Pattern Recognition 90 (2019) 196–209

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Domain learning joint with semantic adaptation for human action recognition

Junxuan Zhang, Haifeng Hu*

School of Electronic and Information Engineering Sun Yat-sen University, Guangzhou, China

ARTICLE INFO

Article history:
Received 9 February 2018
Revised 17 November 2018
Accepted 24 January 2019
Available online 29 January 2019

MSC:
00-01
99-00

Keywords:
Knowledge adaptation
Two-stream network
Video representation
Action recognition
Cascaded convolution fusion strategy

ABSTRACT

Action recognition is a challenging task in the field of computer vision. The deficiency in training samples is a bottleneck problem in the current action recognition research. With the explosive growth of Internet data, some researchers try to use prior knowledge learned from various video sources to assist in recognizing the action video of the target domain, which is called knowledge adaptation. Based on this idea, we propose a novel framework for action recognition, called Semantic Adaptation based on the Vector of Locally Max Pooled deep learned Features (SA-VLMPF). The proposed framework consists of three parts: *Two-Stream Fusion Network* (TSFN), *Vector of Locally Max-Pooled deep learned Features* (VLMPF) and *Semantic Adaptation Model* (SAM). TSFN adopts a cascaded convolution fusion strategy to combine the convolutional features extracted from two-stream network. VLMPF retains the long-term information in videos and removes the irrelevant information by capturing multiple local features and extracting the features with the highest response to action category. SAM first maps the data of the auxiliary domain and the target domain into the high-level semantic representation through the deep network. Then the obtained high-level semantic representations from auxiliary domain are adapted into target domain in order to optimize the target classifier. Compared with the existing methods, the proposed methods can utilize the advantages of deep learning methods in obtaining the high-level semantic information to improve the performance of knowledge adaptation. At the same time, SA-VLMPF can make full use of the auxiliary data to make up for the insufficiency of training samples. Multiple experiments are conducted on several couples of datasets to validate the effectiveness of the proposed framework. The results show that the proposed SA-VLMPF outperforms the state-of-the-art knowledge adaptation methods.

1. Introduction

Recognizing human behavior in the video, namely action recognition, is a challenging task that draws much attention in the field of computer vision and pattern recognition. Previous works typically use classical hand-crafted features joint with the encoding methods such as improve dense trajectories [1] combining with improve Fisher Vector (iFV) [2] or Vector of Locally Aggregated Descriptors (VLAD) [3] to obtain effective semantic representation. With the rise of deep learning, a large amount of methods based on deep neural networks emerge such as [4–8]. These methods can automatically learn the high-level semantic information due to their high-order nonlinearity and data-driven characteristic. However, there is a problem with the deep learning that under the circumstance of insufficient training data, it often leads to overfitting problems due to the large amount of parameters. Besides, the characteristic of data-driven causes the model to be highly data-dependent. Thus, the model's generalization ability will be easily affected by occlusion, viewpoint variation and camera jittering. One solution is to obtain a large number of training samples using data augmentation tricks. However, it can only lead to very limited performance improvement. To effectively address the limitation, another approach is to exploit the knowledge learned from auxiliary data that can be easily acquired on Internet.

In Natural Language Processing (NLP) and Multimedia Event Detection (MED), some researchers propose to exploit auxiliary data to make up for the insufficiency of training data in the target domains. For instance, Jiang et al. [9] propose to use heterogeneous sources of knowledge for domain-adaptive video searching. Duan et al. [10] utilize a large amount of loosely-labeled web videos to process visual event recognition for consumer domain video. Many existing algorithms require that the features in both the auxiliary domains and target domains must be mapped to a common feature space. However, the requirement of feature consistency is too strict and it is hard to learn the mapping matrices. In this paper, we present a knowledge adaptation framework to solve the

* Corresponding author.
E-mail address: huhaif@mail.sysu.edu.cn (H. Hu).

https://doi.org/10.1016/j.patcog.2019.01.027
0031-3203/

problem. Our method can exploit knowledge from different types of features and adapt the learned knowledge from auxiliary domains to target domains without the need of feature consistency.

For feature extraction, two-stream network [11] is a most widely used deep network structure in the field of action recognition. It utilizes two different CNNs to process the video frames and the optical flow images respectively and then generates two kinds of features, namely spatial features and temporal features. To exploit the spatiotemporal cues in these features, previous works usually adopt element-wise sum and concatenation for feature fusion. These fusion methods are characterized by fewer parameters and convenience to implementation. However, they ignore to take the correlation between the different channels in features into consideration [12,13]. propose a strategy based on bilinear fusion, which can fully explore the correlation of different channels in features. Nevertheless, the dimensionality of the feature will increase significantly when performing bilinear fusion, thus causing serious over-fitting problems. Recently, Feichtenhofer et al. [14] proposes a convolution fusion strategy to explore the correlation of different channels for the spatial features and temporal features. The parameters of the fusion layer can be learned through the back propagation. Similar to their work, this paper proposes a novel Cascaded Convolution Fusion Strategy (CCFS) based on double-layer convolutional architecture. In our work, the input video is first divided into several temporal chunks. Then we obtain the spatial features and the temporal features using the two-stream network. The first layer fuses two types of features by convolutional operation and outputs the fusion features. The fusion features are next convolved by a bank of 3D filters with the size of $1 \times 1 \times 1$. Compared with the strategy proposed in [14], the cascaded convolutional fusion strategy need fewer parameters. In addition, the fusion features can obtain nonlinear gain by adopting the cascaded network structure.

Although the two-stream network achieves superior performance, there still exists two limitations in obtaining robust video representation. First, as stated in [15–18], long-term temporal structure plays an important role in understanding the dynamics in action video. However, CNN is only applicable to modelling short-term patterns of action, resulting in the difficulty to construct long-range video representation. Second, it is difficult for those methods to filter out the redundant visual information irrelevant to the action, which makes them sensitive to noise and weakens their generalization ability. In order to overcome the abovementioned limitations, this paper introduces a feature encoding method called *Vector of Locally Max-Pooled deep learned Features* (VLMPF) to capture the entire video information and eliminate the information irrelevant to action.

To sum up, in this paper, the main contributions are summarized as follows:

(1) This paper proposes a novel adaptation framework, called Semantic Adaptation based on Vector of Locally Max Pooled deep learn Features (SA-VLMPF), for action recognition. SA-VLMPF combines knowledge adaptation with deep learning methods. By using deep learning methods, the data of auxiliary domain and target domain can be mapped into the specfic high-level semantic space, providing rich semantic information for knowledge adaptation. At the same time, abundant auxiliary data can alleviate the over-fitting problem of deep learning methods caused by the insufficiency of training data. By constructing a joint analysis matrix, the semantic information can be adapted from the auxiliary domain to the target domain to optimize the target classifier. Moreover, we propose an iterative algorithm to optimize our adaptation model efficiently.

(2) CCFS is proposed to fuse the two-stream features effectively. It automatically learns the channel correlation of two types of features. Meanwhile, with fewer parameters, it can avoid the over-fitting problem and speed up the computation. Moreover, the obtained features can obtain higher-order non-linear fusion gain with the cascaded network architecture.

(3) In order to fully explore the long-term information in video, we introduce a feature encoding method VLMPF for long-range information representation. By applying the local max pooling to all features of the video, we can obtain a deep vector representation that has the highest response to the action category. The representation can not only capture the complete video information but also suppress the information irrelevant to the action.

2. Related works

2.1. Action recognition

In the early days of action recognition, researchers focused on the design of effective visual descriptors based on the hand-crafted local features with powerful encoding schemes [19,20] such as BOW [21], FV [2] and VLAD [3]. Commonly used hand-crafted local features are *Histograms of Gradient* (HOG) [22], *Histograms of Flow* (HOF) [23], *Space-Time Interest Point* (STIP) [24], *HOG3D* [25] and *3D-Sift* [26]. These features are used to describe the low-level visual information. Based on these hand-crafted local features, Peng et al. [20] provide a comprehensive study on BoVW and different fusion methods for action recognition. Wu et al. [19] improve the performance of action recognition by proposing a new pooling strategy for VLAD and several effective transformations for both FV and VLAD. However, there are large intra-class differences due to viewpoint change, occlusion and jitter in different videos. In these cases, the performance of the hand-crafted local features will be affected significantly. Instead of directly using the hand-crafted features, Liu et al. [27] automatically learn the spatiotemporal features for action via genetic programming. Liu et al. [28] propose a hierarchical clustering multi-task learning (HC-MTL) method for joint human action grouping and recognition. Ben et al. [29] develop a comprehensive suite of computational tools based on skeleton trajectories for 3D action classification. Yuan et al. [30] propose a novel framework for action recognition by combining the parameterized representation and discriminative classifier. The framework employs a novel probabilistic representation to obtain the key information of low-level features. Carmona et al. [31] improve the performance of improve dense trajectories [1] by adding new features based on the temporal templates. They construct the templates considering a video sequence as a third-order tensor and computing three different projections.

Recently, CNNs have made great progress in the field of computer vision. As a result, researchers have extended convolutional neural networks to video action recognition task. Simonyan et al. [11] creatively propose the two-stream architecture which adopts two mutually independent CNNs to process the RGB frames and their optical flow counterparts respectively. The great success of two-stream network arouses great interest of many researchers. Several recent works propose the variant of the two-stream architecture to fuse spatial and temporal features. For example, Donahue et al. [32] propose a two-stream structure combining CNN with RNN to extract appearance features and motion features, and integrate these features to obtain the final decision. Tran et al. [33] propose a novel 3D convolution network to fuse the spatiotemporal features. Tu et al. [34] propose a multi-stream CNN by additionally considering the human-related regions. Yang et al. [35] propose an efficient asymmetric 3D ConvNet to address the

limitation of traditional 3D ConvNet such as expensive computation and difficulty in network learning. These methods mostly adopt the classical fusion strategies such as *element-wise sum fusion, concatenation fusion* and *scores combination*. They are easy to implement, but do not take full advantage of the potential correlation of spatiotemporal cues. Although some researchers propose to explore the correlation of features by bilinear fusion [12,13], it may lead to over-fitting problem due to excessive parameters. To overcome the over-fitting problem and exploit the channel correlation between spatial features and temporal features, Feichtenhofer et al. [14] propose a convolutional fusion strategy. They insert a convolution fusion layer after the classical two-stream architecture such that the network can automatically learn the channel responses at the same pixel position. Therefore, the parameters of fusion layer can be learned by back propagation. Similar to their work, we propose CCFS to explore the correlation between spatial features and temporal features. Moreover, benefiting from the lightweight network structure, CCFS can avoid over-fitting problem.

