Boosted Cross-Domain Categorization

Fan Zhu[1]
fan.zhu@sheffield.ac.uk
Ling Shao[1]
ling.shao@ieee.org
Jun Tang[2]
tangjunahu@gmail.com

[1] Departmetn of Electronic and Electrical Engineering
The University of Sheffield
Sheffield, S1 3JD, UK

[2] School of Electronics and Information Engineering
Anhui University
Hefei, 230601, China

Abstract

A boosted cross-domain categorization framework that utilizes labeled data from other visual domains as the auxiliary knowledge for enhancing the original learning system is presented. The source domain data under a different data distribution are adapted to the target domain through both feature representation level and classification level adaptation. The proposed framework is working in conjunction with a learned domain-adaptive dictionary pair, so that both the source domain data representations and their distribution are optimized in order to match the target domain. By iteratively updating the weak classifiers, the categorization system allocates more credits to "similar" source domain samples, while abandoning "dissimilar" source domain samples. Using a set of Web images and selected categories from the HMDB51 dataset as the source domain data, the proposed framework is evaluated with both image classification and human action recognition tasks on the Caltech-101 and the UCF YouTube datasets, respectively, achieving promising results.

1 Introduction

During the last decade, knowledge transfer-based learning methods have gained their popularity in computer vision and pattern recognition [22]. Utilizing auxiliary data from other resources, transfer learning techniques aim at either dealing with the insufficient training data issue, e.g., one-shot-learning [5], [19] and zero-shot-learning [20], [21], or enhancing the discriminability of existing learning systems [33]. Data representation level knowledge transfer and classification level knowledge transfer are two major transfer learning branches in terms of learning stages. The former includes learning a bipartite graph via co-clustering [14], learning a cross-domain dictionary pair [33], etc., and the latter includes learning an adaptive-SVM (A-SVM) classifier [28], a projective model transfer SVM (PMT-SVM) classifier [2], TrAdaBoost [7], etc. Some popular knowledge transfer scenarios in computer vision include 1) cross-view action recognition [14], where training and query actions come from different observation viewpoints; 2) cross-domain image classifciation [6], where the depth information can be incorporated in the training stage to mine more information from the RGB channel of query images; 3) cross-modality information retrieval [18], which attempts to retrieve documents from one media form (e.g., image) to another media form (e.g., text).

We introduce a boosted cross-domain categorization (BCDC) framework that utilizes labeled data from other domains as the source data to span the intra-class diversity of the original learning system. In addition to the manually annotated information in the target domain, partially labeled data from another visual domain are provided as the source domain. A boosted classification framework is introduced to work in conjunction with a cross-domain dictionary learning method [33]. Through iteratively updating both the source domain data representations and their distribution, the source domain training instances can be optimized, and thus can help improve the visual categorization tasks in the target domain. In comparison, the proposed learning framework shares the same basic principle of sequentially updating the impacts of training instances; yet our learning framework attempts to sequentially update the data representations of those “dis-similar” samples instead of simply weighting less on them. Thus, unlike most transfer learning frameworks, knowledge is transferred through both the data representation level and the classification level in the proposed BCDC framework.

2 Related work

Dictionary learning has seen a variety of applications in computer vision tasks along with sparse representations, e.g., face recognition [25] and image denoising [32]. Using an overcomplete dictionary, sparse modeling of signals can approximate the input signal by a sparse linear combination of items from the dictionary. Many algorithms [12], [24], [25] have been proposed to learn such a dictionary according to different criteria, among which the K-Singular Value Decomposition (K-SVD) algorithm [1] is a classical dictionary learning algorithm that focuses on the reconstructive ability. Zhu and Shao [33] extended the classical dictionary learning approach to a weakly-supervised cross-domain learning scenario. By learning a discriminative, reconstructive and domain-adaptive dictionary pair, such a cross-domain dictionary learning approach achieves promising performance on popular visual categorization benchmarks. However, such a method considers all training data equally, which does not allow the existence of a large number of “noisy” data.

