Contents lists available at ScienceDirect

Image and Vision Computing

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

Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition[☆]

Tiantian Xu[a, c], Fan Zhu[a, b], Edward K. Wong[a, c], Yi Fang[a, b, *]

[a] *NYU Multimedia and Visual Computing Lab, United Arab Emirates*
[b] *Department of Electrical and Computer Engineering, New York University Abu Dhabi, United Arab Emirates*
[c] *Department of Computer Science and Engineering, Tandon School of Engineering, New York University, United States*

ARTICLE INFO

Article history:
Received 16 September 2015
Received in revised form 10 December 2015
Accepted 2 January 2016
Available online xxxx

Keywords:
Cross-dataset
Action recognition
Neural network
Transfer learning
Domain adaptation

ABSTRACT

The emergence of large-scale human action datasets poses a challenge to efficient action labeling. Hand labeling large-scale datasets is tedious and time consuming; thus a more efficient labeling method would be beneficial. One possible solution is to make use of the knowledge of a known dataset to aid the labeling of a new dataset. To this end, we propose a new transfer learning method for cross-dataset human action recognition. Our method aims at learning generalized feature representation for effective cross-dataset classification. We propose a novel dual many-to-one encoder architecture to extract generalized features by mapping raw features from source and target datasets to the same feature space. Benefiting from the favorable property of the proposed many-to-one encoder, cross-dataset action data are encouraged to possess identical encoded features if the actions share the same class labels. Experiments on pairs of benchmark human action datasets achieved state-of-the-art accuracy, proving the efficacy of the proposed method.

1. Introduction

Human action recognition has drawn immense interests over the years, with its applications in a wide range of fields, including video labeling, video content retrieval, video surveillance, and sports video analysis. With the growing convenience of capturing and sharing videos, the computer vision community has seen a growing variety of human action datasets with substantial amount of videos. While the majority of these video data do not have annotations on them and hand labeling large datasets requires considerable amount of human efforts, researchers are interested in developing mechanisms to automatically generate annotations to these video data. Considering the fact that large-scale datasets always exhibit high intra-class variations, the requirement for a rational number of training data can easily go beyond the number of existing labeled data. Thus, researchers are thinking about the possibility of employing previously annotated datasets to facilitate automatic labeling of new datasets. For the human action recognition problem, different datasets share common actions. If it is possible to transfer knowledge between source and target action datasets, the annotated source dataset can serve as an augmentation to the training data, based on which effective labeling of the target dataset can be carried out. However, the auxiliary domain data may suffer from the serious domain-shift problem. For example, the action 'run' in one dataset may consist of videos of athletes running on field tracks while the same action from another dataset may contain videos of people running on the streets. In order to alleviate such problems, an algorithm that can reduce cross-domain variance is required. Algorithms of this type belong to transfer learning, which is a particular branch of machine learning that aims to utilize knowledge from one source or task to assist the same or a different task on another source.

In this paper, we tackle the problem of action recognition across four benchmark datasets. The major challenge comes from the significant cross-dataset variations. For example, *Diving* sequences in UCF Sports dataset [1] consist of TV broadcast videos with steady camera movements, controlled lighting, and trivial viewpoint changes, while videos of the same action class in HMDB 51 dataset [2] exhibit large variances in lighting, background, and viewpoints. Considering the significance of cross-dataset variations, hand-crafted features, which are designed to capture the discriminative properties of images or videos, are incapable of producing domain-invariant features for cross-domain datasets. Thus, we are interested in learning generalized feature representations across datasets, so that the resulting features from the source dataset can be used in the recognition of unseen instances in the target dataset. Our proposed method learns cross-dataset generalized features by training two

[☆] This paper has been recommended for acceptance by Ling Shao.
[*] Corresponding author. New York University Abu Dhabi, C1-156, P.O. Box 129188, Abu Dhabi, United Arab Emirates.
E-mail address: yfang@nyu.edu (Y. Fang).

http://dx.doi.org/10.1016/j.imavis.2016.01.001
0262-8856/© 2016 Elsevier B.V. All rights reserved.

many-to-one encoders on the source and target datasets in parallel, and maps raw features of instances from the two datasets to the same feature space. To our knowledge, this is the first time a dual many-to-one encoder architecture is used in cross-dataset action recognition. Contributions in this work are as follows:

- We have proposed a new transfer-learning method for cross-dataset action recognition. Our method can be easily generalized to other recognition tasks.
- We have designed a novel dual many-to-one encoder architecture for extracting generalized features across action datasets.
- We achieved over 10% increase in recognition accuracy over recent work in cross-dataset action recognition.

The organization of the paper is as follows: in Section 2 we present some background and related work; in Section 3 we present details of the proposed method; in Section 4 we describe our experimental results, and we conclude the paper in Section 5.

2. Background and related work

In this section, we present the background and related works that are essential to understanding the objective and methodologies behind our proposed method. We briefly review the concepts and methods in transfer learning and its benefits in cross-dataset human action recognition.

2.1. Transfer learning and its applications

Traditional machine learning methods follow the convention that the training data and the testing data are from the same distribution. However, this assumption is not always satisfied in real world applications. For example, when classifying web documents into several predefined categories, the test web documents may contain categories that are not sufficiently represented in the training data. Therefore, traditional learning methods may fail in such cases [3,4]. The objective of transfer learning is to transfer the knowledge from a source domain to the target domain. More specifically [5] gives the following definition:

Definition (*Transfer learning*)**.** Given a source domain D_s and learning task T_s, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_s \neq T_T$.

Surveys like [5] provides taxonomy related to transfer learning. Transfer learning can be further divided into three categories depending on the presence of labeled data in target and source tasks: 1) Inductive transfer learning, when labeled data is available in the target domain, 2) Transductive transfer learning, when labeled data is available only in source domain, and 3) Unsupervised transfer learning, when no labeled data is available in either source or target domains. Inductive transfer learning is closely related to multitask learning, its purpose is to learn both source and target tasks simultaneously, and transfer knowledge between the two tasks to improve performance on both [5]. Our proposed method falls into this category by learning two many-to-one encoders in parallel.

The type of knowledge being transferred can be divided into four categories: 1) instance transfer, 2) feature-representation transfer, 3) parameter transfer and 4) relational-knowledge transfer. Instance transfer uses source instances to train a classifier on target instances, usually by selecting or weighting source instances according to a certain metric. Feature-representation transfer reduces the differences between the source and target datasets by mapping one representation to the other, or by mapping both source and target representation to a common representation. Parameter transfer learns parameters shared by the source and target datasets. Finally, relational knowledge transfer uses a network or graph to explore the relationships within a dataset, and transfer the relationships from the source to the target [5].

Transfer learning has been adopted by the computer vision community in areas including object detection [6,7], object classification [8–11] and video event detection [12–16]. For example, Duan et al. [14] proposed an adaptive multiple kernel learning method by adapting classifiers based on base kernels and pre-learned average classifiers. Their method was applied to event recognition in consumer videos on the web. Lim et al. [7] proposed an object detection model that augments training data by borrowing and transforming instances from other classes. Gao et al. [17] designed a low-level feature model that could be used as a prior for learning new object class models from scarce training data. Tommasi and Caputo [18] performed domain adaptation of object classification inside the naive Bayes nearest neighbor framework by iteratively learning class metrics that can induce large between-class variance. Kulis et al. [11] addressed domain adaptation between different types of features and dimensionality for object classification by using asymmetric kernel transformation.

In human action recognition in particular, transfer learning has received increased interest over the years [19–23]. We will expand on the application of transfer learning in action recognition in Section 2.4.

2.2. Transfer learning with neural networks

With the revival of neural networks in recent years, many researchers have applied neural networks to transfer learning. Zhang [4] tackled the problem of cross-domain document classification using restricted Boltzman machine to learn a set of hierarchical features from source domain, then select a subset of the features by kernel-task alignment. Girshick et al. [24] adopted region proposal Convolutional Neural Networks in a two-stage object detection framework, where regions of interests were first detected and then semantics were learnt by fine-tuning. Oquab *et al.* [25] designed a method to reuse CNN layers trained on ImageNet, and transferred parameters for object classification in the PASCAL VOC dataset.

