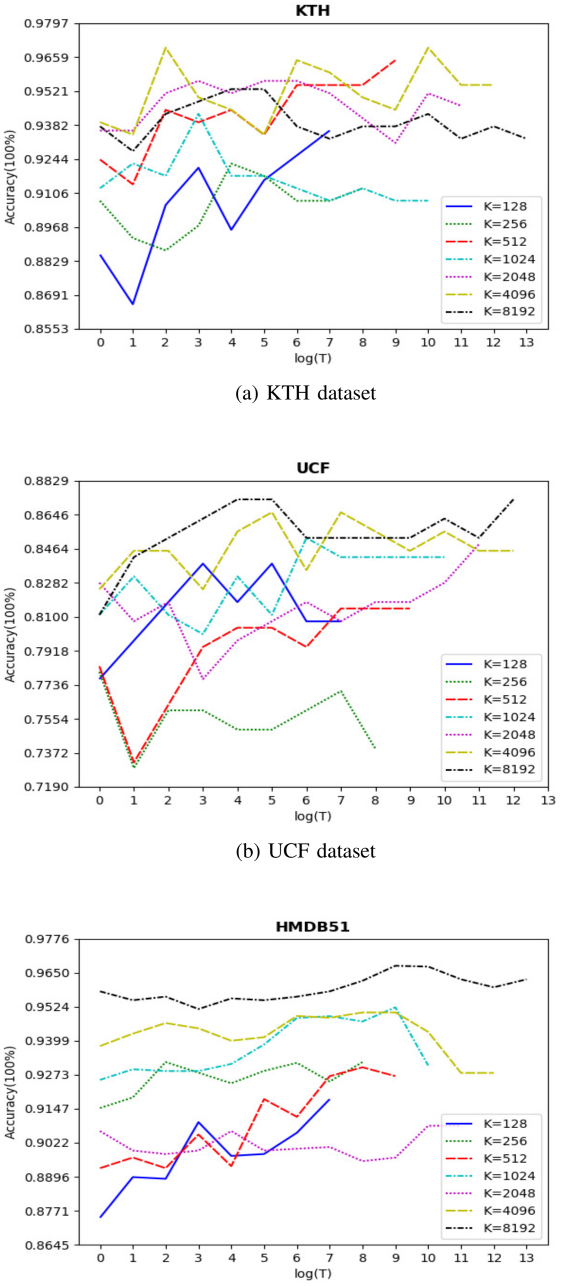
with respect to the logarithm to base two of $T(\log T)$. The hard cluster assignment is a special case of the soft cluster assignment when $\log T = 0$. So, in Fig. 7, results yielded by $\log T = 0$ are for the hard cluster assignment's. The performance of the B-DSAFEC-S varies greatly with the increment of $\log T$, and the soft cluster assignment outperforms the hard cluster assignments in the majority of cases of $\log T$ with different K values. In the KTH dataset, the B-DSAFEC-S with $\log T = 7$ yields 5.05% better accuracy than the B-DSAFEC when $K = 128$. Although the soft cluster assignment strategy improves the accuracy on the KTH by 1.51%, it has no significant improvement on the UCF when $K = 256$. The B-DSAFEC-S with $\log T = 3$ yields 6.19% better accuracy than the B-DSAFEC when $K = 128$. When $K = 256$ for the UCF dataset, the B-DSAFEC-S predicts cluster assignment probabilities close to zero or one which makes the soft cluster assignment fail to improve the performance. Similarly, the B-DSAFEC-S with $\log T = 7$ yields 4.75% better accuracy than the B-DSAFEC when $K = 128$ for the HMDB51 dataset. When $K = 2048$, the range of accuracy is not large and the contribution of soft clustering to improvement of B-DSAFEC performance is not outstanding. Results on the UCF and the HMDB51 datasets are not available due to a memory error when $K = 8192$ and $logT = 14$. The fundamental reason is that the UCF and the HMDB51 datasets have more feature points than the KTH dataset.

In summary, soft cluster assignment improves the performance of the B-DSAFEC substantially when compared with the hard cluster assignment.

E. Training manner of the DSAFEC

The performance of the end-to-end joint learning in the B-DSAFEC is compared with a greedy layer-wise disjoint learning [8] version of the B-DSAFEC. The greedy layer-wise disjoint learning method trains the DSAF first, and then uses its outputs to train parameters of the softmax regression.