AdaBoost [9] is a classical machine learning algorithm that aims at boosting the performance of weak classifiers by carefully adjusting the weights of training instances. AdaBoost can be easily generalized to a wide range of applications by jointly working with other learning algorithms to achieve improved performance. Specifically, AdaBoost constructs a “strong" classifier as a linear combination of weak classifiers, where each weak classifier is considered to be helpful as long as it results in an error rate lower than 0.5 for binary classification. In each iteration, previous predictions are used to update the weights of training instances so that the weights of the incorrectly-classified instances in the previous iteration are increased while the weights of the correctly-classified instances are decreased. Leveraging such a weight updating mechanism, Zhang *et al.* [30] attempted to capture more discriminative information by learning a set of codebooks in sequence. As an extension to AdaBoost, Dai *et al.* [7] proposed TrAdaBoost to utilize the mismatched data in an auxiliary feature domain for the classification task in the target feature domain. In each boosting iteration of TrAdaBoost, the weights of those wrongly predicted training instances in the auxiliary domain are decreased so that their impacts towards the global data distribution are weakened. However, similar as other classifier level transfer learning techniques, the intrinsic data representations of those “dis-similar” training instances are not changed through the boosting procedure of TrAdaBoost.

3 Boosted cross-domain categorization

3.1 Problem formulation

Some general notions are defined as follows for later usage: let $\mathcal{D}^t = \mathcal{D}^t_l \cup \mathcal{D}^t_u$ denote the target domain data, where the labeled parts are denoted by $\mathcal{D}^t_l$ and the unlabeled parts are denoted by $\mathcal{D}^t_u$. Similarly, let $\mathcal{D}^s = \mathcal{D}^s_l \cup \mathcal{D}^s_u$ denote the source domain data. Since the unlabeled source domain data $\mathcal{D}^s_u$ and the labeled source domain data $\mathcal{D}^s_l$ share the same feature distribution, $\mathcal{D}^s_u$ are first labeled in a semi-supervised manner using an efficient manifold ranking algorithm [31] so that they can be formed as labeled parts $\mathcal{D}^s_{l*}$ for the next stage usage. In order to bring data across different domains into the same feature space, knowledge transfer is conducted upon both $\mathcal{D}^t_l$ and $\hat{\mathcal{D}}^s = \mathcal{D}^s_l \cup \mathcal{D}^s_{l*}$. The goal is to learn a combination of a set of classifiers and specify different class labels to the unlabeled instances $\mathcal{D}^t_u$ using the labeled ones $\mathcal{D}_l = \mathcal{D}^t_l \cup \hat{\mathcal{D}}^s$ through an iterative boosting procedure.

3.2 Learning a cross-domain dictionary pair

Let Y_t be the set of target domain n-dimensional input signals, which contain N training instances, i.e., $Y_t = [y^1_t, y^2_t, ..., y^N_t] \in \Re^{n \times N}$. Learning a reconstructive dictionary for obtaining the sparse representation of the target domain signals Y_t can be accomplished by solving the following optimization problem:

$$\begin{aligned} \langle D_t, X_t \rangle =& arg \min_{D_t, X_t} \|Y_t - D_t X_t\|_2^2 \\ & s.t. \forall i, \|x^i_t\|_0 \leq T, \end{aligned} \tag{1}$$

where $D_t \in \Re^{\mathcal{K} \times n'}$ denotes the target domain dictionary and $X_t = [x^1_t, ...x^N_t] \in \Re^{n' \times N}$ denotes the set of sparse signals. The number of dictionary items K_t is set to significantly exceed the number of training instances N to secure that the dictionary is over-complete. T is the sparsity constraint factor that limits the number of non-zero elements in the sparse codes, so that the number of items in the decomposition of each signal x_i is less than T.