Different from the methods used in prior works, we propose a new method by training a pair of many-to-one encoders in parallel, and then map raw features from the source and target datasets to the same feature space. We will expand on the details of our method in Section 3.

2.3. Feature representation for action recognition

Human action recognition is a widely studied topic. The goal of human action recognition is to correctly classify actions performed by one or more persons into a pre-defined category. Survey papers like [26] provide taxonomy for this topic.

Efficacy of any action recognition method depended on the feature representation method used. Among recent work, Shao et al. [27] proposed a novel descriptor based on spatio-temporal Laplacian pyramid coding, which effectively extracts holistic representation without loss of information. Yu et al. [28] developed a structure-preserving binary representation for videos with depth information. Shao et al. [29] presented a method that extracts discriminative features by efficient spatio-temporal localization of human actions. Inspired by evolutionary method, Liu et al. [30] proposed a genetic programming approach toward learning spatio-temporal representations. To fuse different types of representations into one, Shao et al. [31] designed a spectral coding algorithm based on kernel multiview projections.

Although the above methods achieved satisfactory performance in action recognition, they were designed with the assumption that the training data and the testing data came from the same dataset. Unlike these methods, we propose a novel feature extraction method that can handle cross-dataset classification by learning generalized features. We show its efficacy in labeling unseen videos from the target dataset. Though our method was developed for action recognition, we argue that the proposed method can be generalized to other cross-dataset applications as well.

2.4. Action recognition via transfer learning

There has been an exploding interest in applying transfer learning techniques to action recognition [19,23,32,33]. There are interests in applying transfer learning to datasets generated by different sensors, *e.g.*, cameras, wearable sensors, or other sensor modalities [20,34]. Others are interested in applying transfer learning to datasets of the same domain, and most commonly from video sequences. While video sequences are the most common type of action datasets, the use of cameras to capture actions introduces complex issues; for example, different viewing angles, cluttered background, or changes in illumination, all contribute to significant variance in the captured videos. Therefore, action recognition in videos is a challenging problem.

Some researchers focus on transferring knowledge for action recognition between camera views [32]. For example, Liu et al. [10] extracted high level features shared across views by using bipartite graph partitioning on two view-dependent vocabularies. Zheng et al. [35] proposed to learn a pair of dictionaries on videos taken from different views and build a view-invariant sparse representation. Zhang et al. [36] used linear transformation to transform source view to target view via virtual path. Zhu and Shao [37] proposed to learn a sparse representation based on dense trajectory features by learning a view-independent basis dictionary and by forcing the same actions to have an identical representation. For applications in smart homes, Wu et al. [38] presented a multiview activity recognition technique by performing spatio-temporal feature fusion.

Other researchers tackled problems that are related to event search and abnormality recognition. Lam et al. [13] integrated transfer learning with relevance feedback to aid user's event query with known classification problems. Nater et al. [39] addressed the problem of abnormal event detection by adapting a Least-Square Support Vector Machine learnt from normal events to unseen events.

Some researchers proposed new adaptive transfer learning models. Yang et al. [40] proposed an adaptive support vector machine that transforms a classifier trained on a source dataset to work on a target dataset. Lin adn Li [41] extended the boosting-based learning method which allows classifiers built on a source action dataset to adapt to a new dataset.

In this paper, we tackle the problem of cross-dataset action recognition. In recent work, Cao et al. [42] used a probabilistic approach where a Gaussian mixture model learnt from a source dataset serves as a prior and is iteratively adapted to a target dataset for action detection. Sultani and Saleemi [43] proposed to use a weighted histogram by putting more weights on areas with high foreground confidence. This alleviates the problem in cross-dataset action recognition when the background may obscure recognition results. Different from these approaches, our method transfers knowledge across datasets by simultaneously learn a pair of many-to-one encoders. Using this model, we implicitly learn the distribution of raw features from the two datasets and map them to the same feature space.

2.5. Background

In this section, we present background for the two components essential to our method: action bank features and neural networks.

2.5.1. Action bank features

Feature extraction for action recognition from videos is a well-studied topic. Many established action recognition methods make use of low/mid-level features; *e.g.*, local space-time features [44,45,46], 3D dense point trajectories [47], and 3D gradient histograms [48,49]. Although these features achieve promising results on benchmark datasets [50], they lack robustness with respect to the range of semantics and the amount of intra-class variances they can handle. These drawbacks of low/mid-level features are particularly acute when used in cross-dataset recognition, where the capability of high level semantic representation and adaptivity toward large intra-class variances are needed.

Recently, Sadanand and Corso [51] proposed *action bank*, an action feature which is capable of learning high level semantic representation and is immune to a certain amount of intra-class variance. Action bank feature is essentially a template-based feature, where the templates, or 'banks', are extracted from video sequences across a range of actions and viewpoints. Given a video sequence, the banks are used as detectors, which decompose the video sequence into a conglomeration of responses at various spatiotemporal orientations. Then max pooling is applied to the responses, resulting in robustness to local displacements. Because of its high-level semantic representation and embedded view-invariance, action bank feature is an ideal candidate for use in our cross-dataset action recognition method. We will discuss the details of how we generate a generalized high-level feature from the action bank feature for cross-dataset action recognition in Section 3.

2.5.2. Neural networks

After a long plateau, neural network research has been revived in recent years with the emergence of deep learning [52,53]. The computer vision community has adopted deep learning in many applications; most notably object detection, object classification, hand-written digit recognition, and action recognition. In general, a neural network can be trained to approximate a function $f : X \rightarrow Y$, where X is the input and Y is the target output. A typical neural network consists of one or multiple layers of neurons whose inputs are determined by certain weights and biases, and outputs, or *activations*, are determined by a function of choice and passed among the neurons [54]. The layered architecture of neural networks, together with the adaptivity of tuning inherent weights and bias to fit specific tasks, make it a powerful model which can generalize to the mapping of input X to output Y.

In this work, we use a type of neural network called many-to-one encoders. Unlike auto-encoders, which force target outputs to be identical to the inputs, many-to-one encoders force target outputs to be identical among certain subsets of the inputs. In our case, the target outputs are set to be the same for input instances that belong to the same action class. The identical representation of same action class serves as a guide for encoder training, where intra-class variances are minimized across datasets. The trained encoders are used for feature extraction: given raw feature vectors as inputs, the encoders' outputs are a set of new high-level features which generalize across datasets. We will follow up with the details in Section 3.

3. Proposed method

In this section, we describe our proposed method in details. An overview of our method is presented in Fig. 1. Our method consists of three stages. In the first stage, preprocessing is done on both datasets, where action bank features are extracted and reduced to a smaller subspace. In the second stage, centroids of training instances from each class are extracted, which are then used to guide encoder training. In the third stage, a pair of many-to-one encoders are trained on the source and target datasets in parallel, after which

