Visual Feature Learning

Fan Zhu

Department of Electronic and Electrical Engineering

University of Sheffield

Submitted to the department in partial fulfillment of the requirements for
Doctor of Philosophy

EEE

February, 2015

Declaration

Parts of this thesis have been taken from certain published/to be published journal/conference papers. All these papers were written primarily by me, Fan Zhu, during and as a result of my Ph.D. research. These papers are listed below:

1. **F. Zhu**, L. Shao and M. Yu, "Cross-Modality Submodular Dictionary Learning for Information Retrieval", ACM International Conference on Information and Knowledge Management, Shanghai, Nov. 2014. (Chapter 3)

2. **F. Zhu**, L. Shao and J. Tang, "Boosted Cross-Domain Categorization", British Machine Vision Conference, Nottingham, UK, Sep. 2014. (Chapter 3)

3. **F. Zhu** and L. Shao, "Weakly-Supervised Cross-Domain Dictionary Learning for Visual Recognition", International Journal of Computer Vision, vol. 109, no. 1-2, pp. 42-59, Aug. 2014. (Chapter 3)

4. **F. Zhu**, Z. Jiang and L. Shao, "Submodular Object Recognition", IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Jun. 2014. (Chapter 2)

5. L. Shao, **F. Zhu** and X. Li, "Transfer Learning for Visual Categorization: A Survey", Accepted by IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2014.2330900. (Chapter 1)

6. **F. Zhu**, L. Shao, "Correspondence-Free Dictionary Learning for Cross-View Action Recognition", International Conference on Pattern Recognition, Stockholm, Sweden, Aug. 2014. (Chapter 3)

7. **F. Zhu**, L. Shao, "Enhancing Action Recognition by Cross-Domain Dictionary Learning", British Machine Vision Conference, Bristol, UK, Sep. 2013. (Chapter 3)

8. **F. Zhu**, L. Shao and M. Lin, "Multi-View Action Recognition Using Local Similarity Random Forests and Sensor Fusion", Pattern Recognition Letters, vol. 34, no. 1, pp. 20-24, Jan. 2013. (Chapter 4)

Fan Zhu
February, 2015

Acknowledgements

I would like to express my deepest appreciation to a lot of people.

First of all, I would like to thank my parents. They have been providing their best love and maximum support to me in the past 27 years of my life.

I would like to thank my supervisor Prof. Ling Shao, who has been advising me from MSc to PhD, along all my four years life in Sheffield. Under his supervision, we are allowed to have maximum freedom to work on whichever research topics that we prefer in our field. Apart from providing us thorough advices and novel research ideas, he is also willing to discuss with us. He offered me such a unique opportunity to work and study in Sheffield, and introduced me into such a fantastic academic world.

I would like to thank the external examiner Prof. S.J. Maybank for his invaluable suggestions for the improvement of this thesis.

I would like to thank the internal examiner Dr. Mohammed Benaissa for his supports during the viva and the thesis editing process.

I would like to thank my fellow colleagues: Dr. Simon Jones, Dr. Ruomei Yan, Dr. Di Wu, Li Liu, Feng Zheng, Bo Dong, Mengyang Yu, Ziyun Cai, Yang Long, Redzuan Bin Abdul Manap, Peng Peng, Xu Dai, Yawen Huang, Danyang Wang, Zia Khan, Russ Drieberg, Zhizhong Yu and Tian Feng. Thank them for their generous help, their helpful discussions, and their contributions to the group. It has been my most valuable experience of working with them.

I would like to thank my co-author, who has also been advising me as my external mentor, Dr. Zhuolin Jiang. I learned a lot of things within a few months of time when working with him. I thank our academic visitors Dr. Hui Zhang, Dr. Jun Tang, Dr. Shoubiao Tan, Prof. Peng Li, Dr. Dabo Guo and Dr. Xuezhi Wen for their kind support and suggestions towards my future career. I would like to thank several staffs in our department, my second supervisor Prof. Jie Zhang, Dr. Mohammed Benaissa and Dr Charith Abhayaratne for their help and care through my MSc and PhD.

I would like to thank my friends, Lei Zeng, Huaxi Wang, Xingru Tao, Bo Peng, Haoyu Zhang, Jiaxin Xu, Tianwei Ye, Xun Yu, Dr. Ji Ni, Dr. Yilong Cao, Dr. Wenting Duan, Dr

Jing Li, Yi Shi, Yaqiong He, Yu Qin, Miaofeng Kang, Guangzong Li, Zhuo Zhou, Jingqu Zhang and Juntao Shi, for their support and care.

I specially thank Postgraduate Administrator Ms. Hilary J Levesley, our departmental IT Technician Mr. James Screaton and Mr. Steve Marsden for their kind help through my PhD time.

I also would like to thank the British Machine Vision Association (BMVA) and our departmental Learned Society Fund (LSF) for the financial support of my conference trips through my PhD.