Although CNN is successfully applied to action recognition, there is a limitation in video analysis, that is, the mainstream CNN frameworks usually focus on appearance information and short-term motion information, thus lacking the capacity to incorporate the long-term temporal structure. Recently, there are a few attempts to deal with this problem. For example, Donahue et al. [32], Varol et al. [36], Ng et al. [37] represent the video information by dense temporal sampling with a pre-defined sampling interval. These methods would lead to huge computational cost when applied to long videos. Zhang et al. [38] separate the video into several chunks by capturing switch shot using color histogram. Then they extract the features for each chunk and combine these features as video representation. This method can use a few local features to represent videos and thus reduce the computational cost. But it poses a risk of missing important information caused by the separation error. In this paper, we propose a feature encoding method, called VLMPF, which can integrate arbitrary number of features into a vector representation, thus enabling to learn the entire video information without the limit of video length. Moreover, VLMPF can capture features that has the highest response to action to produce a competitive video representation.

2.2. Knowledge adaptation

Due to the difficulty in action video collection and annotation, current public action datasets are limited in both size and diversity. To address the problem of data insufficiency, some scholars apply knowledge adaptation to the field of action recognition, that is, adapting the prior knowledge learned from the auxiliary dataset to the target domain. Knowledge adaptation can be illustrated through the following simple examples. As shown in Fig. 1, the first row shows three video frames of the action “archery” in the target domain. From the bounding boxes, we can see that “human”, “bow”, and “arrow” are the basic semantic components of archery. The images on the second row are taken from auxiliary sources, which also contains “human” “bow” and “arrow”. It implies that images containing these semantic components have a high probability belonging to the archery category. By exploring these shared semantic components and using knowledge adaptation, we can further improve the recognition performance.

Generally, knowledge adaptation methods can be divided into two categories.

The first category requires that the data in both target domains and auxiliary domains are represented in a common feature space. For example, Yang et al. [39] propose a Cross-Media Tag Transfer (CMTT) model that uses the same type of features to transfer tag knowledge from image data to video data. Bian et al. [40] propose a transfer topic model (TTM) which assumes that each action is the combination of a series of topics. They use the topic learned from the auxiliary domain to regulate the topic estimation in the target domain. Duan et al. [10] propose an Aligned Space-Time Pyramid Matching (ASTPM) to measure the distance between two video clips according to same type of features. These methods has a common limitation, that is, they require the features from different domain to be mapped to a common space. Hence, it is difficult to guarantee the satisfactory performance when dealing with heterogenous features.

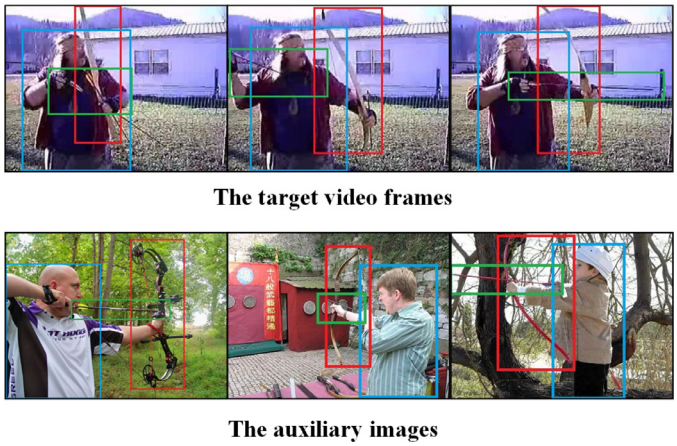

Fig. 1. The illustration of action “shooting an arrow” in auxiliary images and target video frames with the bounding boxes on the related semantic components.

The second category adapts knowledge in heterogeneous semantic feature space [38,41–44], thus it has become the main topic of the current knowledge adaptation field. For instance, Pereira et al. [43] present a simple and effective Semi-Supervised Transfer Subspace (SSTS) method for domain adaptation. SSTS establishs the pairwise constraints between the source and labeled target data. Wang et al. [44] propose a novel Transfer Fredholm Multiple Kernel Learning (TFMKL) framework for semi-supervised domain adaptation. TFMKL suppresses the noise for complex data distributions by developing a Fredholm integral based kernel prediction framework. Duan et al. [41] propose a Heterogeneous Features Adaptation (HFA) model for heterogeneous domain adaptation, which utilizes two feature projection matrices to map the heterogeneous features of two domains into a common feature space. Then the enhanced features are obtained by using two different projection functions. However, HFA is complicated and requires specific settings and adjustments due to the large amount of involved parameters. Recently, Zhang et al. [38] extend the structural adaptive regression [42] to action recognition tasks and achieve promising results. However, in the process of optimization, they utilize the *Forbenius Norm* to constrain the consistency of the mapping vectors obtained by two projection matrices. In such case, only the common components shared by two types of features can be mapped to corresponding action categories, resulting in loss of information. Different from Zhang et al. [38], we explore the knowledge of the appearance features with a linear projection matrix while the fusion features are learned by linear-SVM. In this way, heterogenous components of appearance features and fusion features can be fully retained and utilized for recognition. Meanwhile, projection matrices can be optimized using the knowledge learned from auxiliary domains through the structural adaptive regression methods.

3. Proposed method

The proposed SA-VLMPF framework is illustrated in Fig. 2. In our method, we first extract the appearance features for both the target domain and auxiliary domain, and extract spatiotemporal

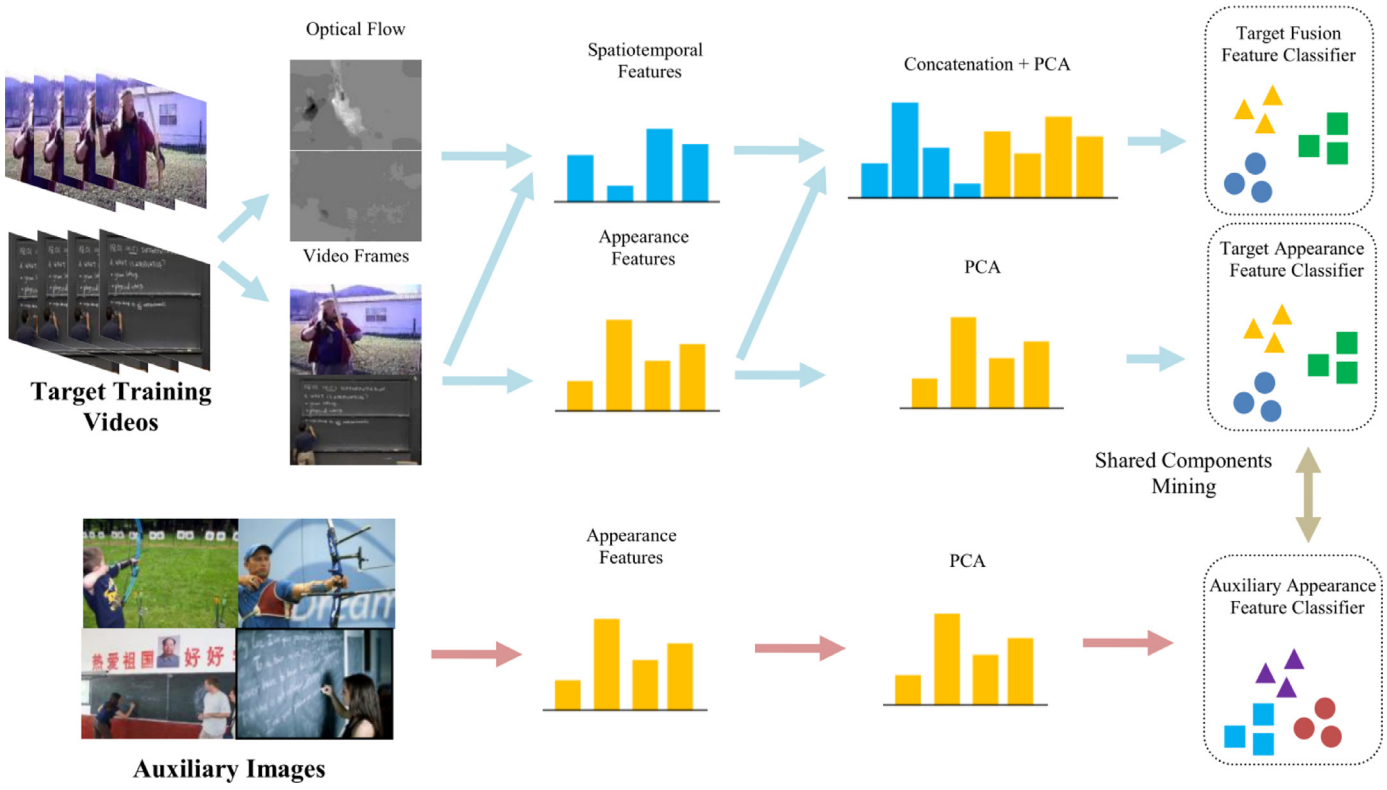

Fig. 2. The overview of the propose SA-VLMPF framework which leverages the deep semantic representation for knowledge adaptation.

features only for the target domain. The spatiotemporal features are obtained by TSFN using both optical flow images and the video frames. Then the proposed VLMPF encodes the extracted features for video representation. Based on the representation, SAM adapts the knowledge from the auxiliary domain to the target domain to help optimize the classifier.

In the following, we first introduce the extraction process of the deep feature representation. Then we present the knowledge adaptation model based on the extracted deep representation.

3.1. Deep feature representation

Although significant progress has been made in the convolutional neural network, there are mainly two limitations in the field of video analysis:

(1) Existing fusion strategies do not take the channel correlation between spatial features and temporal features into account;
(2) Limited by computation cost, CNN models can only process video sequences of fixed lengths, and thus are unable to learn the information from the entire video.

To overcome the aforementioned limitations, we propose CCFS for feature fusion and a feature encoding method called VLMPF for video representation.