Accuracy of the B-DSAFEC with end-to-end joint learning and greedy layer-wise disjoint learning are shown in Fig. 8. It can be observed that the end-to-end joint learning always



(a) KTH dataset

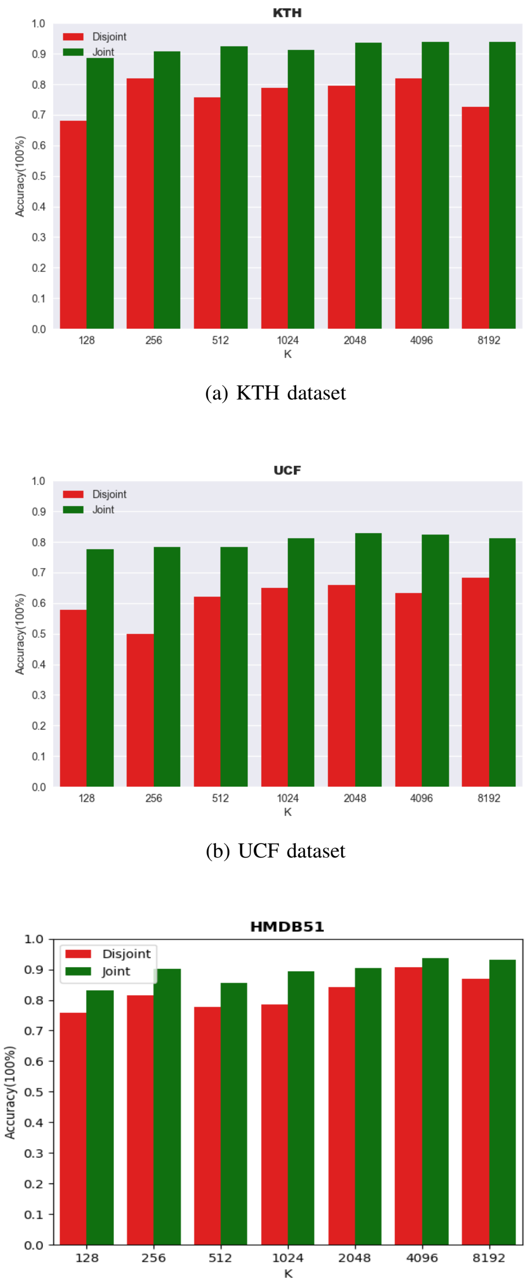
(b) UCF dataset

(c) HMDB51 dataset

Fig. 7. The accuracy of the B-DSAFEC with soft cluster assignment on the KTH, the UCF, and the HMDB51 datasets with respect to the logarithm to base two of T ($\log T$).

outperforms the greedy layer-wise disjoint learning on the KTH, the UCF, and the HMDB51 datasets in all cases of K. The B-DSAFEC with end-to-end joint learning improves the performance by 21.04% when $K = 8192$, 28.51% when $K = 256$, and 18.08% when $K = 4096$ on the KTH, the UCF, and the HMDB51 datasets, respectively.

Experimental results confirm that training the DSAFEC with the end-to-end joint learning method is more appropriate than the greedy layer-wise disjoint learning method. This shows that the unified objective function may prevent negative effects of optimizing several different objectives.



(a) KTH dataset

(b) UCF dataset

(c) HMDB51 dataset

Fig. 8. The performance of joint learning and disjoint learning of the B-DSAFEC on the KTH, the UCF, and the HMDB51 datasets.

F. Discriminative histograms of different classes

In this subsection, we carried out experiments to investigate the discriminative ability of the proposed method on the KTH, the UCF, and the HMDB51 datasets, respectively. Figs. 9, 10, and 11 show histograms generated by the proposed B-DSAFEC with $K = 128$ for six videos from four classes, namely $hand{-}clapping$, $golf{-}swing$, $ride{-}bike$, and $ride{-}horse$. Fig. 9 depicts two people clapping hands with different gestures from the KTH dataset. Fig. 10 depicts two people swinging golf from different angles from the UCF dataset. Fig. 11 (a) depicts people riding a bike heading to the left and Fig. 11 (b) depicts people riding a horse heading to the right. Both Fig. 11 (a) and (b) are from the HMDB51 dataset. These figures show that histograms of the two videos are very discriminative even when human actions are similar in scenes. This demonstrates the discriminative power of the B-DSAFEC for HAR.