The choice of a method for dictionary learning critically determines the performance of sparse representation. The K-SVD algorithm [1] is a popular and efficient dictionary learning method that focuses on minimizing the reconstruction error. Some discriminative approaches [29], [27], [16], [17], [15], [3] show their privilege over the K-SVD algorithm by incorporating extra discriminative terms into the objective function for dictionary learning. However, the discriminative terms appear to be introduced to these approaches without considering the data distribution of the training samples, i.e., samples with high confidence possess the same impact as those with low confidence. Such weakness becomes even more severe when dealing with data mismatching scenarios. When allocated with discriminative elements under no smoothness guarantee, performing dictionary learning on both target domain data and mismatched data from a different feature domain can break the smoothness property of the original target domain.

Zhu and Shao [33] extended the dictionary learning function in equation 1 to a cross-domain scenario, and included a discriminative term into the objective function. The discriminative cross-domain dictionary learning function is formulated as:

$$\begin{aligned}&\langle D_t, D_s, X_t, \Phi, \mathcal{P}\rangle\\ &= arg \min_{D_t,D_s,X_t,\Phi,\mathcal{P}} \|Y_t - D_t X_t\|_2^2\\ &+\alpha\|Q-\Phi X_t\|_2^2+\beta\|\mathcal{H}-\mathcal{P}X_t\|_2^2\\ &+\|Y_s\mathbb{A}^T - D_s X_t\|_2^2 \quad s.t.\forall i,\ \|x_t^i\|_0 \le T.\end{aligned} \tag{2}$$

Similar as Y_t and D_t, Y_s and D_s denote the input signals and the dictionary in the source domain, respectively. Scalers α and β are set to control the relative contribution of the terms $|Q-\Phi X_t\|_2^2$ and $\|\mathcal{H}-WX_t\|_2^2$, where Φ is a linear transformation matrix that maps the original sparse codes to be in correspondence with the sparse codes $Q=[q_1,q_2,\cdots,q_N]\in\Re^{K\times N}$ of the target domain input signal Y_t and $\mathcal{H}=[h_1,h_2,\cdots,h_N]\in\Re^{C\times N}$ are the class labels of Y_t, given that the non-zero element indicates the class of an input signal within each column $h_i=[0,\cdots,1,\cdots,0]^T\in\Re^C$. The term $\mathbb{A}$ is a reversible binary matrix which establishes the one-to-one correspondences across the target domain data Y_t and the source domain data Y_s. $\mathbb{A}$ is assumed as leading to a perfect mapping across the sparse codes X_t and X_s, thus each matched pair of samples in different domains possesses an identical representation after encoding, i.e., $\|X_t^T-\mathbb{A}X_s^T\|_2^2=0$ and $\|Y_t^T-\mathbb{A}Y_s^T\|_2^2=0$. The discriminative information are included in the dictionary learning function through the terms Q and $\mathcal{H}$. As can be observed from the definitions of both terms, equal credits are allocated to all data. However, such an assumption that all samples can equally contribute to the categorization system does not apply real-world scenarios.

3.3 Boosted classification

In order to distinguish the "dissimilar" data from the smooth data, we include the weighted discriminative sparse codes into the learning function. Specifically, $q_i=[q_i^1,q_i^2,\cdots,q_i^K]^T=[0,\cdots,w_i,w_i,\cdots,0]^T\in\Re^K$, where the non-zeros occur at those indices where $y_t^i\in Y_t$ and $X_t^k\in X_t$ share the same class label. Given $X_t=[x_1,x_2,\cdots,x_6]$ and $Y_t=[y_1,y_2,\cdots,y_6]$, and assuming x_1, x_2, y_1 and y_2 are from class 1, x_3, x_4, y_3 and y_4 are from class 2, x_5, x_6, y_5 and y_6 are from class 3, Q is then defined with the following form:

$$\begin{pmatrix} w_1 & w_2 & 0 & 0 & 0 & 0\\ w_1 & w_2 & 0 & 0 & 0 & 0\\ 0 & 0 & w_3 & w_4 & 0 & 0\\ 0 & 0 & w_3 & w_4 & 0 & 0\\ 0 & 0 & 0 & 0 & w_5 & w_6\\ 0 & 0 & 0 & 0 & w_5 & w_6 \end{pmatrix}, \tag{3}$$