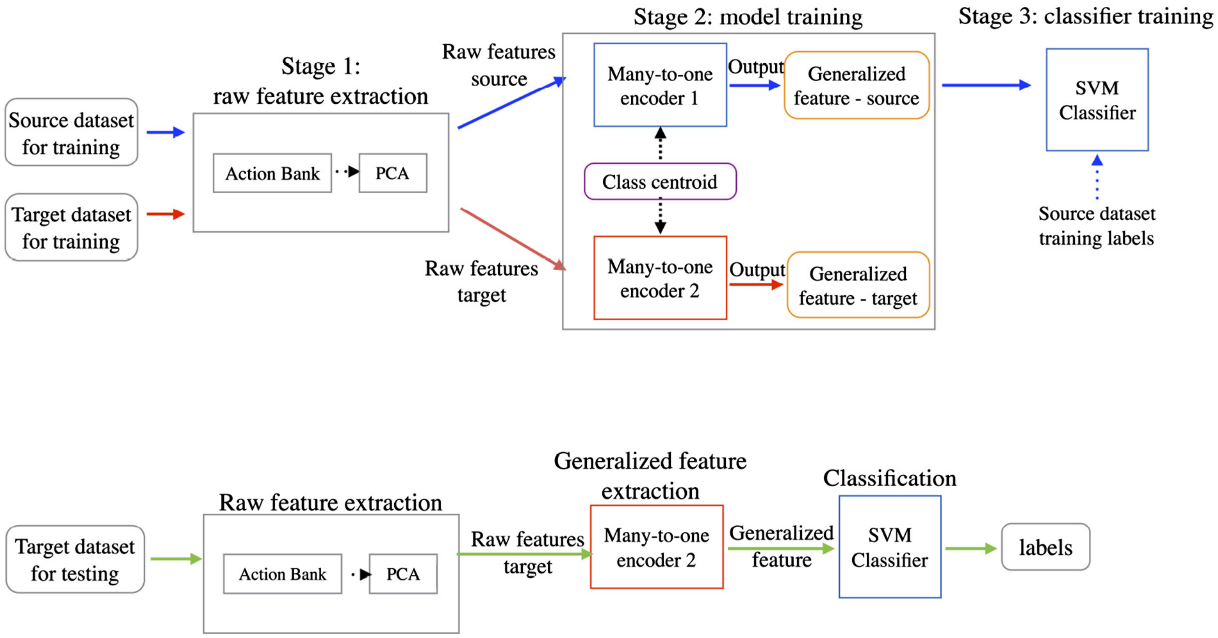


Fig. 1. Overview of proposed method. Top: trainig. Bottom: testing.

the activations of the output layer are extracted as the final features. Finally, a linear support vector machine classifier is built on the encoded features extracted from the source dataset, and then tested on the new features extracted from unseen samples of the target dataset.

3.1. Preprocessing

3.1.1. Action bank

We first process the datasets by extracting their action bank features [51]. Unlike low- or mid-level feature extraction methods [50], action bank method aims at extracting high level representation through semantic transfer. Given a test video clip, the action bank feature is extracted by first decomposing the video to energy responses at various spatiotemporal orientations. The responses are then correlated with a set of video clips across various semantic and viewpoints. More specifically, decomposition of videos are done via 3D Gaussian third derivative filtering $G_{3_{\hat{\theta}}}(p)$, where $\hat{\theta}$ captures the 3D direction of the filter symmetry axis and p denotes space-time position. Then, point-wise responses of video sequence I to this filter are squared and summed over spatiotemporal neighborhood Ω, s.t.

$$E_{\hat{\theta}}(p) = \sum_{p \in \Omega} (G_{3_{\hat{\theta}}} * I)^2 \qquad (1)$$

Seven types of spatiotemporal energies are evaluated, including static E_s, leftward E_l, rightward E_r, downward E_d, upward E_u, flicker E_f and lack of structure E_o. Each type of energy is computed as a sum over 4 basis third-order filters; *i.e.*, $E_{\hat{n}}(p) = \sum_{i=0}^{3} E_{\hat{\theta}_i}(p)$, where each $\hat{\theta}_i$ is a basis computed according to the conventional steerable filters as follows [55]

$$\hat{\theta}_i = \cos\left(\frac{\pi i}{4}\right) \hat{\theta}_a(\hat{n}) + \sin\left(\frac{\pi i}{4}\right) \hat{\theta}_b(\hat{n}) \qquad (2)$$

where $0 \le i \le 3$, $\hat{\theta}_a(\hat{n}) = \hat{n} \times \hat{e_p} / \parallel \hat{n} \times \hat{e_p} \parallel$, $\hat{\theta}_b(\hat{n}) = \hat{n} \times \hat{\theta}_a(\hat{n})$, and ê is the unit vector along the spatial x axis in the Fourier domain.

Finally, standard Bhattacharyya coefficient [56] is used to compute the correlation between energy responses of the test video and template videos. The correlations are then max-pooled, and concatenated to form the final action bank descriptor. Given N_a templates, N_s scales and an octree of three levels [57], the total length of action bank feature vector is $N_a \times N_s \times 73$. In this work, we use $N_a = 205$ templates and $N_s = 1$ scale, which result in feature vectors of length 14,965.

Note that the authors in [51] provided the source code and the action bank features for several benchmark datasets. We used the provided features directly in our experiments, and generated action bank features for datasets that do not already have the features available. The features and source code can be found at the authors' website.[1]

3.1.2. Principle component analysis

Action bank features have high dimensionality, which makes it inefficient for neural network training. To obtain low-dimensional features in the subspace, we use principle component analysis (PCA) [58]. PCA aims at reducing the dimensionality of a set of features by finding their eigenvectors and then project the original features onto the top p eigenvectors. By doing so, the original features can be represented by coordinates of the p eigenvectors in the subspace. In our method, we retain the top p eigenvectors such that the cumulative corresponding eigenvalues cover 99% of the total eigenvalues. In our experiments, this reduces feature dimension down to the range of 200 to 300, varying between datasets.

3.2. Model training

In this section, we present the architecture and details on the training of the neural networks.

[1] http://www.cse.buffalo.edu/jcorso/r/actionbank/.

Please cite this article as: T. Xu, F. Zhu, E. Wong, Y. Fang, Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.01.001

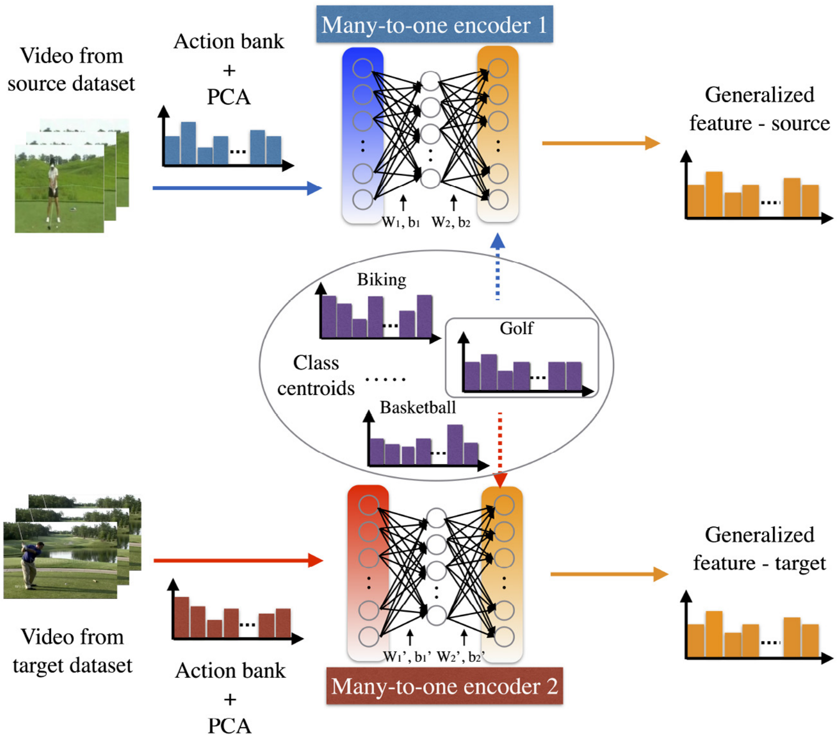


Fig. 2. Dual many-to-one encoder training. Top: encoder training for source dataset; bottom: encoder training for target dataset.

3.2.1. *Target output generation*

We use the centroid of each class as target output (Fig. 2). The class centroid is computed by averaging over instances' raw features in each class. Let $X_{s,c}^{i}$ and $X_{t,c}^{j}$ denote the i-th and j-th training instances (*i.e.*, raw feature vectors) from class c in the source and target dataset, respectively, the target output for instances from class c is:

$$T_c = \frac{1}{N_{s,c} + N_{t,c}} \left(\sum_{i=1}^{N_{s,c}} X_{s,c}^{i} + \sum_{j=1}^{N_{t,c}} X_{t,c}^{j} \right) \tag{3}$$

where $N_{s,c}$ and $N_{t,c}$ are the total number of training instances of class c from the source and target datasets.