G. Computational efficiency analysis

A good model is expected to yield both high efficiency and high accuracy. Our experiments in previous sections show that our proposed method yields high accuracy performance over a number of different datasets. In this section, we compare computational efficiencies of different methods by comparing their run time on the HMDB51 dataset using Python 3.6 and a personal computer with Window10, 8 GB RAM, an Intel Core i5-95000 3.00GHz CPU, and a GeForce RTX 2080Ti GPU. Table VII presents run time of different methods on the HMDB51 using the aforementioned computer. The numbers of feature points for the training and the testing sets are 4,364,865 and 1,831,961, respectively. The DSAFEC uses 20.83 seconds on average for each training epoch with a batch size of 8192. The average accuracy of our model is consistently greater than 80% after 10 epochs and converges before 100 epochs. It uses 8.74 seconds on average for the testing. It is also worth noting that the SR-K-means takes the longest computational time. The main reason is that the objective function of SR-K-means integrates discriminative models with the K-means. A large computational time is needed by the SR-K-means to obtain balanced parameters in optimization processes for both latent cluster assignments and deep network parameters. Overall, the proposed method demonstrates the computational advantage over other bench marking approaches.

V. Conclusion

In this paper, the dual stacked autoencoders features embedded clustering (DSAFEC) and a novel BOW based on the DSAFEC (B-DSAFEC) as an efficient deep clustering algorithm are proposed for HAR tasks. The DSAFEC predicts cluster assignment probabilities of feature points and then the B-DSAFEC uses these probabilities to build BOW vectors for HAR. Experimental results show that the proposed B-DSAFEC outperforms BOW models built either using traditional clustering algorithms or using deep clustering methods. Experimental results also demonstrate that end-to-end joint learning is more appropriate than greedy layer-wise disjoint learning for training of the DSAFEC. The soft cluster assignment improves the performance of the B-DSAFEC.

In this work, the deep clustering is not integrated with the final classification phase. Therefore, a more sophisticated combination of the classification and the clustering could be further investigated for further improvement of HAR. In addition to activity recognition, the proposed method can also be applied to other computer vision tasks, such as image classification and segmentation.

References

[1] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2016.

[2] A. Richard and J. Gall, "A bag-of-words equivalent recurrent neural network for action recognition," *Computer Vision and Image Understanding*, vol. 156, pp. 79–91, 2017.

[3] J. R. Cózar, J. M. González-Linares, N. Guil, R. Hernández, and Y. Heredia, "Visual words selection for human action classification," in *2012 International Conference on High Performance Computing Simulation (HPCS)*, 2012, pp. 188–194.

[4] S. Yu, S. W. Chu, C. Wang, Y. Chan, and T. Chang, "Two improved k-means algorithms," *Applied Soft Computing*, vol. 68, pp. 747–755, 2017.

[5] M. Roux, "A comparative study of divisive and agglomerative hierarchical clustering algorithms," *Journal of Classification*, vol. 35, no. 2, pp. 345–366, 2018.

[6] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 1985–1994.

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[8] L. Shao, S. Jones, and X. Li, "Efficient search and localization of human actions in video databases," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 504–512, 2014.

[9] Y. Gang, J. Yuan, and Z. Liu, "Unsupervised random forest indexing for fast action search," in *Computer Vision & Pattern Recognition*, 2011.

[10] J. Chang, G. Meng, L. Wang, S. Xiang, and C. Pan, "Deep self-evolution clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 809–823, 2020.

[11] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *IEEE International Conference on Computer Vision*, 2017, pp. 5747–5756.

[12] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5147–5156.

[13] E. Aljalbout, V. Golkov, Y. Siddiqui, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," *arXiv: Learning*, 2018.

[14] X. Zhen and L. Shao, "Action recognition via spatio-temporal local features: A comprehensive study," *Image and Vision Computing*, vol. 50, pp. 1–13, 2016.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[17] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *2013 IEEE International Conference on Computer Vision*, vol. 159, 2013, pp. 3551–3558.

[18] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, 2005.

[19] X. Yan, S. Hu, and Y. Ye, "Multi-task clustering of human actions by sharing information," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6401–6409.