Since predictions are made with respect to the data distribution of X_t, w_i is included in each item of $\mathcal{H}$. Thus $\mathcal{H}$ can be defined as follows according to the same example in Equation (3)

$$\begin{pmatrix} w_1 & w_2 & 0 & 0 & 0 & 0\\ 0 & 0 & w_3 & w_4 & 0 & 0\\ 0 & 0 & 0 & 0 & w_5 & w_6 \end{pmatrix}. \tag{4}$$

By defining $Y = (Y_t^T, (Y_s\mathbb{A}^T)^T, \sqrt{\alpha}Q^T, \sqrt{\beta}\mathcal{H}^T)^T$ and $D = D_t^T, D_s^T, \sqrt{(\alpha)}\Phi^T, \sqrt{(\beta)}\mathcal{P}^T)^T$, where column-wise L_2 normalization is applied to D, the objective function in equation 2 can be solved through sequentially updating dictionary atoms and sparse codes as in [33].

Algorithm 1 Boosted cross-domain dictionary learning

```
Input the labeled target domain data $\mathcal{D}_l^t$ and the source domain data $\hat{\mathcal{D}}^s$, the maximum number of iterations $Max.iter$ and the Weak Learner.
Output a "strong" classifier $\mathcal{F}(\cdot)$ and updated representations of the source domain instances.
Initialize the data distribution as uniform, i.e., the initial weights $w^1 = (w_1^1, w_2^1, \cdots, w_{N+M}^1)$ have an identical value. Cross-domain discriminative dictionary learning is applied to both target domain and source domain data under the initialized uniform distribution, so that $\mathcal{D}_l^t$ and $\hat{\mathcal{D}}^s$ can be represented by $X_t$ and $X_s^1$ respectively.
for $j = 1$ to $Max.iter$ do
    1. Set data distribution $p^j = \frac{w^j}{\sum_{i=1}^{N+M} w_i^j}$
    2. Update $X_s^j$ as the new representation of $\hat{\mathcal{D}}^s$ under data distribution $p^j$ with cross-domain discriminative dictionary learning.
    3. Compute the hypothesis $h_t^j : X_t \to l(X_t)$ and $h_s^j : X_s^t \to l(X_s)$, providing that $p^j$ is over both $\mathcal{D}_l^t$ and $\hat{\mathcal{D}}^s$.
    4. Calculate the error $\varepsilon^j$ of $h_t^j$:
           $$\varepsilon^j = \sum_{i=1}^{N} \frac{w_i^j \times |h_t^j(x_i) - l(x_i)|}{\sum_i^N w_i^j},$$
       where $\varepsilon^j$ is required to be less than 0.5.
    5. Set $\beta_t^j = \frac{\varepsilon^j}{1-\varepsilon^j}$ and $\beta_s = \frac{1}{1+\sqrt{2\ln M/Max.iter}}$
    6. Update the new weight vector:
           $$w_i^{j+1} = \begin{cases} w_i^j {\beta_t^j}^{-|h_t^j(x_i)-l(x_i)|}, & 1 \leqslant i \leqslant N \\ w_i^j \beta_s^{|h_s^j(x_i)-l(x_i)|}, & otherwise. \end{cases}$$
end for
```

We consider the similarities between the source domain training instances $\hat{\mathcal{D}}^s$ and the target domain training instances $\mathcal{D}_l^t$ according to the present distribution. When a set of source domain instances are incorrectly predicted due to distribution changes by the present learner, these instances are considered to be most "dissimilar" to the target domain instances. Thus, the weights of these training instances are decreased correspondingly by multiplying the factor $\beta_s^{|h_s^j(x_i)-l(x_i)|} \in (0,1]$, where $l(x_i)$ returns the label of instance x_i, so that these instances will affect the learning process less in the next iteration. In addition to updating the weights, cross-domain discriminative dictionary learning is applied to lead those "dissimilar"