3.1.1. Cascaded convolution fusion strategy

The process of the newly proposed CCFS is shown in Fig. 3. Concretely, we first extract the spatial feature maps $S_{map} = [s^1_{map}, \ldots, s^T_{map}]$ and temporal feature maps $T_{map} = [t^1_{map}, \ldots, t^T_{map}]$ from the last convolution layer of the TSFN, where T represents the number of chunks. s^i_{map} and t^i_{map} respectively denote the spatial feature maps and the temporal feature maps obtained from the i^{th} chunk. The temporal chunks are τ frames apart and each chunk contains a video frame and 10 continuous optical flow images. Then s^i_{map} and t^i_{map} are stacked across the feature channels and fed

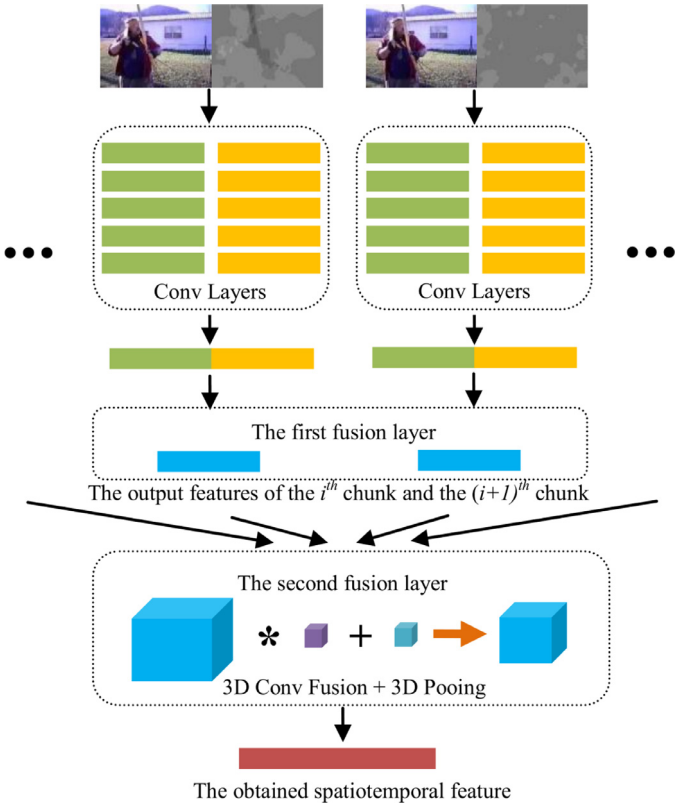

Fig. 3. The illustration of the cascaded convolution fusion strategy. The video is first processed by the two-stream network whose parameters are shared by all temporal chunks. The first fusion layer is used to learn the channel correlation of features. The second fusion layer learns the spatiotemporal cues and produces a robust spatiotemporal features.

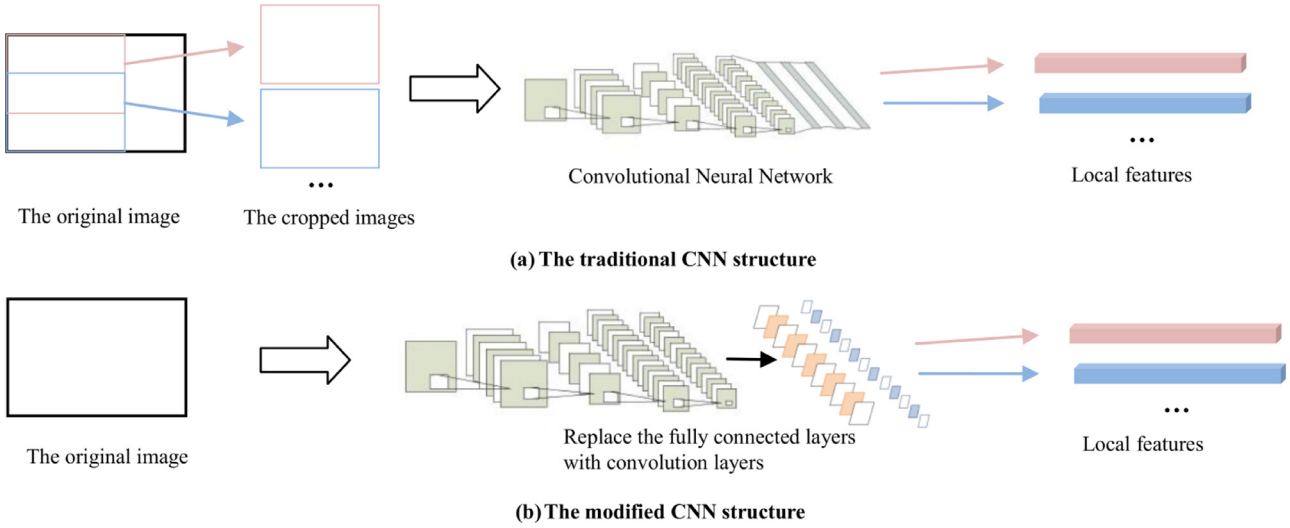

Fig. 4. Illustration of the original fully connected structure and the modified fully convolution structure.

into the first fusion layer. We use a bank of filters $F_1 \in R^{1\times 1\times 2D\times D}$ at the first fusion layer to convolve the stacked features and obtain,

$$st^i_{cf1} = [s^i_{map}, t^i_{map}] * F_1 + b_1, \tag{1}$$

where st^i_{cf1} is the output of the first convolutional fusion layer and b_1 is the bias term. We concatenate $\{st^i_{cf1}\}^T_{i=1}$ across the temporal dimension to obtain $ST_{cf1} = [st^1_{cf1}, \ldots, st^T_{cf1}] \in R^{H\times W\times T\times D}$. Considering the temporal structure information in ST_{cf1}, we adopt 3D convolution kernel to fuse the feature maps $\{st^i_{cf1}\}^T_{i=1}$ from different chunks. In fact, 3D convolution is the extension of 2D version with extra temporal dimension, which enables our model to integrate spatial and temporal information. Similarly, we use a bank of 3D filters $F_2 \in R^{H'\times W'\times T'\times D\times D'}$ at the second fusion layer to convolve ST_{cf1},

$$ST_{cf2} = ST_{cf1} * F_2 + b_2, \tag{2}$$

where ST_{cf2} is the output of the second fusion layer, b_2 is the bias term, D' is the channel number of ST_{cf2}. Here, we set $D' = D$ to adapt the structure of the original two-stream network, which allow ST_{cf2} to be directly fed into the pre-trained fully connected layers. Since the interval $\tau \in [1, 10]$ and the start chunk can be randomly selected, we can extract a series of features that cover the information of the entire video. It should be noted that in the experiments, we find that the parameter number of the proposed CCFS (27.5M) is just half as the convolution fusion strategy (54M) [14]. Besides, CCFS has high flexibility since it can be applied to arbitrary CNNs.

As shown in Fig. 4 (a), the traditional CNN adopts fully connected layer for the high-level feature extraction and we call it fully connected structure. In order to obtain multiple local features in images, we usually need to crop the images to obtain local regions and reshape them to fit the input size of CNN. However, it will lead to computation redundancy for feature extraction as the local features are spatially overlapped. Here, we introduce an alternative way to directly feed the original images into the CNN and obtain a set of local features. As shown in Fig. 4 (b), all the fully connected layers are replaced by convolution layers. We call it fully convolution structure. Since the fully convolution structure does not need to pull the feature map into a vector, it can perform local feature extraction on images with arbitrary size. In addition, since the obtained local features are spatially overlapped, CCFS has obvious advantages in the following two aspects: 1) it is easy to implement and can obtain multiple features in a short time; 2) the incremental redundancy of feature information will enhance the robustness of the features.

3.1.2. Vector of locally max-pooled deep learned features

The mainstream CNN-based models focus on modeling short-term information and lack the capability to represent long-term temporal structure. Therefore, we develop the feature encoding method, i.e. VLMPF to integrate multiple local features of videos for long-term temporal modeling. In our method, we first obtain the appearance features and spatiotemporal features with the spatial network and the TSFN respectively. Then features in two categories are combined together to obtain the fusion features:

$$U_j = Concat[V_j, X_j] = [u^1_j, u^2_j, \ldots, u^{d_u}_j], \quad j = 1, 2, \ldots, M, \tag{3}$$

where *Concat* denotes the concatenation operation, $X_j = [x^1_j, x^2_j, \ldots, x^{d_i}_j]$ denotes the appearance features, $V_j = [v^1_j, v^2_j, \ldots, v^{d_s}_j]$ denotes the spatiotemporal features, M denotes the number of videos and $d_u = d_i + d_s$. With the fusion features U_j and the appearance features X_j, we can obtain VLMPFs $U = [u^1, u^2, \ldots, u^{d_u}]$ and $X = [x^1, x^2, \ldots, x^{d_i}]$, where the elements x^k and u^k of the vectors U and X are formally computed as follows:

$$x^k = sign(x^k_j) \max_{j=1,2,\ldots M} |x^k_j|, \quad k = 1, 2, \ldots, d_i, \tag{4}$$

$$u^k = sign(u^k_j) \max_{j=1,2,\ldots M} |u^k_j|, \quad k = 1, 2, \ldots, d_u, \tag{5}$$

where the *sign* function returns the sign of the number and |.| denotes the absolute operation. By applying the local max pooling strategy to the local features in videos, VLMPF can capture the features with the highest response to the action categories. Besides, since VLMPF can integrate the information of arbitrary number of features, videos with different duration can be represented by the features with same dimension, which is convenient for the subsequent classification task.

3.2. Knowledge adaptation

In this section, we describe how to exploit the prior knowledge learned from auxiliary datasets to optimize the target classifier based on the deep semantic representation. In the following, we first introduce our *Semantic Adaptation Model* (SAM). Then, we

propose an alternative algorithm to solve the objective function of SAM. Finally, we discuss the convergence and the recognition strategy.

3.2.1. Notations

We use $X_a = [X_a^1, X_a^2, \ldots X_a^{n_a},] \in R^{d_i \times n_a}$ and $X_t = [X_t^1, X_t^2, \ldots, X_t^{n_t}] \in R^{d_i \times n_a}$ to represent the appearance feature vectors of auxiliary images and target videos respectively, where n_a denotes the number of images in auxiliary domain, n_t denotes the number of images in target domain and d_i is the dimension of vectors X_a and X_t. The fusion features are represented by $U = [U^1, U^2, \ldots, U^{n_t}] \in R^{d_u \times n_t}$, where d_u is the dimension of U. $Y_a = [y_a^1, y_a^2, \ldots, y_a^{n_t}]^T \in \{0,1\}^{n_a \times c_a}$ and $Y_t = [y_t^1, y_t^2, \ldots, y_t^{n_t}]^T \in \{n_t \times c_t\}$ are the label matrices for the auxiliary domains and target domains respectively. c_a and c_t denote the number of classes for auxiliary images and target videos. $y_a^k = [y_a^{k,1}, y_a^{k,2}, \ldots, y_a^{k,c_a}]$ and $y_t^l = [y_a^{l,1}, y_a^{l,2}, \ldots, y_a^{l,c_a}]$ are the labels of the k^{th} auxiliary image and l^{th} training video. $y_a^{k,j} = 1$ and $y_t^{l,j} = 1$ if X_a^k and X_t^l belongs to the j^{th} class, while $y_a^{k,j} = 0$ and $y_t^{l,j} = 0$ otherwise.