3.2.2. *Neural network training*

At this stage, our method trains a pair of many-to-one encoders on the source and target datasets in parallel. Recall that for instances of the same action class, the target outputs of the two many-to-one encoders are identical. This setting forces the two many-to-one encoders to generalize to varying inputs and guide the outputs of same class instances to be similar. The advantages are twofold: 1) intra-class variance is minimized, and 2) instances of the same action class from the two datasets are mapped to the same feature space, thereby allowing knowledge transfer between the two datasets. While the first advantage has been demonstrated in recent publications [59,36] using single many-to-one encoder, in this work, we demonstrate the benefits of using a pair of many-to-one encoders in the context of transfer learning.

The architecture of the dual many-to-one encoders we used is illustrated in Fig. 2. They are fully-connected feedforward neural networks with an input layer, a hidden layer and an output layer. The two many-to-one encoders trained on the source and target datasets have identical architecture; *i.e.*, same number of layers and same number of neurons in each layer. In our method, we use only one hidden layer. Although conceptually, a deep network architecture with more than one hidden layer is beneficial for learning powerful representations, it has been shown that carefully configured and trained single-hidden-layer networks can achieve good performance in many tasks [60]. This observation is validated in our experiments as well.

After the preprocessing stage, the training videos from both datasets are represented by action bank features, which are then reduced to a smaller subspace via PCA. For simplicity, we denote one training instance as x_i, with $|x_i| = L$. Thus, both many-to-one encoders have an input layer of size L, a hidden layer of size H and an output layer of size L, where L is the output feature vector length, and H is a user defined parameter. In this work, we experimented with hidden layer sizes that range from 50 to 200, and input and output layer sizes that range from 200 to 300.

The goal of training the two networks in parallel is to find a mapping between training instances and the target outputs. Both networks have the same architecture but different parameters. More specifically, for the source network, the mapping is done via $f_1 : \mathbb{R}^L \to \mathbb{R}^H$ and $f_2 : \mathbb{R}^H \to \mathbb{R}^L$. f_1 and f_2 are defined as follows:

$$\{ f_1(x_i) = \sigma(W_1 x_i + b_1) f_2(f_1(x_i)) = \sigma(W_2 f_1(x_i) + b_2) \tag{4}$$

where $\sigma(\cdot)$ is the activation function, W_1, b_1, W_2 and b_2 are the parameters for f_1 and f_2, respectively. The network parameters are indicated in Fig. 2, where W_1, b_1, W_2 and b_2 are parameters to the network trained on the source dataset and W_1', b_1', W_2' and b_2' are parameters of the network trained on the target dataset.

We experimented with various hidden layer sizes and report the results in Section 4.5. Given a hidden layer size H, we initialized the weights and biases in both networks with random numbers drawn

Please cite this article as: T. Xu, F. Zhu, E. Wong, Y. Fang, Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.01.001

from a uniform distribution and with range between 0 and 1. We used the Sigmoid function as the activation function, *i.e.*,

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{5}$$

We define the objective function as follows:

$$E = \frac{1}{2N}\sum_{i=1}^{N} \| T_i - h(x_i, W_1, b_1, W_2, b_2) \|_2^2 \tag{6}$$

where $h(\cdot) = f_2(f_1(\cdot))$, W_1, W_2, b_1 and b_2 are the weights and biases of the network, and N is the number of training instances.

Training is done via stochastic gradient descent, where the objective function is minimized by iteratively updating the weights and biases [61]. For example, W_1 is updated as follows:

$$\left\{\Delta W_1(t+1) = \eta\frac{\delta E_{t+1}}{\delta W_1} + \mu \Delta W_1(t) W_1(t+1) = W_1(t) - \Delta W_1(t+1)\right. \tag{7}$$

where $\Delta W_1(t+1)$ is the update to W_1 at the $(t+1)$-th iteration, E_{t+1} is the value of the objective function at the $(t+1)$-th iteration, η denotes the learning rate, and μ denotes the momentum. W_2, b_1, and b_2 are updated in a similar manner.

After training, we use the values at the output layer as the final features. Note that given the complexity of the problem, when E is minimized, the values at the output layers of the network are approximate solutions instead of being identical to the pre-defined target outputs. The approximate solutions lie in the vicinity of the target outputs. Thus, the final features extracted from instances of the same action class, from both the source and target datasets, would lie in the same cluster. We will illustrate this phenomenon in Section 4.4. As will be shown in the following section, these features are linearly separable and therefore a linear classifier can be used for classification, which is very efficient compared to using kernel-based non-linear classifiers [43].

3.3. *Classification*

We use multi-class linear support vector machine (SVM) as classifier. Unlike a naïve binary classifier which does not apply constraints on the separation plane, SVM aims at choosing the separation plane where the distances between support vectors and the plane are maximized. This constraint introduces robustness to the classifier [62].

To perform cross-dataset classification, a linear SVM classifier is trained on features extracted from the source dataset and tested on features extracted from unseen instances from target dataset as shown in the bottom of Fig. 1. In Section 4, we show that such classification scheme can effectively classify unseen data across datasets. This is due to the successful knowledge transfer from the source dataset to the target dataset in stage 2 (Fig. 1).

4. Experiments and results

In this section, we present our experimental results on several benchmark datasets. We applied our method to the following pairs of datasets: UCF 50 - UCF Sports, UCF 50 - HMDB 51, UCF 50 - Olympic Sports and HMDB 51 - UCF Sports. Fig. 3 illustrates sample actions from these datasets. As can be seen in the figure, these datasets were taken from different sources and they exhibit significant variance for the same action class. We will start with describing the individual datasets, followed by details of the experimental settings.

4.1. *Datasets*

A. UCF Sports
The UCF Sports dataset [1] consists of 150 video sequences and contain 10 actions: diving, golf swinging, kicking, lifting, riding horse, running, skate boarding, swing bench, swing side, and walking. The videos have resolution of 720 x 480 pixels and most were collected from broadcast television channels like BBC and ESPN. The videos feature a wide range of scenes, viewpoints, and scales.

B. UCF 50
The UCF 50 dataset [63] features 50 real-life activities. They contain consumer videos taken from YouTube and have significant intra-class variance in background, viewpoints, and lighting conditions.

C. HMDB51
HMDB 51 [2] is a challenging large-scale action dataset with 51 classes and 6,849 video clips. This dataset consists of movie clips and consumer video clips taken from Prelinger archive, YouTube, and Google videos. The 51 action classes span from facial actions, general body movements, to human interactions, all accompanied by varying viewpoints, scale, background, and lighting conditions.

D. Olympic Sports
The Olympic Sports dataset [64] contains 16 actions. The videos were taken from YouTube and each video shows an athlete practicing one sport. Videos are subject to varying viewpoints, scale, background, and lighting.