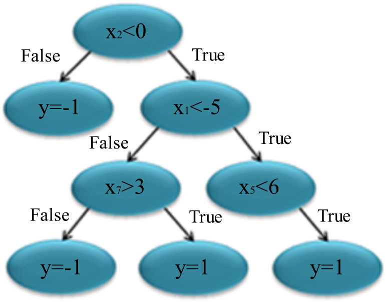

Figure 1: Example of a CART tree.

instances towards more appropriate representations. The confidence of the discriminative term is measured by allocating the updated weights to the binary representation, so that those correctly predicted instances can make more impacts when learning the dictionary pair. Consequently, when the stop criterion is reached, some "dissimilar" training instances can be represented in a "similar" form, and the training instances in $\hat{\mathcal{D}}^s$ which lead positive impacts to the learning system will process larger training weights than those "dissimilar" ones. The weight updating mechanism in the target domain is kept in accordance with the original AdaBoost [9] by multiplying the factor $\beta_t^{j-|h_t^j(x_i)-l(x_i)|}$, so that the weights of those incorrectly classified instances in $\mathcal{D}_l^t$ are increased in order to make the new classifier focus on those instances in the next iteration. Since the aim is only to guarantee the instances in the target domain being correctly classified, the two cross purpose weighting mechanisms within the same learning system do not conflict. The pseudo code of the proposed boosted learning technique is given in Algorithm 1.

3.4 Weak classifiers

The Classification and Regression Trees (CART) [4] is used as the weak classifier in this work. The CART classification is a process of tree traverse, where a tree node represents a predicate and the value associated with a tree leaf is the class of the presented instance. For the construction of a node in CART, we first find a threshold for each of the n dimensions that separates the training instances with the least error. When the dimension i with the least error is chosen, the node can be constructed as either a predicate or branches that are connected with tree leafs. All the errors are evaluated according to the updated weights, so that the training instances can be learned with respect to their present distribution. An example of a CART tree is shown in Figure 2.

4 Experiments

The experiments are conducted on 4 different data sources, where the UCF YouTube dataset [13] and the Caltech-101 dataset [8] are treated as the target domains, and the HMDB51 dataset [11] and some Web images indexed by Google are treated as the source domains. To obtain the source domain Web images, we select the first 20 categories out of the 101 categories, and randomly choose 20-30 images for each category among the first 100 searching results returned by Google when using category names as the key words. For both action recognition and image classification tasks, the BCDC method is evaluated on the categories which exist in both the target and source domains.

4.1 Image classification

We adopt the dense SIFT descriptors plus the sparse coding approach [26] for low-level and mid-level image representations. The weight α on the label constraint term and the weight β on the classification error term are set as 4 and 2 respectively. We run our method on five different partitions of the Caltech-101 dataset, where the number of 10/15/20/25/30 images are randomly chosen as the training images while the remaining images are used for testing for each partition. In order to demonstrate the effectiveness of our proposed approach, we compare with the baseline Sparse-coding Spatial Pyramid Matching (ScSPM) [26], K-Singular Value Decomposition (K-SVD) [1], Label Consistent-Singular Value Decomposition (LC-KSVD) [10], AdaBoost [9], and Weakly Supervised Cross-Domain Dictionary Learning (WSCDDL) [33] [1] and Transfer AdaBoost (TrAdaBoost) [7]. Experimental results are reported in TABLE 1 and TABLE 2 when source domain data are applied or not applied respectively. Results on the first 20 selected image categories of the Caltech-101 dataset using five different numbers of training data are reported, and all the results are obtained by averaging 5 runs of randomly selected training and testing images to guarantee the reliability. The proposed BCDC method consistently leads to the best performance over other methods. The reported results of ScSPM, K-SVD and LC-KSVD in TABLE 1 are obtained by simply treating the source domain data as extra training data without knowledge transfer. Note that the performance of ScSPM, K-SVD and LC-KSVD is even decreased when source domain data are used, which further validates the importance of our boosted cross-domain categorization method. Figure 3 shows the error rate comparison of the proposed method and TrAdaBoost according to the boosting iterations on the Caltech-101 dataset when using 30 training samples per category.