3.2.2. Semantic adaptation model

In this subsection, we introduce the SAM in our adaptation framework. We denote the obtained appearance features of auxiliary domain and target domain by X_a and X_t respectively. In order to exploit the semantic knowledge from two types of features, we first define two different functions f_t and f_a to map the features into the corresponding action categories. Specifically, f_t and f_a are defined as follows:

$$f_t(X_t) = X_t^T W_t + 1_t b_t, \tag{6}$$

$$f_a(X_a) = X_a^T W_a + 1_a b_a, \tag{7}$$

where $W_a \in R^{d_i \times c_a}$, $W_t \in R^{d_i \times c_t}$, $b_a \in R^{1 \times c_a}$ and $b_t \in R^{1 \times c_t}$ are the projection matrices and bias terms with respect to feature X_a and X_t. $1_t \in R^{n_t \times 1}$ and $1_a \in R^{n_a \times 1}$ denote the column vector with all ones. Since W_a and W_t are used to map the feature X_a and X_t to the corresponding action categories, we call W_a and W_t the auxiliary appearance feature classifier and the target appearance feature classifier, respectively. With label matrices Y_a and Y_t, f_t and f_a are obtained by minimizing the following objective function:

$$\min_{f_a, f_t} loss(f_t(X_t), Y_t) + loss(f_a(X_a), Y_a). \tag{8}$$

According to [45,46], $\ell_{2,1} - norm$ loss is more robust to outliers than hinge loss and least square loss. Therefore, by adopting the $\ell_{2,1} - norm$ loss, we can rewrite Eq. (8) as follows:

$$\begin{aligned} \min_{W_t, W_a, b_t, b_a} & \|X_t^T W_t + 1_t b_t - Y_t\|_{2,1} + \beta \|W_t\|_F^2 \\ & + \|X_t^T W_t + 1_t b_t - Y_a\|_{2,1} + \beta \|W_a\|_F^2, \end{aligned} \tag{9}$$

where $\|W_t\|_F^2$ and $\|W_a\|_F^2$ are the regularization terms, β is the regularization parameter.

Next, we introduce a joint analysis matrix to adapt semantic knowledge from auxiliary domain to target domain. Concretely, $W_t = [w_t^1, w_t^2, \ldots, w_t^{d_i}]^T$ and $W_a = [w_a^1, w_a^2, \ldots, w_a^{d_i}]^T$ are first combined to obtain a joint analysis matrix $W = [w^1, w^2, \ldots, w^{d_i}]^T$, where w^m is the vertical concatenation of w_a^m and w_t^m, i.e., $w^m = [w_a^m; w_t^m]$. Then, we propose $\|W\|_{2,p} = (\Sigma_{i=1}^{d_i} (\Sigma_{j=1}^{c_a + c_t} W_{i,j}^2)^{(p/2)})^{(1/p)}$ to constrain the correlation of semantic mapping, where $\|.\|_{2,p}$ denotes the $\ell_{2,p} - norm$ $(0 < p < 2)$. The objective function of the proposed SAM can be expressed as follows:

$$\begin{aligned} \min_{W_t, W_a, b_t, b_a} & \|X_t^T W_t + 1_t b_t - Y_t\|_{2,1} + \beta (\|W_a\|_F^2 + \|W_t\|_F^2) \\ & + \|X_t^T W_t + 1_t b_t - Y_a\|_{2,1} + \alpha \|W\|_{2,p}^p, \end{aligned} \tag{10}$$

where α is the regularization parameter.

The proposed SAM can be comprehended by the following two aspects.

Firstly, as described earlier, action can be described by the combination of several basic semantic components. Two actions are likely to belong to the same category if they share the similar semantic components. By applying deep neural network, the data from auxiliary domains and target domains can be mapped into the high-level semantic feature space. Hence, our work can obtain richer semantic representation, which is beneficial to the process of exploring the common semantic components shared by two domains. According to [42], if there are common semantic components shared by the auxiliary domains and the target domains, we should find the similar distribution pattern in the projection matrices W_a and W_t. However, some noise and the irrelevant semantic components such as the scene and the posture of human, are inevitably incorporated in the projection matrices W_a and W_t. By using semantic adaptation, shared semantic components in two domains can be explored for recognition and the irrelevant noise can be suppressed.

Secondly, previous works [47,48] have shown that sparse models are useful for eliminating redundancy and noise. Hence, we adopt the $\ell_{2,p} - norm$ to achieve that goal. By minimizing $\|W\|_{2,p}$, the corresponding rows of projection matrix W_a or W_t for the irrelevant noisy features will gradually shrink to zero. In this way, SAM can suppress the irrelevant noise and thus explore the common semantic components for action recognition. Besides, SAM has the flexibility to control its sparse degree according to the degree of correlation between the auxiliary domains and target domains. In general, a large p will be used in the case that two domains have high correlation. When p increases to 2, $\|W\|_{2,p}^p$ becomes $\|W\|_F$, where $\|.\|_F$ denotes the *Frobenius Norm*. In the case of low correlation of two domains, decreasing p can eliminate the useless semantic components. When the p is close to zero, we consider there is no sharing component between two domains.

In the next subsection, we will explain how to solve the optimization problem of Eq. (10). Since there are multiple variables to be optimized in the Eq. (10), we provide an alternative algorithm for efficient optimization.

3.2.3. Optimization algorithm for SAM

To solve the objective function, we first denote two intermediate matrices:

$$G_t = X_t^T W_t + 1_t b_t - Y_t = [g_t^1, g_t^2, \ldots, g_t^{n_t}]^T, \tag{11}$$

$$G_a = X_a^T W_a + 1_a b_a - Y_a = [g_a^1, g_a^2, \ldots, g_a^{n_a}]^T, \tag{12}$$

Then we define three diagonal matrices D, D_t and D_a with corresponding diagonal elements, $D^{k,k} = \|w^k\|_2^{p-2}$, $D_t^{k,k} = \|g_t^k\|_2^{-1}$ and $D_a^{k,k} = \|g_a^k\|_2^{-1}$ respectively. It should be noted that the above three diagonal matrices are treated as constants in the optimization process as we adopt an alternative approach. With the diagonal matrices, we can rewrite the objective function as follows:

$$\begin{aligned} \min_{W_t, W_a, b_t, b_a} & Tr((X_t^T W_t + 1_t b_t - Y_t)^T D_t (X_t^T W_t + 1_t b_t - Y_t)) \\ & + Tr((X_a^T W_a + 1_a b_a - Y_a)^T D_a (X_a^T W_a + 1_a b_a - Y_a)) \\ & + \alpha Tr(W^T D W) + \beta (Tr(W_t^T W_t) + Tr(W_a^T W_a), \end{aligned} \tag{13}$$

where $Tr(.)$ denotes the trace operator. We find that Eq. 13 becomes convex with respect to b_t and b_a by fixing the other variables. Hence, by setting the derivative of Eq. (13) with respect to b_t and b_a to zero respectively, we can obtain:

$$b_a = \frac{1}{n_a} 1_a^T Y_a - \frac{1}{n_a} 1_a^T X_a^T W_a, \tag{14}$$

$$b_t = \frac{1}{n_t}1_t^T Y_t - \frac{1}{n_t}1_t^T X_t^T W_t, \tag{15}$$

where $I_t \in R^{n_t \times n_t}$ and $I_a \in R^{n_a \times n_a}$ are two identity matrices. By substituting b_a and b_t into Eq. (13), the objective function becomes:

$$\begin{aligned} \min_{W_t, W_a, b_t, b_a} & Tr((H_t X_t^T W_t - H_t Y_t)^T D_t (H_t X_t^T W_t - H_t Y_t)) \\ & + Tr((H_a X_a^T W_a - H_a Y_a)^T D_a (H_a X_a^T W_a - H_a Y_a)) \\ & + \alpha Tr(W^T D W) + \beta (Tr(W_t^T W_t) + Tr(W_a^T W_a), \end{aligned} \tag{16}$$

where $H_t = I_t - \frac{1}{n_t}1_t 1_t^T$, $H_a = I_a - \frac{1}{n_a}1_a 1_a^T$, $I_a \in R^{n_a \times n_a}$ and $I_t \in R^{n_t \times n_t}$. Then we set the derivative of Eq. (16) with respect to W_a and W_t to zero and get:

$$W_a = (X_a H_a D_a H_a^T X_a^T + \beta I_{d_i} + \alpha D)^{-1} X_a H_a D_a H_a Y_a, \tag{17}$$

$$W_t = (X_t H_t D_t H_t^T X_t^T + \beta I_{d_i} + \alpha D)^{-1} X_t H_t D_t H_t Y_t, \tag{18}$$

where $I_{d_i} \in R^{d_i \times d_i}$ is an identity matrices. The optimization of our adaptation objective function is introduced in Algorithm 1.

Algorithm 1 Optimizing the proposed SAM.

Input: The target training data $X_t \in R^{d_i \times n_t}$, $Y_t \in R^{n_t \times c_t}$; The auxiliary data $X_a \in R^{d_i \times n_a}$, $Y_a \in R^{n_a \times c_a}$;Hyper-parameters α, β and p.
Output: Optimized $W_t \in R^{d_i \times c_t}$ and $b_t \in R^{1 \times c_t}$

1: Set $t = 0$, initialize $W_t \in R^{d_i \times c_t}$ and $W_a \in R^{d_i \times c_a}$ randomly;
2: Compute H_a and H_i according to $H_t = I_t - \frac{1}{n_t}1_t 1_t^T$ and $H_a = I_a - \frac{1}{n_a}1_a 1_a^T$
3: **while** The loss in Eq. (10) does not converge **do**
4: Compute $G_t = X_t^T W_t + 1_t b_t - Y_t = [g_t^1, g_t^2, \ldots, g_t^{n_t}]^T$, $G_a = X_a^T W_a + 1_a b_a - Y_a = [g_a^1, g_a^2, \ldots, g_a^{n_a}]^T$ and $W = [w^1, w^2, \ldots, w^{d_i}]^T$.
5: Compute the diagonal matrices D_t, D_a and D according to $D^{k,k} = \|w^k\|_2^{p-2}$, $D_t^{k,k} = \|g_t^k\|_2^{-1}$ and $D_a^{k,k} = \|g_a^k\|_2^{-1}$ respectively;
6: Compute W_a according to Eq. (17);
7: Compute b_a according to Eq. (14);
8: Compute W_t according to Eq. (18);
9: Compute b_t according to Eq. (15);
10: Compute the loss of Eq. (10).
11: **end while**
12: **return** W_t and b_t

3.2.4. Convergence of the algorithm

In this subsection, we discuss the convergence of the algorithm. To obtain optimal b_t, we fix the other parameters to make Eq. (10) convex. In this way, the loss value will decrease by solving the optimal b_t. Similarly, we iteratively solve other parameters. Since Eq. (10) is convex in each iteration, the loss value will monotonically decrease when optimizing our SA-VLMPF, which can ensure the convergence.

3.2.5. Recognition strategy

We use the optimized W_t and b_t to classify the appearance features X_a and use Linear-SVM to classify the fusion features, obtaining two types of score vectors, $S^x = [s_1^x, s_2^x, \ldots, s_{c_t}^x]$ and $S^u = [s_1^u, s_2^u, \ldots, s_{c_t}^u]$. Considering the scale difference between two types of vectors, we transform the vectors into the same scale by *Min-Max Normalization* (*MMN*). The transformation function is defined as follows:

$$s_i = \frac{s_i - s_{min}}{s_{max} - s_{min}}, \tag{19}$$

where s_i is the i^{th} value of score vector, $s_{\min}$ and $s_{\max}$ are the minimum and maximum value of score vector S. Here, we drop the superscript of S^x and S^u for convenience. The final prediction of k^{th} testing video is decided by $y^k = \arg\max_j (s_j^x + s_j^u)$, where s_j^x and s_j^u are the j^{th} element of score vectors of appearance feature and fusion feature respectively.