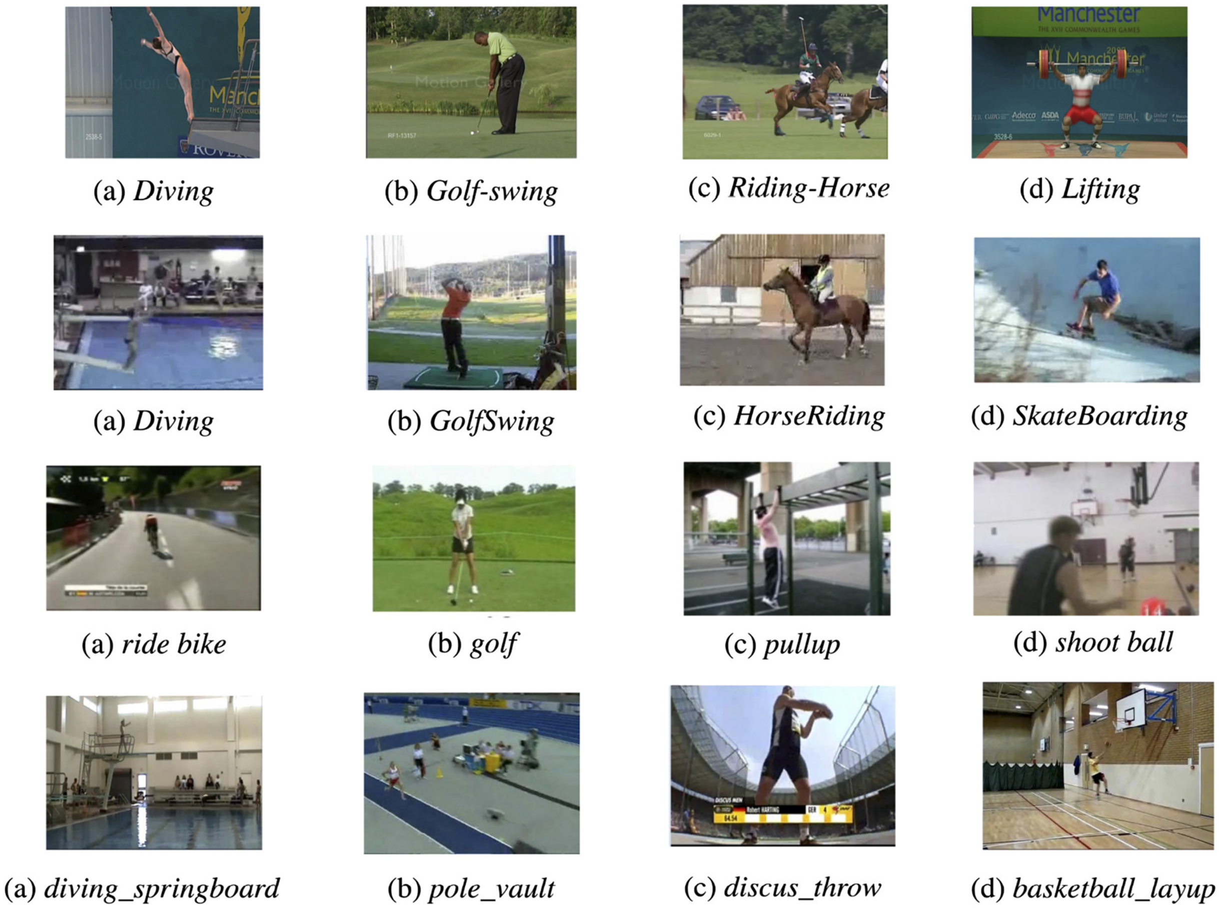
4.2. *Experimental settings*

All experiments were conducted with Matlab R2012a on a 64-bit Windows 7 PC with two 4-core 2.93 GHz Intel i7 CPUs and 8 GB of memory. We used the action bank features provided by the authors of [51] for datasets UCF 50, UCF Sports, and HMDB 51, and the code by the same authors to extract action bank features for the Olympic Sports dataset. We wrote the code for the many-to-one encoder and the stochastic gradient descent algorithm. For classification, we used the publicly available LIBSVM package [65].

In all experiments, each dataset was randomly split into training and testing sets. For videos from the same action class, roughly 70% of the videos were used for training and 30% were used for testing. Since the neural networks for the encoders were initialized randomly and the datasets were randomly split into training and testing sets, we repeat the experiments 3 times and report the average accuracy. When performing stochastic gradient descent for encoder training, we set learning rate $\eta = 0.1$ and momentum $\mu = 0.9$ (Eq. 7). Each encoder is trained for about 1,000 iterations.

4.3. *Cross-dataset recognition results*

In this section, we report the experimental results and compare them to prior work. The numbers in the tables below represent average class accuracy and H represents the number of neurons in the hidden layer of the many-to-one encoders. Following [43], we hand picked visually similar or common actions between each pair of datasets. For each pair, we experimented with classification in both directions; *e.g.*, for pair UCF 50 and UCF Sports, we trained classifier on UCF 50 and tested on UCF Sports, and then trained classifier on UCF Sports and tested on UCF 50. We describe the common actions and the experimental settings for each pair as follows. Note that the action names below are identical to the names from the original datasets.



(a) *Diving* (b) *Golf-swing* (c) *Riding-Horse* (d) *Lifting*

(a) *Diving* (b) *GolfSwing* (c) *HorseRiding* (d) *SkateBoarding*

(a) *ride bike* (b) *golf* (c) *pullup* (d) *shoot ball*

(a) *diving_springboard* (b) *pole_vault* (c) *discus_throw* (d) *basketball_layup*

Fig. 3. Illustration of benchmark action datasets used in this paper. Top to bottom row: *UCF Sports*, *UCF 50*, *HMDB 51*, and *Olympic Sports*.

A. UCF 50–UCF Sports.

The common actions between UCF 50 and UCF Sports are: Diving/dive, GolfSwing/golf, HorseRiding/riding, SkateBoarding/skate and CleanAndJerk/lift. The total number of videos are 723 and 58, respectively. Experimental results are shown in the following table (Table 1). It is interesting to see that although UCF Sports dataset has far less video clips than UCF 50 (58 *vs* 723), the classifier trained on UCF Sports dataset can still label UCF 50 fairly well.

Table 1
Experimental results for pair UCF 50 and UCF Sports. The best accuracies of each task are marked bold.

	$H = 50$	$H = 100$	$H = 150$	$H = 200$
UCF 50 → UCF Sports	**0.90**	0.83	0.85	0.85
UCF Sports → UCF 50	0.61	0.71	**0.76**	0.77

B. HMDB 51–UCF 50.

The common actions between HMDB 51 and UCF 50 are: ride_bike/Biking, golf/GolfSwing, pullup/PullUps, ride_horse/HorseRiding and shoot_ball/Basketball. The total number of videos are 559 and 740, respectively. Experimental results are shown in the table below (Table 2).

Table 2
Experimental results for pair HMDB 51 and UCF 50. The best accuracies of each task are marked bold.

	$H = 50$	$H = 100$	$H = 150$	$H = 200$
HMDB 51 → UCF 50	0.79	**0.82**	0.81	0.81
UCF 50 → HMDB 51	0.74	**0.82**	0.76	**0.82**

C. UCF 50–Olympic Sports.

The common actions between UCF 50 and Olympic Sports are: Basketball/basketball_layup, CleanAndJerk/clean_and_jerk, ThrowDiscus/discus_throw, Diving/diving_springboard_3m, PoleVault/pole_vault and TennisSwing/tennis_serve. The total number of videos are 860 and 304, respectively. The results are shown in Table 3. We compare our results with the best accuracy reported in [43] in Section 4.7. Note that unlike [43], we did not augment the Olympic Sports dataset with horizontally flipped version of the videos.

Table 3
Experimental results for pair UCF 50 and Olympic Sports. The best accuracies of each task are marked bold.