Table 1: Performance comparison between the BCDC and state-of-the-art methods on the Caltech-101 dataset with source domain data.

Algorithm	ScSPM [26]	K-SVD [1]	LC-KSVD [10]	TrAdaBoost [7]	WSCDDL [33]	BCDC
Source data	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
30	79.11%	79.98%	81.32%	84.37%	86.52%	**87.34**%
25	75.05%	75.06%	79.68%	81.46%	84.31%	**85.90**%
20	65.44%	67.40%	73.04%	79.72%	80.02%	**82.32**%
15	49.66%	54.12%	69.23%	75.53%	77.59%	**78.69**%
10	30.65%	46.28%	64.89%	72.87%	74.98%	**76.04**%

4.2 Action recognition

We extract the dense trajectories [23] as local features from raw action videos and project local features on a codebook using Locality Constrained Linear Coding (LLC) [24]. We run

[1] Since BCDC requires the target domain images share identical image categories as the source domain images, results are reported for the first 20 categories on the Caltech-101 dataset in this paper. On the other hand, results are reported for all image categories in [33].

Table 2: Performance comparison between the BCDC and state-of-the-art methods on the Caltech-101 dataset when the source domain data are only used by the BCDC.

Algorithm	ScSPM [26]	K-SVD [1]	LC-KSVD [10]	AdaBoost [9]	BCDC
Source data	*No*	*No*	*No*	*No*	*Yes*
30	85.36%	84.69%	85.60%	79.46%	**87.34**%
25	83.23%	82.16%	83.47%	74.83%	**85.90**%
20	80.11%	80.07%	80.59%	74.22%	**82.32**%
15	76.66%	74.82%	76.96%	71.91%	**78.69**%
10	72.87%	72.55%	72.37%	68.35%	**76.04**%

our method on three different partitions of the UCF YouTube dataset, where we randomly choose all action categories performed by the number of 5/9/16 actors as the training actions while using the remaining actions as the testing actions for each partition. 30 most relevant actions are chosen from each of the 7 source domain categories, and they are represented in the same manner as the target domain actions and coded with the same codebook. The same values of the weights α, β and K-SVD iterations are adopted as in the image classification task. Similarly, we compare the performance of BCDC with LLC, K-SVD, LC-KSVD, AdaBoost, TrAdaBoost and WSCDDL[2] in TABLE 3 and TABLE 4 when source domain data are included or not respectively. The reported results of LLC, K-SVD and LC-KSVD in TABLE 3 are obtained by treating the source domain data as extra training data without knowledge transfer. Again, the proposed BCDC method consistently outperforms the other methods. As expected, simply including source domain data without considering the data divergence degrades the performance of LLC, K-SVD and LC-KSVD in TABLE 4.

According to the obtained results on both image classification and action recognition tasks, the proposed BCDC method can effectively deal with the data distribution mismatch problem. It outperforms ScSPM and LLC by 22.07% and 6.41% in average respectively, and outperforms TrAdaBoost by 3.27% and 3.17% in average, on the Caltech-101 and the UCF YouTube datasets respectively when using the source domain data. Additionally, when using the transferred source domain data as auxiliary training samples, the BCDC method can improve the performance of the original ScSPM and LLC, which are free of the data mismatch problem, by 2.41% and 3.53% in average, which are significant improvements over the leading results.

Table 3: Performance comparison between the BCDC and state-of-the-art methods on the UCF YouTube dataset with source domain data.

Algorithm	LLC [24]	KSVD [1]	LC-KSVD [10]	TrAdaBoost [7]	WSCDDL [33]	BCDC
Source data	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
16	79.78%	75.43%	82.87%	82.40%	83.26%	**84.64**%
09	68.38%	64.54%	67.14%	69.20%	72.01%	**73.05**%
05	63.35%	59.35%	63.68%	65.46%	67.37%	**68.89**%

[2] For the same reason as stated in the above footnote, results are reported for the 7 selected action categories in this work, while results are reported for all action categories in [33].