3.3. Discussion

In this section, we will discuss the architecture construction of our SA-VLMPF. Many existing works perform domain adaptation in an end-to-end way based on deep neural network as they can automatically learn the feature from the raw data without the need of manual pre-processing. Different from these works, the construction of our SA-VLMPF is based on the following two considerations. Firstly, the proposed SA-VLMPF framework consists of three key components, i.e., TSFN for feature extraction, VLMPF for feature encoding and SAM for domain adaption. They are independently constructed and and it is hard to integrate them together to build an end-to-end architecture. It should be noted that the separated architecture can enhance the flexibility and applicability in real-world applications as it can adopt better feature extractor for improvement. Secondly, a key problem in action recognition task is the shortage of training data, which may easily induce the over-fitting problem for the classical end-to-end training strategy [49]. In our work, TSFN is independently designed as the feature extractor which is pre-trained on the large scale datasets. In this way, the learned network structure with strong generalization capability may be utilized to alleviate the over-fitting problem.

4. Experiments

4.1. Datasets of auxiliary domains and target domains

To verify the effectiveness of our SA-VLMPF framework, we construct three couples of auxiliary domains and target domains. As mentioned before, the auxiliary data should contain semantically related components for representation and adaptation. To this end, [38] obtains the auxiliary data by selecting the images that have similar labels to the target action data. In our work, we adopt the similar strategy for obtaining auxiliary data from the given datasets. For example, “shooting an arrow” in Stanford40 and “archery” in UCF101 have similar action label, which means that they are highly semantically correlated. Therefore, when performing action classification on the UCF101, it may be beneficial to adaptation by using the action data 'shooting an arrow' to construct the auxiliary dataset. Some of the auxiliary images are shown in Fig. (5). The datasets used in our experiments are as follows:

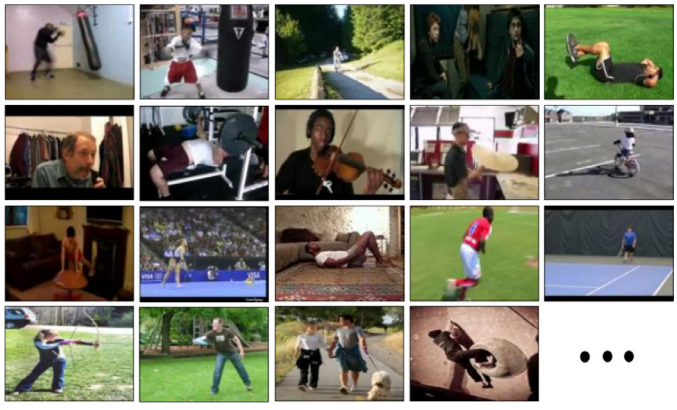

Fig. 5. Samples images from auxiliary datasets used in the experiments.

(1) From Stanford40 [50] to UCF101 [51]: The UCF101 contains 101 action classes and there are at least 100 videos for each class. The dataset contains 13,320 videos which are divided into 25 groups for each action category. The Stanford40 contains images including 40 action categories. There are 9532 images in total with 180–300 images per action category. Table 1 shows the selected categories of Stanford40 → UCF101, where the token " → " denotes the direction of adaptation from auxiliary domain to target domain. Specifically, we select 10 related categories from Stanford40 and UCF101 as auxiliary images and target videos respectively. To ensure effective adaptation, the selected action pairs should have similar labels. In the construction of the following two couple of datasets, i.e., EAD → UCF101 and Stanford40 → HMDB51, we adopt the same strategy for selecting the auxiliary data for semantic adaptation.

(2) From Expanded Action Datasets (EAD) to UCF101: To evaluate the ability of our SA-VLMPF to adapt knowledge from different sources of auxiliary images, we obtain auxiliary data from EAD which includes three sources of data (i.e. People Playing Musical Instrument (PPMI) [52], Willow-Actions (WA) [53] and Images collected from Google (IFG)). Similarly, we select another 8 overlapping classes from both UCF101 and EAD and the selected categories of EAD → UCF101 are shown in Table 2.

(3) From Stanford40 to HMDB51 [54]: The HMDB51 consists of 6766 video clips including 51 action categories. There are at least 100 videos for each action category. HMDB51 is a challenging dataset for action recognition due to its complicated scenes. We select 9 related categories between Stanford40 and HMDB51. Table 3 shows the selected categories of Stanford40 → HMDB51.

4.2. Experimental setup

The TSFN is built upon the VGG-M-2048 network [55] which is pre-trained on ImageNet [56]. For the CCFS illustrated in Fig. 3, the kernel size of the filters F_1 and F_2 are $1 \times 1 \times 1024 \times 512$ and $3 \times 3 \times 3 \times 512 \times 512$ respectively. We first fine-tune the parameters of TSFN on the UCF101. In the training process, we fix the parameters of the convolution layer and only train the parameter of the fully connected layers and the fusion layers. To verify the generalization ability of the proposed SA-VLMPF framework, we do not fine-tune the parameters of TSFN on the HMDB51.

Table 1
Chosen related categories of Stanford40 → UCF101.

Dataset	Stanford40	UCF101
The related categories	brushing teeth	brushing teeth
	clearing the floor	mopping floor
	climbing	rock climbing indoor
	cutting vegetables	cutting in kitchen
	playing guitar	playing guitar
	rowing a boat	rowing
	shooting an arrow	archery
	throwing frisbee	frisbee catch
	walking with dog	walking with dog
	writing on a board	writing on board

Table 2
Chosen related categories of EAD → UCF101

Dataset	EAD	UCF101
The related categories	riding bike (WA)	biking
	riding horse (WA)	horse riding
	playing cello (PPMI)	playing cello
	playing flute (PPMI)	playing flute
	playing violin (PPMI)	playing violin
	blow dry hair (IFG)	blow dry hair
	ice dancing (IFG)	ice dancing
	tennis swing (IFG)	tennis swing

Table 3
Chosen related categories of Stanford40 → HMDB51.

Dataset	Stanford40	HMDB51
The related categories	applauding	clap
	drinking	drink
	jumping	jump
	pouring liquid	pour
	pushing a cart	push
	running	run
	smoking	smoke
	waving hands	wave
	climbing	climb

4.3. Algorithms for comparison

To validate the performance of our SA-VLMPF framework, we adopt the following seven state-of-the-art knowledge adaptation algorithms for comparison.

(1) Heterogeneous Feature based Structural Adaptive Regression (HFSAR) [42]: A knowledge transfer framework based on semantic information. Semantic information is obtained by encoding the SIFT features [57] and STIP features [24] of image using the standard BoW method.

(2) Image-to-Video Adaptation (IVA) [38]: A semi-supervised knowledge adaptation framework. Similar to HFSAR, semantic information is obtained by encoding the SIFT feature and STIP feature of image using the standard BoW method. Besides, both the labeled and unlabeled data are used in the IVA framework.

(3) Heterogeneous Feature Adaptation (HFA) [41]: A supervised knowledge adaptation methods based on SVM. To obtain the enhanced feature, HFA maps the target feature and the auxiliary feature to the same feature space through two different projection functions.

(4) Multiple kernel transfer learning (MKTL) [58]: A multi-class transfer learning framework built upon multiple kernel learning method. The RBF kernel is adopted in MKTL.

(5) Image Attribute Adaptation (IAA) [59]: A semi-supervised knowledge adaptation framework built upon the multiple kernel learning methods.

(6) Deep Convolutional Activation Feature (DeCAF) [60]: An end-to-end deep learning framework for domain adaptation. DeCAF performs knowledge adaptation by directly fine-tuning the deep CNN pre-trained on the large-scale dataset.

(7) Domain Adaptation Network (DAN) [61]: An end-to-end deep learning framework for domain adaptation. It introduces maximum mean discrepancy term to construct a maximum mean discrepancy loss for domain invariant representation learning.

4.4. Adaptation results of different datasets

In this subsection, we compare the proposed SA-VLMPF framework with the Heterogeneous Feature Structure Adaptive Regressive model (HFSAR) proposed in [38,42], which is one of the most effective knowledge adaptation methods for action recognition. Similar to [38], we use the results obtained by VLMPF+linear-SVM as the baseline. For fair comparison, all the methods use same deep learned representation (i.e. the proposed VLMPF) for training and testing. We use HFSAR-D to denote the HFSAR model using VLMPF as feature representation. Since VLMPF+linear-SVM does

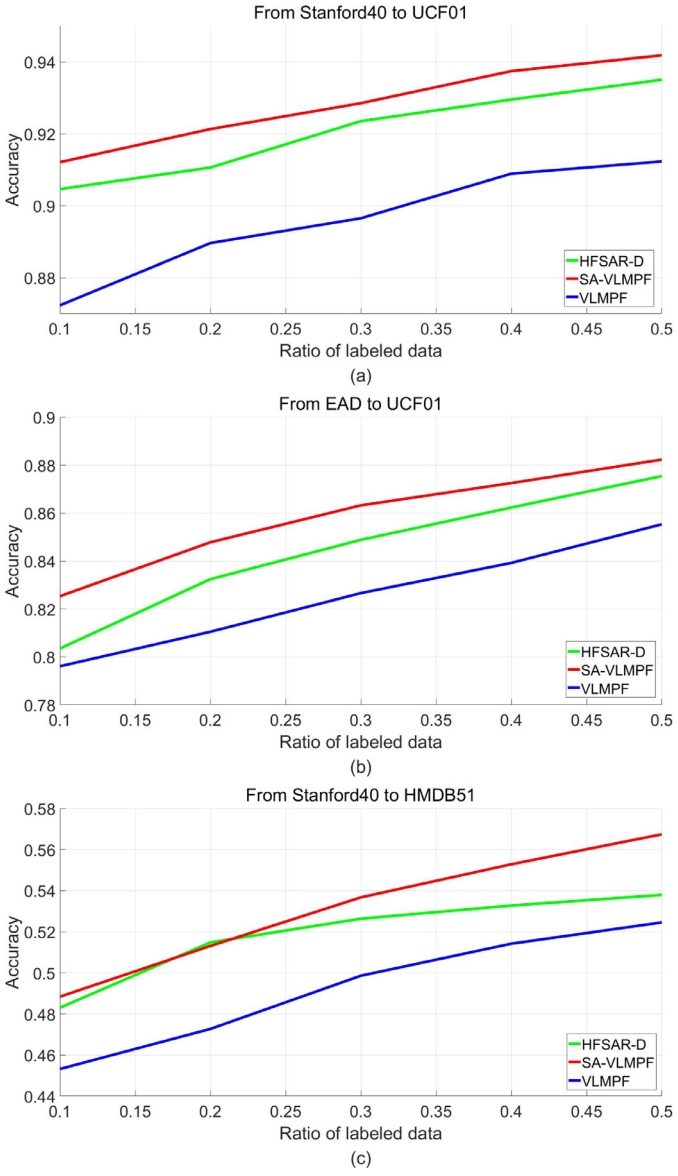

Fig. 6. Recognition performance on three couples of auxiliary datasets and target datasets.