	$H = 50$	$H = 100$	$H = 150$	$H = 200$
UCF 50 → Olympic Sports	0.70	0.83	0.85	**0.87**
Olympic Sports → UCF 50	0.68	**0.75**	**0.75**	0.74

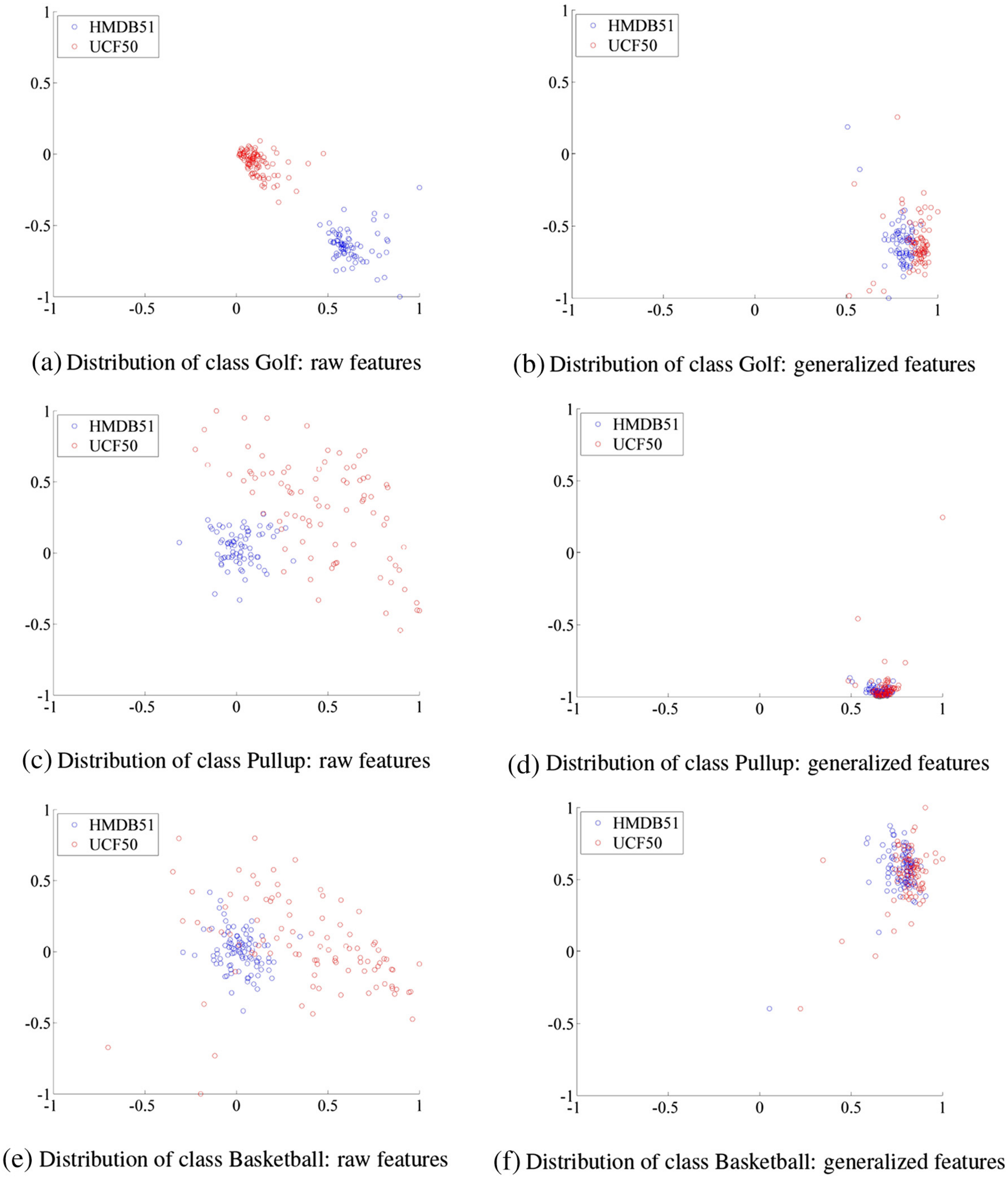
D. HMDB 51–UCF Sports.
The common actions between HMDB 51 and UCF Sports are: dive/dive, golf/golf, kick_ball/kick, ride_horse/riding and run/run. The total number of videos are 708 and 71, respectively. Experimental results are reported in the following table (Table 4).

Table 4
Experimental results for pair HMDB 51 and UCF Sports. The best accuracies of each task are marked bold.

	$H = 50$	$H = 100$	$H = 150$	$H = 200$
HMDB 51 → UCF Sports	0.91	0.90	0.88	**0.93**
UCF Sports → HMDB 51	0.49	0.51	**0.53**	0.53

4.4. Generalized feature extraction

To verify that by using the dual many-to-one encoder architecture, varying intra-class distributions from different datasets are indeed transformed to the same cluster, we plotted the distribution of the raw features of instances from the action class 'Golf' in datasets HMDB 51 and UCF 50 (Fig. 4). We compared the distributions of the raw action bank features and the learnt generalized feature. For illustration purposes, all features were reduced to a 2-dimensional subspace via PCA, and scaled to lie within the range [−1, 1]. As shown in Fig. 4, raw features from instances of the Golf class for HMDB 51 and UCF 50 form 2 distinct clusters (Fig. 4a), but they are merged into a single cluster after mapping to the generalized feature space (Fig. 4b).



(a) Distribution of class Golf: raw features

(b) Distribution of class Golf: generalized features

(c) Distribution of class Pullup: raw features

(d) Distribution of class Pullup: generalized features

(e) Distribution of class Basketball: raw features

(f) Distribution of class Basketball: generalized features

Fig. 4. Illustration of HMDB 51 (blue) and UCF 50 (red) class feature distributions before and after transfer learning.

Please cite this article as: T. Xu, F. Zhu, E. Wong, Y. Fang, Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.01.001

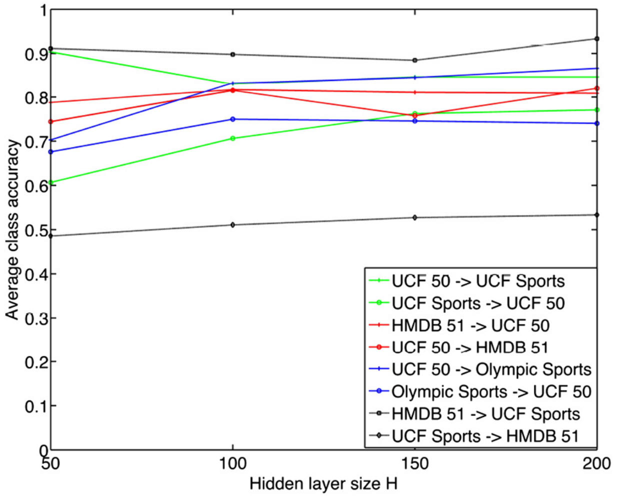


Fig. 5. Many-to-one encoder hidden layer size H and classification accuracy.

4.5. Hidden-layer size

We are interested in knowing how the hidden-layer size influences classification accuracy. As can be seen from Fig. 5, although different tasks tend to peak at certain hidden-layer size, in general, classification accuracy tends to increase as the number of neurons in the hidden layer increases. This can be explained by the increased flexibility of the network as hidden layer size increases, thus the model can generalize to more complex mapping.

4.6. Computation time

We evaluate the computation time of the proposed method and report the results in Table 5. All experiments were conducted on our lab PC and the language used was Matlab. The reported times were average running time over both directions of dataset pair, and the computation time for action bank features was not included. The reported times include model training, classifier training, and classification. As can be seen from the following table, our method can perform model training, classifier training, and test set classification very efficiently. The longest run time is just a little over 10 minutes for the pair UCF 50 and Olympic Sports and for $H = 200$. This pair of datasets contain over 1,100 training and test sequences. We attributed this efficient execution to the PCA dimension reduction of raw features, and the shallow single-hidden-layer network architecture.

Table 5
Average computation time of proposed method.

	$H = 50$	$H = 100$	$H = 150$	$H = 200$
UCF 50–UCF Sports	192.2 s	273.6 s	349.7 s	421.8 s
HMDB 51–UCF 50	259.8 s	355.9 s	441.0 s	455.3 s
UCF 50–Olympic Sports	319.2 s	408.9 s	499.8 s	667.2 s
HMDB 51–UCF Sports	154.0 s	209.0 s	257.9 s	269.7 s

4.7. Comparison with related work

We compare our experimental results with the best reported result in [43] (Table 6). For all four experiments, we used the same action classes and train/test ratio as [43]. It is worth mentioning that we did not augment the Olympic Sports dataset. Our method outperforms all four cross-dataset classification tasks. The key difference between our method and [43] is that our method uses transfer learning to learn a generalized feature across datasets while the latter aims at hand-crafting features on one dataset, and then apply them to the other dataset without explicit knowledge transfer. The strong performance of our method demonstrates the advantage of transferring knowledge between datasets, and more importantly, the efficacy of the proposed dual many-to-one encoder transfer learning method.