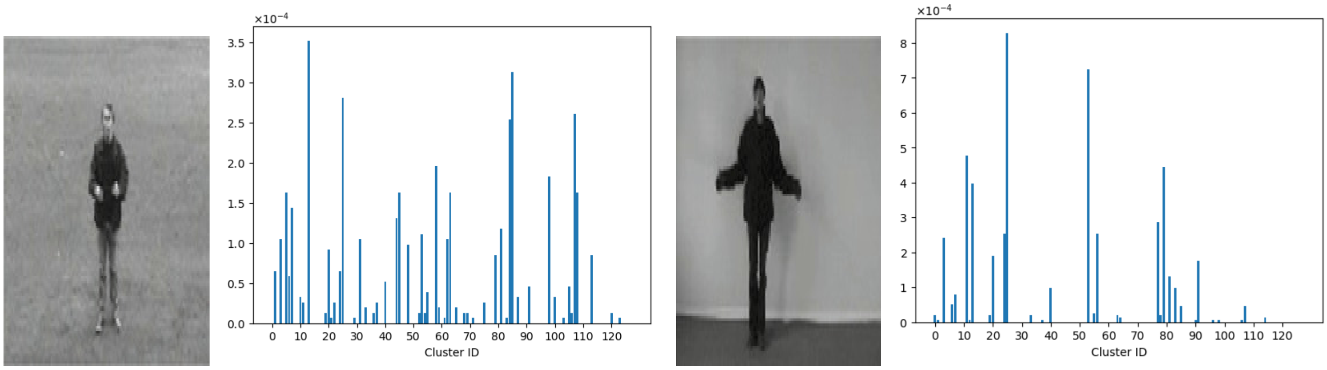
[20] X. Zhen, F. Zheng, L. Shao, X. Cao, and D. Xu, "Supervised local descriptor learning for human action recognition," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2056–2065, 2017.

[21] B. Tahayna, M. Belkhatir, S. M. Alhashmi, and T. O'Daniel, "Human action detection and classification using optimal bag-of-words representation," in *6th International Conference on Digital Content, Multimedia Technology and its Applications*, 2010, pp. 75–80.

[22] Q. Wu, Z. Wang, F. Deng, Y. Xia, W. Kang, and D. D. Feng, "Discriminative two-level feature selection for realistic human action recognition," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1064–1074, 2013.

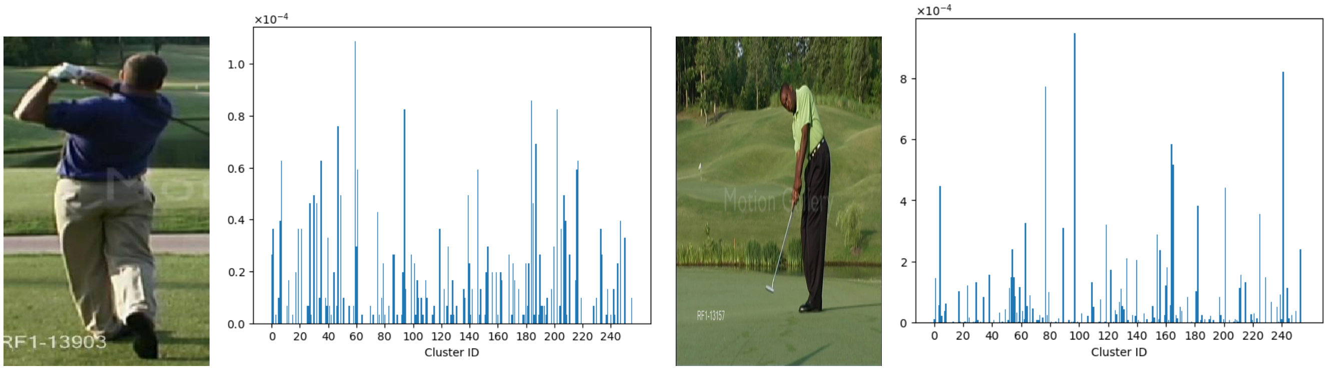
[23] W. W. Y. Ng, J. Li, J. Zhang, Q. Wu, and J. Li, "Visual words selection for human action recognition using rbfnn via the minimization of l-gem," in *2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, 2017, pp. 183–187.