not involve semantic adaptation, we denote it as VLMPF for convenience. For SA-VLMPF and HFSAR-D, we use the same hyper-parameter settings, i.e, α, β and p is set to 1000, 10 and 1.2 for Stanford40 → UCF101 and EAD → UCF101, and 1000, 1000, 1.2 for Stanford40 → HMDB51 respectively. For all the models, we adopt the same linear-SVM as classifier and the penalty coefficient C is set to 100. In the experiments, labelled data are divided into two parts, one for training and the other for testing. To further investigate the influence of the amount of labelled data on the performance, we set the ratio of training data to 10%, 20%, 30%, 40% and 50%. The remaining 50% of labelled data are used for testing. The experimental results are shown in (Fig. 6) with the following observations:

(1) Our SA-VLMPF achieve the best performance in all the experiments, which proves the better knowledge adaptation ability of SA-VLMPF.
(2) We notice that both HFSAR-D and SA-VLMPF outperform the VLMPF, which demonstrates that the prior knowledge learned from the auxiliary images can effectively improve the performance.
(3) EAD is a combination of various sources of auxiliary images. From Fig. 6 (b) we can find that SA-VLMPF is consistently superior to the other two methods, which demonstrates that SA-VLMPF can better utilize the semantic information in various sources of data to adapt knowledge.
(4) From Fig. 6 (c), we observe that the results of HFSAR-D is close to that of SA-VLMPF when using small amount of labelled data, e.g., the labelled data ratio is 10% or 20%. As the ratio of labelled data increases, SA-VLMPF consistently outperforms HFSAR-D. Besides, SA-VLMPF shows the improvement ranging from 3% to 4% compared with VLMPF, which demonstrates the generalization ability of SA-VLMPF to various data.
(5) When the labelled ratio is 10%, SA-VLMPF gains improvement over VLMPF by 4.2%, 0.9% and 3.1% on Stanford40 → UCF101, EAD → UCF101 and Stanford40 → HMDB51 respectively. The success of SA-VLMPF lies that it can effectively adapt knowledge from auxiliary domains to target domains thus making up for the insufficiency of labelled data. This advantage is particularly beneficial to real-world application due to the scarcity of labelled videos.
(6) In the figure, the recognition curves obtained by VLMPF and SA-VLMPF show the same trend, that is, when using more labeled training data, the performance of both methods gets consistently improved. In our work, VLMPF can be regarded as the counterpart of SA-VLMPF without using the semantic knowledge learned from auxiliary domains. Therefore, they may learn the similar discriminative information and have similar properties.

4.5. Performance with different hyper-parameters settings

There are three hyper-parameters involved in the proposed framework, i.e., α, β and p. In the following, we conduct three experiments to further investigate how these parameters affect the performance of the proposed SA-VLMPF. The results are shown in Figs. 7–9, respectively.

Fig. 7 shows the recognition results for different α and β when fixing the parameter p. From Fig. 7 (a), we notice that β has little effect on the recognition performance on the experiment of Stanford40 → UCF101, whereas the recognition performances are very sensitive to both α and β on EAD → UCF101 and Stanford40 → HMDB51. Especially, we can see from Fig. 7 (c) that larger α gives better performance on the experiment of Stanford 40 → HMDB51. One possible reason is that our TSFN is not fine-tuned on HMDB51. By enlarging α, the effect of $\|W\|_{2,p}^{p}$ can be enhanced, thereby strengthening the generalization ability of the model to adapt the data in HMDB51.

Fig. 8 shows the recognition results for different β and p when fixing parameter α. We find that the performance shown in (a) and (b) are not sensitive to β while the performance shown in (c) increases as β and p increase. Besides, the best results for Stanford40 → UCF101 and EAD → UCF101 are obtained when $p = 1.2$, whereas the best result for Stanford40 → HMDB51 is obtained when $p = .8$. This indicates that the correlation between Stanford40 and HMDB51 is low.

Finally, we fix β to investigate the effect of α and p. The results are shown in Fig. 9. From Fig. 9 (a) and (b), we are surprised to find that the performance of (a) and (b) decreases significantly when $p = .6$ and $\alpha = 1000$. Since $\alpha\|W\|_{2,p}^{p}$ controls the effect of knowledge adaptation, we argue that the smaller p and the larger α make the parameters of W_t shrink to small values during the optimization process. In this case, some useful features cannot be mapped to score vector because the values of corresponding row of

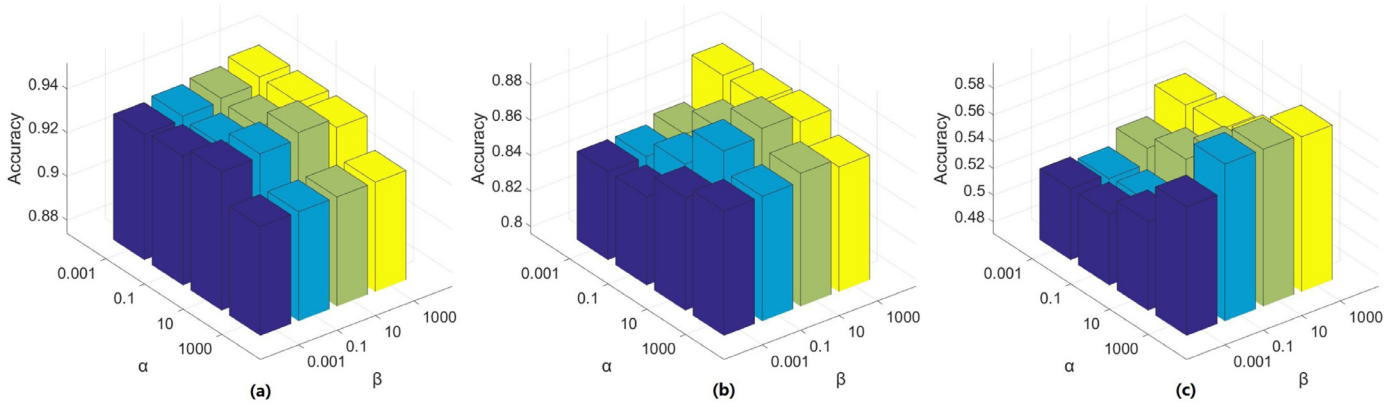

Fig. 7. Recognition performance variance with respect to α and β. (a) Stanford40 → UCF101; (b) EAD → UCF101; (c) Stanford40 → HMDB51.

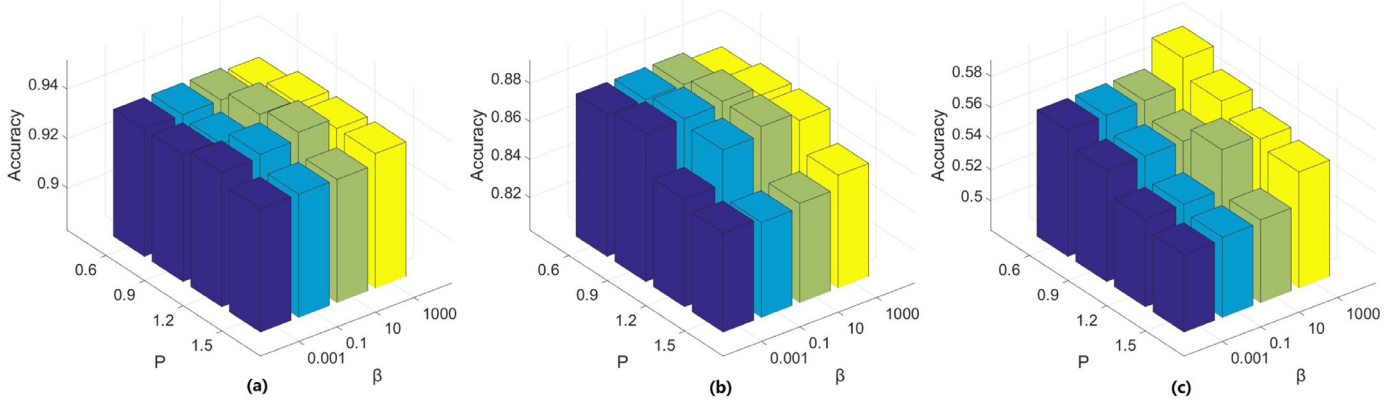

Fig. 8. Recognition performance variance with respect to p and β. (a) Stanford40 → UCF101; (b) EAD → UCF101; (c) Stanford40 → HMDB51.

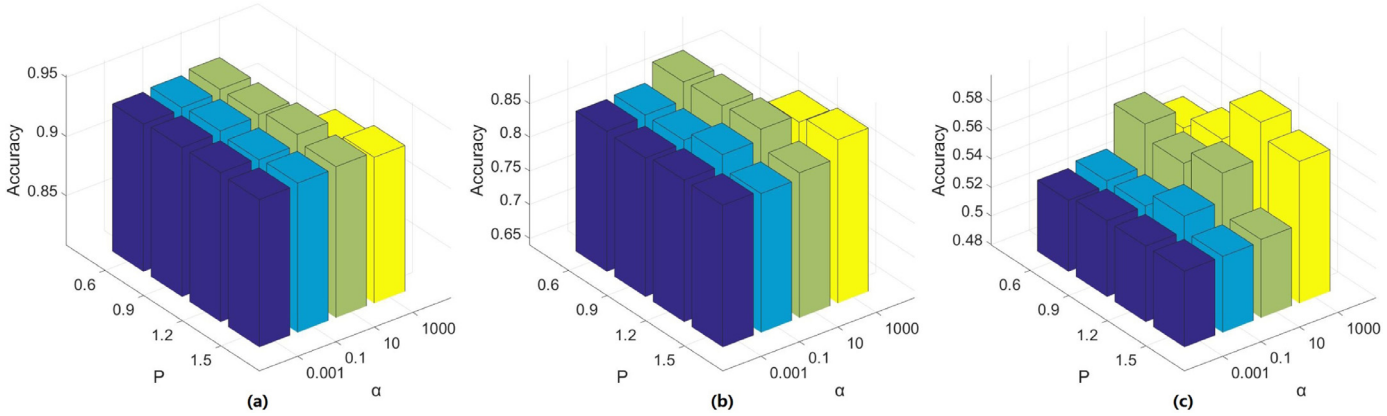

Fig. 9. Recognition performance variance with respect to p and α. (a) Stanford40 → UCF101; (b) EAD → UCF101; (c) Stanford40 → HMDB51.

W_t is close to zero, resulting in information loss and performance degradation.