Table 6
Comparison between our results and related work. Higher accuracies for each task are marked bold.

	Ours	Sultani and Saleemi [43]
UCF 50 → HMDB 51	**0.82**	0.69
HMDB 51 → UCF 50	**0.82**	0.69
UCF 50 → Olympic Sports	**0.87**	0.33
Olympic Sports → UCF 50	**0.75**	0.48

5. Conclusion and future work

We introduced a new transfer learning method based on a novel dual many-to-one encoder architecture. We experimented with cross-dataset action recognition in several benchmark action datasets and demonstrated both the effectiveness and the efficiency of the proposed method on all tasks. We credited the impressive performance of the proposed method to the success of the domain-invariant feature extractions technique, which maps features from different datasets to a unified feature space. In the future, we will develop weakly-supervised or unsupervised algorithms for learning domain-invariant features. Another interesting direction is to investigate the transfer of knowledge between different action datasets using low/mid-level action features. Additionally, the proposed dual many-to-one encoder architecture can be easily generalized to other cross-dataset or cross-domain categorization problems.

References

[1] M.D. Rodriguez, J. Ahmed, M. Shah, Action MACH: A spatio-temporal maximum average correlation height filter for action recognition, 26th IEEE Conf. Comput. Vis. Pattern Recognit. CVPR (2008).

[2] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: A large video database for human motion recognition, Proc. IEEE Conf. Comput. Vis. (2011) 2556–2563. http://dx.doi.org/10.1109/ICCV.2011.6126543.

[3] L. Duan, D. Xu, I.W.H. Tsang, Domain adaptation from multiple sources: a domain-dependent regularization approach, IEEE Trans. Neural Netw. Learn. Syst. 23 (3) (2012) 504–518. ISSN 2162237X. http://dx.doi.org/10.1109/TNNLS.2011.2178556.

[4] J. Zhang, Deep transfer learning via restricted Boltzmann machine for document classification, Proc. — 10th Int. Conf. Mach. Learn. Appl. ICMLA 2011 1 (2011) 323–326. http://dx.doi.org/10.1109/ICMLA.2011.51.

[5] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359. ISSN 1041-4347. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5288526. http://dx.doi.org/10.1109/TKDE.2009.191.

[6] W. Li, L. Duan, D. Xu, I.W. Tsang, Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (6) (2014) 1134–1148. ISSN 01628828. http://dx.doi.org/10.1109/TPAMI.2013.167.

[7] J.J. Lim, R. Salakhutdinov, A. Torralba, Transfer learning by borrowing examples for multiclass object detection, Adv. Neural Inf. Process. Syst. 26 (NIPS 2012) (2012) 1–9.

[8] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6314 LNCS, 2010. pp. 213–226. ISBN 364215560X. http://dx.doi.org/10.1007/978-3-642-15561-1_16.

[9] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: an unsupervised approach, Proceedings of the IEEE International Conference on Computer Vision, 2011. pp. 999–1006. ISBN 9781457711015. http://dx.doi.org/10.1109/ICCV.2011.6126344.

[10] Y. Liu, D. Xu, I.W. Tsang, J. Luo, Textual query of personal photos facilitated by large-scale web data, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1022–1036. ISSN 01628828. http://dx.doi.org/10.1109/TPAMI.2010.142.

[11] B. Kulis, K. Saenko, T. Darrell, What you saw is not what you get: domain adaptation using asymmetric kernel transforms, Proc. IEEE Comput. Sci. Conf. Comput. Vis. Pattern Recognit. (2011) 1785–1792. ISSN 10636919. http://dx.doi.org/10.1109/CVPR.2011.5995702.

[12] L. Duan, I.W. Tsang, D. Xu, S.J. Maybank, Domain transfer SVM for video concept detection, 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, 2009. pp. 1375–1381. ISBN 9781424439935. http://dx.doi.org/10.1109/CVPRW.2009.5206747.

[13] A. Lam, A.K. Roy-Chowdhury, Christian R. Shelton, Interactive event search through transfer learning, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6494 LNCS (PART 3) (2011) 157–170. ISSN 03029743. http://dx.doi.org/10.1007/978-3-642-19318-7_13.

[14] L. Duan, D. Xu, I.W.H. Tsang, J. Luo, Visual event recognition in videos by learning from web data, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1667–1680. ISSN 01628828. http://dx.doi.org/10.1109/TPAMI.2011.265.

[15] L. Duan, D. Xu, S.F. Chang, Exploiting web images for event recognition in consumer videos: a multiple source domain adaptation approach, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012. pp. 1338–1345. ISBN 9781467312264. http://dx.doi.org/10.1109/CVPR.2012.6247819.

[16] L. Duan, I.W. Tsang, D. Xu, Domain transfer multiple kernel learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 465–479. ISSN 01628828. http://dx.doi.org/10.1109/TPAMI.2011.114.

[17] T. Gao, M. Stark, D. Koller, What makes a good detector? Structured priors for learning from few examples, ECCV (2012) 1–14. URL http://link.springer.com/chapter/10.1007/978-3-642-33715-4_26.

[18] T. Tommasi, B. Caputo, Frustratingly easy NBNN domain adaptation, Comput. Vis. (ICCV), 2013 IEEE Int. Conf. on (2013) 897–904. http://dx.doi.org/10.1109/ICCV.2013.116.

[19] D. Cook, K.D. Feuz, N.C. Krishnan, Transfer learning for activity recognition: a survey, Knowl. Inf. Syst. 36 (3) (2013) 537–556. ISSN 02191377. http://dx.doi.org/10.1007/s10115-013-0665-3.

[20] D.H. Hu, V.W. Zheng, Qiang. Yang, Cross-domain activity recognition via transfer learning, Pervasive Mob. Comput. 7 (3) (2011) 344–358. ISSN 15741192. http://dx.doi.org/10.1016/j.pmcj.2010.11.005.

[21] T.L.M. Van Kasteren, G. Englebienne, B.J.A. Kröse, Transferring knowledge of activity recognition across sensor networks, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6030 LNCS, 2010. pp. 283–300. http://dx.doi.org/10.1007/978-3-642-12654-3_17.

[22] C. Liu, P.C. Yuen, Human action recognition using boosted EigenActions, Image Vis. Comput. 28 (5) (2010) 825–835. ISSN 02628856. URL http://dx.doi.org/10.1016/j.imavis.2009.07.009. http://dx.doi.org/10.1016/j.imavis.2009.07.009.

[23] Z. Al-Halah, L. Rybok, R. Stiefelhagen, What to transfer? High-level semantics in transfer metric learning for action similarity, ICPR, Int. Conf. Pattern Recog. (2014) 2775–2780. ISSN 10514651. http://dx.doi.org/10.1109/ICPR.2014.478.

[24] R. Girshick, J. Donahue, T. Darrell, U.C. Berkeley, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR (2014) 2–9. ISSN 10636919. URL http://arxiv.org/abs/1311.2524. http://dx.doi.org/10.1109/CVPR.2014.81.

[25] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and Transferring mid-level image representations using convolutional neural networks, CVPR (2014) 1717–1724. ISSN 10636919.

[26] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (6) (2010) 976–990. ISSN 02628856. http://dx.doi.org/10.1016/j.imavis.2009.11.014.

[27] L. Shao, X. Zhen, D. Tao, X. Li, Spatio-temporal Laplacian pyramid coding for action recognition, IEEE Trans. Cybern. (2014).

[28] M. Yu, L. Liu, L. Shao, Structure-preserving binary representations for RGB-D action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (2015)

[29] L. Shao, S. Jones, X. Li, Efficient Search and localization of human actions in video databases, IEEE Trans. Circuits Syst. Video Technol. (2014).