[24] D. S. Yeung, W. W. Y. Ng, D. Wang, E. C. C. Tsang, and X. Wang, "Localized generalization error model and its application to architecture



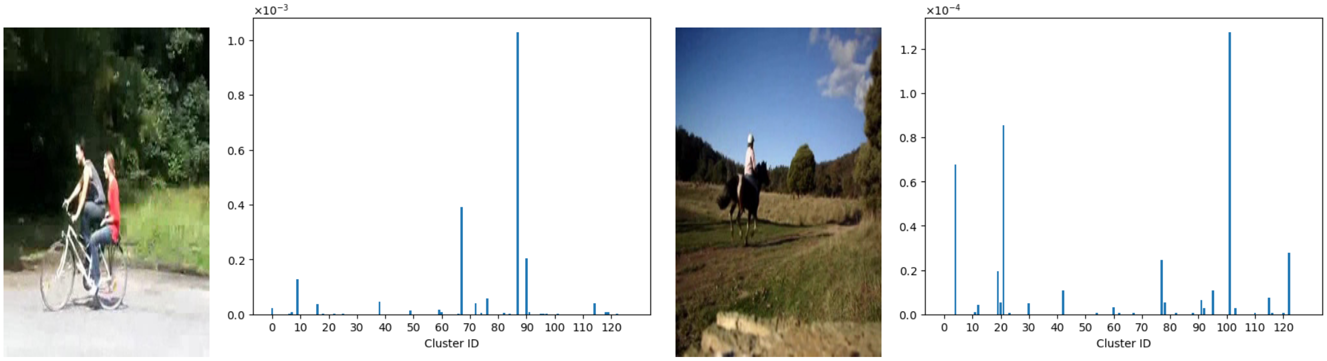
(a) $hand{-}clapping$ (b) $hand{-}clapping$

Fig. 9. Comparison of the histograms generated by the B-DSAFEC for two different actions of same class from the KTH dataset. Example image (left) and the B-DSAFEC (right) for the videos $hand{-}clapping$ in (a) and (b), respectively.



(a) $golf{-}swing{-}back$ (b) $golf{-}swing{-}front$

Fig. 10. Comparison of the histograms generated by the B-DSAFEC for two different angles of same class from the UCF dataset. Example image (left) and the B-DSAFEC (right) for the videos $golf{-}swing{-}back$ and $golf{-}swing{-}front$ in (a) and (b), respectively.



(a) $ride{-}bike$ (b) $ride{-}horse$

Fig. 11. Comparison of the histograms generated by the B-DSAFEC for two videos of two different classes from the HMDB51 dataset. Example image (left) and the B-DSAFEC (right) for the videos $ride{-}bike$ and $ride{-}horse$ in (a) and (b), respectively.

selection for radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1294–1305, 2007.

[25] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1747–1756, 2020.

[26] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. G. Hauptmann, "Semi-supervised multiple feature analysis for action recognition," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 289–298, 2014.

[27] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.

[28] Y. Li, W. Wang, M. Liu, Z. Jiang, and Q. He, "Speaker clustering by co-optimizing deep representation learning and cluster estimation," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[29] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features." in *Proc. European Conference on Computer Vision*, vol. 11218, 2018, pp. 139–156.

[30] Y. Xie, B. Lin, Y. Qu, C. Li, W. Zhang, L. Ma, Y. Wen, and D. Tao, "Joint

TABLE VII
THE RUN TIME OF DIFFERENT METHODS FOR THE HMDB51 DATASET.

Run Time (s)	K=128	K=256	K=512	K=1024	K=2048	K=4096	K=8192	Average
B-K-means	489.24	1399.34	2139.26	3464.3	5560.34	7645.65	21330.79	6004.13
B-AGNES	1415.36	1570.71	1721.47	1929.35	2276.01	5045.76	9429.31	3341.14
B-CNNKMS	1498.89	1600.1	1829.94	2454.09	3565.04	6015.66	8803.04	3680.97
B-DEPICT	546.64	733.14	1116.59	1895.95	3674.57	8451.01	17158.6	4796.64
B-SR-K-means	2336.00	3631.68	4938.16	13605.23	21242.29	42640.58	74860.23	23322.02
B-DSAFEC	**216.75**	**281.44**	**405.87**	**670.65**	**1187.01**	**4279.58**	**7544.86**	**2083.73**

deep multi-view learning for image clustering," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[31] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, "Deep subspace clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[32] C. Hsu and C. Lin, "Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 421–429, 2017.

[33] S. Wu, Q. Ji, S. Wang, H. Wong, Z. Yu, and Y. Xu, "Semi-supervised image classification with self-paced cross-task networks," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 851–865, 2018.

[34] A. Abedin, F. Motlagh, Q. Shi, H. Rezatofighi, and D. Ranasinghe, "Towards deep clustering of human activities from wearables," in *Proceedings of the 2020 International Symposium on Wearable Computers*, ser. ISWC '20, 2020, p. 1–6. [Online]. Available: https://doi.org/10.1145/3410531.3414312

[35] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, and S. Avidan, "Graph embedded pose clustering for anomaly detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] A. Sahu and A. S. Chowdhury, "Summarizing egocentric videos using deep features and optimal clustering," *Neurocomputing*, vol. 398, pp. 209–221, 2020.