4.6. Performance comparison of different fusion methods

In this section, we compare the proposed CCFS with several classical fusion strategies on three datasets. The results are shown in Table 4. *Feature Sum* denotes that the features obtained from the spatial network and temporal network are added together. *Concatenation* denotes that the features of spatial networks and temporal networks are concatenated along the channel dimension and *Score average* denotes that the final result is the averaged scores of two networks. From the results we can see that CCFS has the best performance among all the listed fusion methods. The success of CCFS lies on the fact that it can explore the correlation between spatial features and temporal features. Besides, since the videos in HMDB51 are untrimmed and have a variety of complicated scenes, the classical fusion strategy cannot obtain satisfactory performance. CCFS has 5% ∼ 10% performance improvement compared with these methods, which demonstrates that the proposed strategy has more discriminative power to deal with various kinds of background clutters.

4.7. Effect of semantic adaptation

In order to evaluate the effect of our proposed framework on the adaptation performance of each action category, we compare SA-VLMPF with its counterpart, i.e. VLMPF that doesn't apply se-

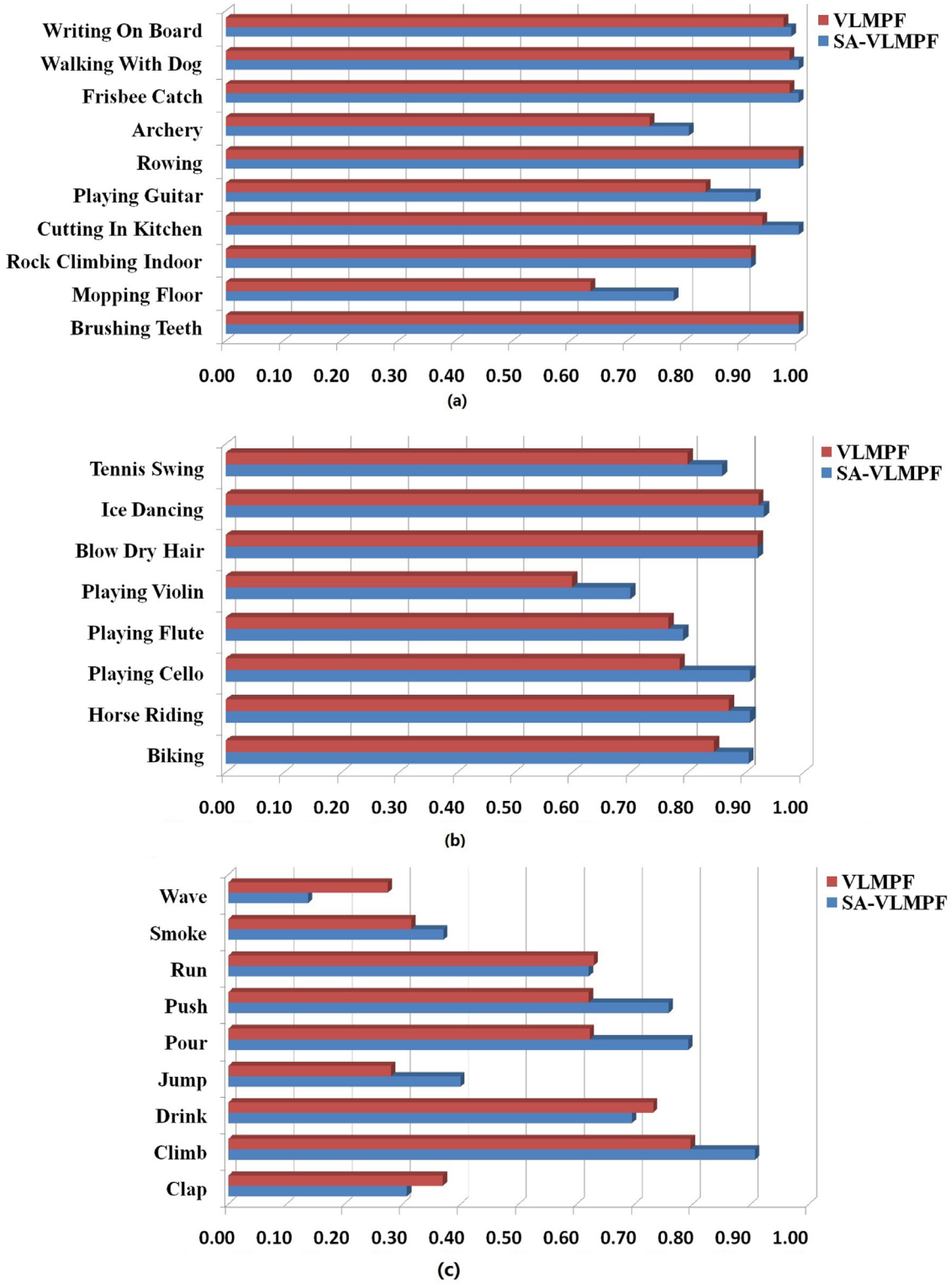

Fig. 10. Performances comparison between SA-VLMPF (with semantic adaptation) and VLMPF (without semantic adaptation) for each action category. (a) Stanford40 → UCF101; (b) EAD → UCF101; (c) Stanford40 → HMDB51.

Table 4
Comparison the proposed CCFS with the classical fusion methods.

Method	Concatenation	Feature Sum	Scores Average	Ours
Stanford40 → UCF101	88.75	90.46	89.65	**91.23**
EAD → UCF101	84.32	83.96	83.13	**85.53**
Stanford40 → HMDB51	42.10	46.22	47.03	**52.45**

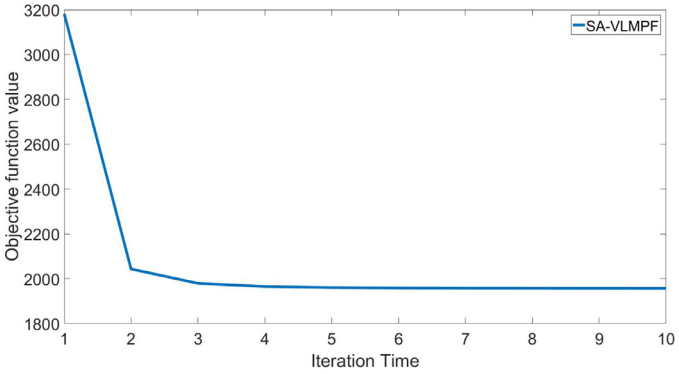

Fig. 11. Convergence curve of the objective function in Eq. 10.

mantic adaptation. In the following, we show the accuracy of each category in detail, and analyze the performance differences between SA-VLMPF and VLMPF. The experimental results are shown in Fig. 10.

From Fig. 10 (a) and (b), we can clearly see that the recognition accuracy of all the action categories have been improved with our SAM, which reflects that SA-VLMPF can utilize the prior semantic knowledge learned from auxiliary domains to improve the performance of action recognition. In addition, the action categories with lower recognition accuracy may gain more improvement from semantic adaptation, e.g., "Mopping Floor" in (a) and "Playing Violin" in (b). We argue that the action can not be well recognized when there are large differences between training data and testing data. By adopting semantic adaptation, the learned semantic knowledge can be utilized to make up for the problem mentioned above, thus improving the performance.

From the results shown in Fig. 10 (c), we can observe that most of action categories are beneficial from semantic adaptation whereas the accuracies of some action categories are unexpectedly dropped such as "wave" and "clap" in (c). It may be caused by the low correlation of these actions in auxiliary domain and target domain. This phenomenon indicates that semantic adaptation still requires a certain degree of correlation between auxiliary domains and target domains.

4.8. Convergence of the proposed algorithm

In this subsection, we experimentally validate the convergence of the proposed algorithm. Fig. 11 plots the value of objective function on Stanford40 → UCF101. It can be seen from the figure that the value of the objective function converges very fast, and is almost close to the local optimum within 10 iterations. Similar results can be found on EAD → UCF101 and Stanford40 → HMDB51. The result further demonstrates the convergence and efficiency of the proposed algorithm.

4.9. Performance comparison with the state-of-the-art methods

In this subsection, we compare the proposed SA-VLMPF framework with seven state-of-the-art knowledge adaptation methods including five traditional methods (HFSAR, IVA, HFA, MKTL and IAA) and two adaptation models based on deep learning (DeCAF and DAN). It is should be noted that we implement the experiment on Stanford40 → UCF101 by following the same settings in [38]. For DeCAF, DDC and DAN, we only perform knowledge adaptation for spatial network since there is no optical flow data in auxiliary domain.

The comparison results with traditional methods are shown in Table 5. From the table, we can clearly see that SA-VLMPF has obvious advantage over the other methods. The superior performance

Table 5
Comparison with traditional knowledge adaptation methods on Stanford40 → UCF101.

Method	MKTL	HFA	HFSAR	IAA	IVA	HFSAR-D	Ours
Accuracy	78.86	82.14	81.82	81.35	84.53	92.16	**92.95**

Table 6
Comparison with knowledge adaptation methods based on deep learning on Stanford40 → UCF101.

Method	DeCAF	DAN	Ours
Accuracy	90.23	91.03	**92.95**

may be explained by the following two aspects. First, we design a video representation method (i.e. VLMPF) to extract the high-level semantic information and incorporate the long-term information in videos. Second, SA-VLMPF can better adapt knowledge from auxiliary data based on the high-level semantic information. Besides, the performance of SA-VLMPF is better than that of HFSAR-D though both methods adopt VLMPF for video representation, indicating that SA-VLMPF can fully utilize the semantic components in heterogenous features for action recognition.

In addition, we compare our SA-VLMPF with two end-to-end deep learning methods and show the results in Table 6. We observe that our SA-VLMPF performs best among the listed methods. The improvement may be due to the deep learned representation obtained by the proposed CCFS which can fully capture the spatiotemporal cues and deep semantic information in videos, and thus our SAM can better exploit the semantic components shared in both domains and achieve better performance. In addition, we can see by comparing Tables 5 and 6 that the models using deep learned feature outperforms traditional methods, which proves that deep learning methods have stronger generalization capability to other data. We believe it is due to the fact that deep learning methods can effectively extract the discriminative semantic components for adaptation. This also shows that the acquisition and utilization of semantic information can effectively improve adaptation performance.

5. Conclusions

In this paper, we propose a novel framework called SA-VLMPF for action recognition. which can leverage the semantic knowledge learned from auxiliary domains to improve the performance in target domains. In our work, a CCFS is proposed to explore the correlation between the spatial stream and the temporal stream. Then we introduce the VLMPF to model the long-range temporal information in videos. Finally, we extract the VLMPF from both the auxiliary domain and the target domain to obtain the semantic representation, and employ the SAM to help optimize the target classifier. In order to verify the effectiveness of our framework, multiple experiments are conducted on the several couples of public datasets. In the experiments, we show the benefit of exploiting semantic knowledge from the auxiliary data. Compared with the current state-of-the-art methods, our framework is able to yield superior performance.