[30] L. Liu, L. Shao, X. Li, L. Lu, Learning spatio-temporal representations for action recognition: a genetic programming approach, IEEE Trans. Cybern. (2015)

[31] L. Shao, L. Liu, M. Yu, Kernelized multiview projection for robust action recognition, IJCV (2015).

[32] A. Farhadi, M.K. Tabrizi, Learning to recognize activities from the wrong view point, ECCV Part I (2008) 154–166.

[33] A. Gupta, J. Martinez, J.J. Little, R.J. Woodham, 3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding, 2014, 2601–2608. http://dx.doi.org/10.1109/CVPR.2014.333.

[34] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, G. Tröster, Real–time transfer and evaluation of activity recognition capabilities in an opportunistic system, ADAPTIVE 2011, The Third International Conference on Adaptive and Self-Adaptive Systems and Applications 2011. pp. 73–78. ISBN 978-1-61208-156-4. URL http://www.thinkmind.org/index.php?view=article&amp;articleid=adaptive_2011_4_20_50035.

[35] J. Zheng, Z. Jiang, J. Phillips, R. Chellappa, Cross-view action recognition via a transferable dictionary pair, Proc. Br. Mach. Vis. Conf. 2012 (2012) 125.1–125.11. URL http://www.bmva.org/bmvc/2012/BMVC/paper125/index.html. http://dx.doi.org/10.5244/C.26.125.

[36] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, C. Shi, Cross-view action recognition via a continuous virtual path, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2013) 2690–2697. ISSN 10636919. http://dx.doi.org/10.1109/CVPR.2013.347.

[37] F. Zhu, L. Shao, Correspondence-free dictionary learning for cross-view action recognition, ICPR, Int. Conf. Pattern Recog. (2014).

[38] Chen. Wu, Amir Hossein. Khalili, Hamid. Aghajan, Multiview activity recognition in smart homes with spatio-temporal features, Proc. Fourth ACM/IEEE Int. Conf. Distributed Smart Cameras ICDSC '10 (2010) 142 URL http://portal.acm.org/citation.cfm?doid=1865987.1866010. http://dx.doi.org/10.1145/1865987.1866010.

[39] F. Nater, T. Tommasi, H. Grabner, L. Van Gool, B. Caputo, Transferring activities: Updating human behavior analysis, Proc. IEEE Int. Conf. Comput. Vis. (2011) 1737–1744. http://dx.doi.org/10.1109/ICCVW.2011.6130459.

[40] J. Yang, R. Yan, A.G. Hauptmann, Cross-domain video concept detection using adaptive svms, ACM Int. Conf. Multimedia (2007) 188. http://dx.doi.org/10.1145/1291233.1291276.

[41] X.M. Lin, S.Z.i. Li, Transfer AdaBoost learning for action recognition, ITME2009 — Proc. 2009 IEEE Int. Symp. IT Med. Educ. (2009) 659–664. http://dx.doi.org/10.1109/ITIME.2009.5236340.

[42] L. Cao, Z. Liu, T.S. Huang, Cross-dataset action detection, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. pp. 1998–2005. ISBN 9781424469840. http://dx.doi.org/10.1109/CVPR.2010.5539875.

[43] W. Sultani, I. Saleemi, Human action recognition across datasets by foreground-weighted histogram decomposition, Eecs. Ucf. Edu. (2014) URL http://www.eecs.ucf.edu/imran/PDFs/Sultani-Saleemi-CVPR-2014.pdf. http://dx.doi.org/10.1109/CVPR.2014.103.

[44] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, Proceedings — 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS, 2005, 2005. pp. 65–72. ISBN 0780394240. http://dx.doi.org/10.1109/VSPETS.2005.1570899.

[45] I. Laptev, T. Lindeberg, Local descriptors for spatio-temporal recognition, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3667 LNCS, 2006. pp. 91–103. ISBN 3540325336. http://dx.doi.org/10.1007/11676959_8.

[46] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007. pp. 1–8. ISBN 978-1-4244-1630-1. http://dx.doi.org/10.1109/ICCV.2007.4408988.

[47] H. Wang, A. Kläser, C. Schmid, C.L.i.n. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79. ISSN 09205691. http://dx.doi.org/10.1007/s11263-012-0594-8.

[48] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, 15th Int. Conf. Multimedia (c) (2007) 357–360. ISSN 14764687. URL http://dl.acm.org/citation.cfm?id=1291311. http://dx.doi.org/10.1145/1291233.1291311.

[49] A. Klaser, C. Schmid, I. Grenoble, A spatio-temporal descriptor based on 3D-gradients, Br. Mach. Vis. Conf. (2008).

[50] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, Proc. Br. Mach. Vis. Conf. 2009 (2009) 124.1–124.11. URL http://www.bmva.org/bmvc/2009/Papers/Paper143/Paper143.html. http://dx.doi.org/10.5244/C.23.124.

[51] S. Sadanand, J.J. Corso, Action bank: a high-level representation of activity in video, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (May) (2012) 1234–1241. ISSN 10636919. http://dx.doi.org/10.1109/CVPR.2012.6247806.

[52] R. Salakhutdinov, G. Hinton, Deep Boltzmann machines, Artif. Intell. 5 (2) (2009) 448–455. ISSN 15324435. URL http://www.cs.utoronto.ca/rsalakhu/papers/dbm.pdf.

[53] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1–9. ISSN 10495258.

[54] A.K. Jain, J.C. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, COMPUT. 29 (3) (1996) 31–44. ISSN 0018-9162. http://dx.doi.org/10.1109/2.485891.

[55] W.T. Freeman, E.H. Adelson, The design and use of steerable filters, IEEE Trans. Pattern Anal. Mach. Intell. 13 (9) (1991) 891–906. ISSN 01628828. http://dx.doi.org/10.1109/34.93808.

[56] K.G. Derpanis, M. Sizintsev, K. Cannons, R.P. Wildes, Efficient action spotting based on a spacetime oriented structure representation, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. pp. 1990–1997. ISBN 9781424469840. http://dx.doi.org/10.1109/CVPR.2010.5539874.

[57] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, 2006. pp. 2169–2178. ISBN 0769525970. http://dx.doi.org/10.1109/CVPR.2006.68.

[58] L.I. Smith, A tutorial on principal components analysis introduction, Stat. 51 (2002) 52 ISSN 03610926. http://dx.doi.org/10.1080/03610928808829796.

[59] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, E. Wong, 3D Deep Shape Descriptor, 2015. pp. 2319–2328. http://dx.doi.org/10.1109/CVPR.2015.7298845.

[60] A. Coates, A. Arbor, A.Y. Ng, An analysis of single-layer networks in unsupervised feature learning, Aistats 2011 (2011) 215–223. http://dx.doi.org/10.1109/ICDAR.2011.95.

[61] Y. LeCun, L. Bottou, G.B. Orr, K.-R. Muller, Efficient BackProp, Springer (1988) 9–50.

[62] J. Suykens, J. Suykens, J. Vandewalle, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.

[63] K.K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Mach. Vis. Appl. 24 (5) (2013) 971–981. ISSN 09328092. http://dx.doi.org/10.1007/s00138-012-0450-4.

[64] J.C. Niebles, C.-W. Chen, Li. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, Lect. Notes Comput. Sci. (Incl. subseries Lect. Notes Artif. Intell. Lect. Notes Bioinforma.) 6312 LNCS (PART 2) (2010) 392–405. ISSN 03029743. http://dx.doi.org/10.1007/978-3-642-15552-9_29.

[65] C.-C. Chang, C.-J. Lin, LIBSVM, 2 (3) (2011) 1–27. ISSN 21576904. http://dx.doi.org/10.1145/1961189.1961199.

Please cite this article as: T. Xu, F. Zhu, E. Wong, Y. Fang, Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition, Image and Vision Computing (2016), http://dx.doi.org/10.1016/j.imavis.2016.01.001