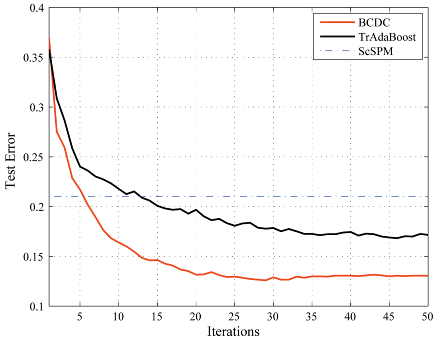

Figure 2: Error rate comparison of the proposed BCDC method with TrAdBoost and ScSPM on the Caltech-101 dataset.

Table 4: Performance comparison between the BCDC and state-of-the-art methods on the UCF YouTube dataset when the source domain data are only used by the BCDC.

Algorithm	LLC [24]	KSVD [1]	LC-KSVD [10]	AdaBoost [9]	BCDC
Source data	*No*	*No*	*No*	*No*	*No*
16	82.77%	74.57%	83.15%	79.40%	**84.64**%
09	68.38%	62.63%	69.82%	69.61%	**73.05**%
05	64.84%	59.37%	65.17%	65.52%	**68.89**%

5 Conclusion

In this paper we have presented a BCDC framework for categorising the target domain data using data from a different domain. In conjunction with a learned cross-domain dictionary pair, the proposed BCDC approach learns a set of boosted weak classifiers. Unlike the popular existing transfer learning techniques, the BCDC approach allows data adaptation through both feature representation level and classification level. Promising results are achieved on both image classification and action recognition, where knowledge from either the Web or a related dataset is transferred to standard benchmark datasets.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(1):4311–4322, 2006.

[2] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *CVPR*, 2011.

[3] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *CVPR*, 2010.

[4] Leo Breiman, JH Friedman, Olshen R. A, and Charles J Stone. Classification and regression trees. In *Wadsworth International Group*, 1984.

[5] Xianbin Cao, Zhong Wang, Pingkun Yan, and Xuelong Li. Transfer learning for pedestrian detection. *Neurocomputing*, 2012.

[6] Lin Chen, Li Wen, and Xu Dong. Recognizing rgb images by learning from rgb-d data. In *CVPR*, 2014.

[7] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *ICML*, 2007.

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[9] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55 (1):119–139, 1997.

[10] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, 2013.

[11] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[12] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[13] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[14] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[15] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[16] Julien Mairal, Marius Leordeanu, Francis Bach, Martial Hebert, and Jean Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *ECCV*, 2008.

[17] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Supervised dictionary learning. In *NIPS*, 2009.

[18] Anand Mishra, Karteek Alahari, CV Jawahar, et al. Image retrieval using textual cues. In *ICCV*, 2013.

[19] Carlos Orrite, Mario Rodríguez, and Miguel Montañés. One-sequence learning of human actions. In *Human Behavior Understanding*, pages 40–51. 2011.

[20] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009.

[21] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011.

[22] Ling Shao, Fan Zhu, and Xuelong Li. Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2014. doi: 10.1109/TNNLS.2014.2330900.

[23] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[24] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[25] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[26] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[27] Jianchao Yang, Kai Yu, and Thomas Huang. Supervised translation-invariant sparse coding. In *CVPR*, 2010.

[28] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.

[29] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.

[30] Wei Zhang, Akshat Surve, Xiaoli Fern, and Thomas Dietterich. Learning non-redundant codebooks for classifying complex objects. In *ICML*, 2009.

[31] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *NIPS*, 2003.

[32] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.

[33] Fan Zhu and Ling Shao. Weakly-supervised cross-domain dictionary learning for visual recognition. *International Journal of Computer Vision*, 109(1-2):42–59, 2014.