Although our SA-VLMPF boosts the accuracy of action recognition, there still exists a limitation in our work, that is, the separated architecture may increase the model's complexity. In the future, we plan to further improve the performance through the following two ways: (1) We will try to reconstruct the semantic adaptation model with deep network structure and make it be trained in an end-to-end way. (2) Considering the fact that the performance can be improved by increasing the depth of network, we will adopt the deeper architecture such as ResNets and DenseNets for discriminative feature extraction.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61673402, Grant 61273270, and Grant 60802069, in part by the Natural Science Foundation of Guangdong under Grant 2017A030311029, Grant 2016B010109002, Grant 2015B090912001, Grant 2016B010123005, and Grant 2017B0909 09005, in part by the Science and Technology Program of Guangzhou under Grant 201704020180 and Grant 201604020024, and in part by the Fundamental Research Funds for the Central Universities of China Grant 17lgzd08.

References

[1] H. Wang, C. Schmid, Action recognition with improved trajectories, in: 2013 IEEE International Conference on Computer Vision, 2013, pp. 3551–3558, doi:10.1109/ICCV.2013.441.
[2] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision, Springer, 2010, pp. 143–156.
[3] H. Jgou, F. Perronnin, M. Douze, J. Snchez, P. Prez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716, doi:10.1109/TPAMI.2011.235.
[4] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4305–4314, doi:10.1109/CVPR.2015.7299059.
[5] J. Zhang, H. Hu, Deep spatiotemporal relation learning with 3d multi-level dense fusion for video action recognition, IEEE Access (2019) 1, doi:10.1109/ACCESS.2019.2895472.
[6] J. Zhang, H. Hu, Residual gating fusion network for human action recognition, in: Chinese Conference on Biometric Recognition, Springer, 2018, pp. 79–86.
[7] W. Zhu, J. Hu, G. Sun, X. Cao, Y. Qiao, A key volume mining deep framework for action recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1991–1999, doi:10.1109/CVPR.2016.219.
[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.
[9] Y.-G. Jiang, C.-W. Ngo, S.-F. Chang, Semantic context transfer across heterogeneous sources for domain adaptive video search, in: Proceedings of the 17th ACM international conference on Multimedia, ACM, 2009, pp. 155–164.
[10] L. Duan, D. Xu, I.W. Tsang, J. Luo, Visual event recognition in videos by learning from web data, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1667–1680, doi:10.1109/TPAMI.2011.265.
[11] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in neural information processing systems, 2014, pp. 568–576.
[12] A. Diba, V. Sharma, L.V. Gool, Deep temporal linear encoding networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1541–1550, doi:10.1109/CVPR.2017.168.
[13] Y. Wang, M. Long, J. Wang, P.S. Yu, Spatiotemporal pyramid network for video action recognition, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2097–2106, doi:10.1109/CVPR.2017.226.
[14] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1933–1941, doi:10.1109/CVPR.2016.213.
[15] J.C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: European Conference on Computer Vision, Springer, 2010, pp. 392–405.
[16] A. Gaidon, Z. Harchaoui, C. Schmid, Temporal localization of actions with actoms, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2782–2795, doi:10.1109/TPAMI.2013.65.
[17] L. Wang, Y. Qiao, X. Tang, Latent hierarchical model of temporal structure for complex activity classification, IEEE Trans. Image Process. 23 (2) (2014) 810–822, doi:10.1109/TIP.2013.2295753.
[18] B. Fernando, E. Gavves, M.J. Oramas, A. Ghodrati, T. Tuytelaars, Modeling video evolution for action recognition, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5378–5387, doi:10.1109/CVPR.2015.7299176.
[19] J. Wu, Y. Zhang, W. Lin, Good practices for learning to recognize actions using fv and vlad, IEEE Trans. Cybern. 46 (12) (2016) 2978–2990, doi:10.1109/TCYB.2015.2493538.
[20] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Comput. Vision Image Understand. 150 (2016) 109–125.
[21] M. Marszałek, C. Schmid, H. Harzallah, J. Van De Weijer, Learning object representations for visual object class recognition, in: Visual Recognition Challange workshop, in conjunction with ICCV, 2007.
[22] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, 2005, pp. 886–893 vol. 1, doi:10.1109/CVPR.2005.177.
[23] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8, doi:10.1109/CVPR.2008.4587756.
[24] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC 2009-20th British Machine Vision Conference, British Machine Vision Association, 2009, pp. 124–131.
[25] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d–gradients, in: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008, pp. 275–281.
[26] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th ACM international conference on Multimedia, ACM, 2007, pp. 357–360.
[27] L. Liu, L. Shao, X. Li, K. Lu, Learning spatio-temporal representations for action recognition: a genetic programming approach, IEEE Trans. Cybern. 46 (1) (2016) 158–170, doi:10.1109/TCYB.2015.2399172.
[28] A. Liu, Y. Su, W. Nie, M. Kankanhalli, Hierarchical clustering multi-task learning for joint human action grouping and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (1) (2017) 102–114, doi:10.1109/TPAMI.2016.2537337.
[29] B.B. Amor, J. Su, A. Srivastava, Action recognition using rate-invariant analysis of skeletal shape trajectories, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2016) 1–13, doi:10.1109/TPAMI.2015.2439257.
[30] Y. Yuan, X. Zheng, X. Lu, A discriminative representation for human action recognition, Pattern Recognit. 59 (2016) 88–97.
[31] J.M. Carmona, J. Climent, Human action recognition by means of subtensor projections and dense trajectories, Pattern Recognit. 81 (2018) 443–455.
[32] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 677–691, doi:10.1109/TPAMI.2016.2599174.
[33] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4489–4497, doi:10.1109/ICCV.2015.510.
[34] Z. Tu, W. Xie, Q. Qin, R. Poppe, R.C. Veltkamp, B. Li, J. Yuan, Multi-stream cnn: learning representations based on human-related regions for action recognition, Pattern Recognit. 79 (2018) 32–43.
[35] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, S.J. Maybank, Asymmetric 3d convolutional neural networks for action recognition, Pattern Recognit. 85 (2019) 1–12.
[36] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1510–1517, doi:10.1109/TPAMI.2017.2712608.
[37] J.Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4694–4702, doi:10.1109/CVPR.2015.7299101.
[38] J. Zhang, Y. Han, J. Tang, Q. Hu, J. Jiang, Semi-supervised image-to-video adaptation for video action recognition, IEEE Trans. Cybern. 47 (4) (2017) 960–973, doi:10.1109/TCYB.2016.2535122.
[39] Y. Yang, Y. Yang, Z. Huang, H.T. Shen, Transfer tagging from image to video, in: Proceedings of the 19th ACM international conference on Multimedia, ACM, 2011, pp. 1137–1140.
[40] W. Bian, D. Tao, Y. Rui, Cross-domain human action recognition, IEEE Trans. Syst. Man Cyberne. Part B (Cyberne.) 42 (2) (2012) 298–307, doi:10.1109/TSMCB.2011.2166761.
[41] L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for heterogeneous domain adaptation, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, Omnipress, 2012, pp. 667–674.
[42] Z. Ma, Y. Yang, N. Sebe, A.G. Hauptmann, Knowledge adaptation with partiallyshared features for event detectionusing few exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 36 (9) (2014) 1789–1802, doi:10.1109/TPAMI.2014.2306419.
[43] L.A. Pereira, R. da Silva Torres, Semi-supervised transfer subspace for domain adaptation, Pattern Recognit. 75 (2018) 235–249.
[44] W. Wang, H. Wang, Z. Zhang, C. Zhang, Y. Gao, Semi-supervised domain adaptation via fredholm integral based kernel methods, Pattern Recognit. 85 (2019) 185–197.
[45] H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, C. Sun, Action recognition using nonnegative action component representation and sparse basis selection, IEEE Trans. Image Process. 23 (2) (2014) 570–581, doi:10.1109/TIP.2013.2292550.
[46] Y.C. Eldar, H. Rauhut, Average case analysis of multichannel sparse recovery using convex relaxation, IEEE Trans. Inf. Theory 56 (1) (2010) 505–519, doi:10.1109/TIT.2009.2034789.
[47] A. Argyriou, T. Evgeniou, M. Pontil, Convex multi-task feature learning, Mach Learn. 73 (3) (2008) 243–272.
[48] Z. Ma, Y. Yang, F. Nie, J. Uijlings, N. Sebe, Exploiting the entire feature space with sparsity for automatic image annotation, in: Proceedings of the 19th ACM international conference on Multimedia, ACM, 2011, pp. 283–292.
[49] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance (2014) 1–9 arXiv:1412.3474.
[50] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L. Guibas, L. Fei-Fei, Human action recognition by learning bases of action attributes and parts, in: 2011 International Conference on Computer Vision, 2011, pp. 1331–1338, doi:10.1109/ICCV.2011.6126386.

[51] K. Soomro, A.R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild (2012) arXiv:1212.0402.
[52] B. Yao, L. Fei-Fei, Grouplet: A structured image representation for recognizing human and object interactions, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 9–16, doi:10.1109/CVPR.2010.5540234.
[53] V. Delaitre, I. Laptev, J. Sivic, Recognizing human actions in still images: a study of bag-of-features and part-based representations, in: BMVC 2010-21st British Machine Vision Conference, 2010.
[54] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: A large video database for human motion recognition, in: 2011 International Conference on Computer Vision, 2011, pp. 2556–2563, doi:10.1109/ICCV.2011.6126543.
[55] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets (2014) arXiv:1405.3531.
[56] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, doi:10.1109/CVPR.2009.5206848.
[57] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[58] L. Jie, T. Tommasi, B. Caputo, Multiclass transfer learning from unconstrained priors, in: 2011 International Conference on Computer Vision, 2011, pp. 1863–1870, doi:10.1109/ICCV.2011.6126454.
[59] Y. Han, Y. Yang, Z. Ma, H. Shen, N. Sebe, X. Zhou, Image attribute adaptation, IEEE Trans. Multimedia. 16 (4) (2014) 1115–1126, doi:10.1109/TMM.2014.2306092.
[60] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International conference on machine learning, 2014, pp. 647–655.
[61] Z. Luo, J. Hu, W. Deng, H. Shen, Deep unsupervised domain adaptation for face recognition, in: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), 2018, pp. 453–457, doi:10.1109/FG.2018.00073.

Junxuan Zhang is currently a graduate student in the School of Electronics and Information Engineering, Sun Yat-sen, University, China. His major research interests include computer vision and pattern recognition. One particular interest is action recognition.

Haifeng Hu received the PhD degree from Sun Yat-sen University in 2004, and he is an associate professor of School of Electronics and Information Engineering at Sun Yat-sen University since July 2009. Now he is a visiting professor in Robotics Institute of Carnegie Mellon University. His research interests are in computer vision, pattern recognition, image processing and neural computation. He has published about 60 papers since 2000.