[37] M. Jabi, M. Pedersoli, A. Mitiche, and I. Ben Ayed, "Deep clustering: On the link between discriminative models and k-means," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[38] Y. Liu, X. Feng, and Z. Zhou, "Multimodal video classification with stacked contractive autoencoders," *Signal Processing*, vol. 120, no. 120, pp. 761–766, 2016.

[39] Y. Lei, W. Yuan, H. Wang, Y. Wenhu, and W. Bo, "A skin segmentation algorithm based on stacked autoencoders," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 740–749, 2017.

[40] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, "Dual autoencoders features for imbalance classification problem," *Pattern Recognition*, vol. 60, pp. 875–889, 2016.

[41] S. Mei, A. Montanari, and P. Nguyen, "A mean field view of the landscape of two-layer neural networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 33, p. 201806579, 2018.

[42] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, 2004, pp. 32–36.

[43] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[44] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *IEEE International Conference on Computer Vision*, 2011.

[45] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. British Machine Vision Conference*, 2009, pp. 127–137.

[46] Z. Ding, M. Shao, W. Hwang, S. Suh, J. Han, C. Choi, and Y. Fu, "Robust discriminative metric learning for image representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3173–3183, 2019.

Ms. Ting Wang (StM'19) received the master degree in computer science from Northeast Normal University, China, in 2017. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, China. Her current research interests include learning methods and generalization capabilities for deep neural networks, and their applications in real-world problems, such as smart healthcare and smart grid, etc.

Wing W. Y. Ng (S'02-M'05-SM'15) received the B.Sc. and Ph.D. degrees from Hong Kong Polytechnic University, Hong Kong, in 2001 and 2006, respectively.

He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, where he is currently the Deputy Director of the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information. His current research interests include neural networks, deep learning, smart grid, smart healthcare, smart manufacturing, and nonstationary information retrieval.

Dr. Ng is currently an Associate Editor of the International Journal of Machine Learning and Cybernetics. He is the Principle Investigator of four China National Natural Science Foundation projects, a Program for New Century Excellent Talents in University from China Ministry of Education, and a Guangdong Province Science and Technology Plan Project. He served as the Board of Governor for IEEE Systems, Man and Cybernetics Society in 2011 and 2013.

Jinde Li received the B.Sc. degree in School of Publication Health from Wuhan University, Wuhan, Chian, in 2016. He received the M.Sc. degree with the School of Computer Science and Engineering from South China University of Technology, Guangzhou, China, in 2019. His current work is mainly video classification.

Qiuxia Wu received her Ph.D. degree in 2012 from the South China University of Technology. From October 2009 to October 2011, she was a Visiting Student with The University of Sydney, Sydney, Australia. Since 2012, she had been working with the South China University of Technology as an Assistant Professor. From July 2012 to March 2016, she was with the Guangzhou Institute of Modern Industrial Technology and now she is an Associate Professor in the School of Software Engineering at the South China University of Technology. Her research interests include content-based video retrieval, biometrics recognition, and biomedical image analysis.

Shuai Zhang is a lecturer in the School of Computing and Mathematics at the Ulster University. Her research interest includes intelligent data analysis in the application areas of connected health and ambient assistive living. Her current research includes activity and behavioral recognition in smart environment, change point detection for sensor data annotation, and modelling user engagement with and adoption of assistive technologies for people with dementia.

Chris Nugent (S'96-A'99-M'03) received the BEng degree in electronic systems and the DPhil degree in biomedical engineering from the University of Ulster, Jordanstown, United Kingdom, in 1995 and 1998, respectively. In 1998, he took the post of research fellow with the University of Ulster and now currently holds the post of professor of biomedical engineering. His research interests include the design and evaluation of pervasive solutions in smart environments for ambient assisted living applications. He is a member of the IEEE.

Colin Shewell received a B.Sc. in Information and Communication Technology and went on to pursuing a Ph.D. in Computer Science at Ulster University. In 2017 he took a Research Associate position within the Connected Healthcare Innovation Centre. He is currently a lecturer in Computing Science within the School of Computing and Mathematics at Ulster University. His research interests include machine vision, connected health, and pervasive computing within the domain of ambient assisted living.