Abstract

Categorization is a fundamental problem of many computer vision applications, e.g., image classification, pedestrian detection and face recognition. The robustness of a categorization system heavily relies on the quality of features, by which data are represented. The prior arts of feature extraction can be concluded in different levels, which, in a bottom up order, are low level features (e.g., pixels and gradients) and middle/high-level features (e.g., the BoW model and sparse coding). Low level features can be directly extracted from images or videos, while middle/high-level features are constructed upon low-level features, and are designed to enhance the capability of categorization systems based on different considerations (e.g., guaranteeing the domain-invariance and improving the discriminative power). This thesis focuses on the study of visual feature learning. Challenges that remain in designing visual features lie in intra-class variation, occlusions, illumination and view-point changes and insufficient prior knowledge. To address these challenges, I present several visual feature learning methods, where these methods cover the following sub-topics: (i) I start by introducing a segmentation-based object recognition system. (ii) When training data are insufficient, I seek data from other resources, which include images or videos in a different domain, actions captured from a different viewpoint and information in a different media form. In order to appropriately transfer such resources into the target categorization system, four transfer learning-based feature learning methods are presented in this section, where both cross-view, cross-domain and cross-modality scenarios are addressed accordingly. (iii) Finally, I present a random-forest based feature fusion method for multi-view action recognition.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review[1]

1.1 Visual Categorization

In the past few years, along with the explosion of online image and video data (Flickr [2], YouTube [3]), the computer vision community has witnessed a significant amount of applications in content-based image/video search and retrieval, human-computer interaction, sport events analysis, etc. These applications are built upon the development of several aspects of classical computer vision tasks, such as human action recognition, object localization and image classification, which, however, remain challenging in real-world scenarios due to cluttered backgrounds, view point changes, occlusions, and geometric and photometric variations of the target [145], [172], [157], [63], [70], [156], [31], [102]. The existence of these issues can break the data smoothness[4], so that general representations (e.g., the Bag-of-Words (BoW) model) become less discriminative. Many previous methods that manage to deal with these issues are proposed. Zhang et al. [180] proposed the construction of a continuous virtual path between different views to solve the view point changing problem. Carreira et al. [16] formulated object recognition as a segmentation-based regression problem, so that cluttered background areas can be removed from foreground objects. Both stated frameworks output proper high level features for specific categorization tasks, where

[1] Part of this chapter will be published at:
L. Shao, **F. Zhu** and X. Li, Transfer Learning for Visual Categorization: A Survey, accepted by IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2014.2330900.

[2] http://www.flickr.com/

[3] http://www.yoututbe.com/

[4] In this report, data smoothness denotes the property that data points, which are close to each other in the feature space, are likely to share the same label.

these features contain either view-invariant information or less background noise. In fact, the study of features is an important topic in computer vision. In a bottom up order, the types of features include low-level features (e.g., pixels and gradients) and middle/high-level features (e.g., the BoW model and sparse coding). Features can also be classified as hand-crafted features and deep learning features. Recent advanced deep learning techniques (e.g., Convolutional Neural Network (CNN) [74]) are proved to be effective on many tasks. For dealing with image classification tasks, the majority deep learning approaches take raw pixels of resized images as inputs, and output high level features or the posterior probability of each class, where parameters of each hidden layer are learned as in a black box. Though learning high level features through such a abrupt manner, deep learning models achieve leading performance in many fields. Instead of abruptly learning on the pixel level, hand-crafted features are well-designed for specific purposes. In this report, only middle/high level hand-crafted features are discussed.

1.2 Transfer Learning

Regular machine learning approaches [126], [146], [124], [125], [124], [135], [133], [185] have achieved promising results under the major assumption that the training and testing data are in the same feature space. However, in real-world applications, due to the high price of human manual labeling and environmental restrictions, obtaining the training data which satisfy the above requirement is not always possible. Typical examples are [15], [111], [162], where only one action template is provided for each action class for training, and [88], where training samples are captured from a different viewpoint. In such situations, regular machine learning techniques are very likely to fail. This reminds us of the capability of the human vision system. Given the gigantic geometric and intra-class variabilities of objects, humans are able to learn tens of thousands of visual categories in their life, which leads to the hypothesis that humans achieve such a capability by accumulated information and knowledge [36]. It is estimated that there are about $10 \sim 30$ thousands object classes in the world [7] and children can learn $4 \sim 5$ object classes per day [36]. Due to the limitation of objects that a child can see within a day, learning new object classes from large amounts of corresponding object data is not possible. Thus, it is believed that the existing knowledge gained from previous known objects assists the new learning process through their connections with the new object categories. For example, assuming we did not know what a water melon is, we would only need one training sample of water melons

together with our previous knowledge on melons-circular shapes, the green color, etc., to remember the new object category water melon. The motivation behind transfer learning is that data in other related domains can be utilized for learning a new subject when data that directly describe such a subject are insufficient. Normally, two types of domains are defined in transfer learning tasks, the target domain and the source domain. The target domain is also the domain of interest, and it normally contains very few or even no labeled data, while the source domain contains a large amount of labeled data. A typical example of transfer learning can be found in [13], where Cao et al. built the target domain with the combination of Microsoft Research Action dataset[5] and the TRECVID surveillance dataset [25], which contain very few annotations and large amounts of noise, and the source domain with the KTH dataset [129], which has a clean background and limited viewpoint and scale changes.

Transfer learning can be considered as a special learning paradigm where partial/all training data used are under a different distribution than the testing data. To understand the significance of knowledge transfer in terms of visual learning problems, the literature (e.g., [113], [46], [21]) has concluded three general issues regarding the transfer process: 1) when to transfer; 2) what to transfer; 3) how to transfer. Firstly, "when to transfer" includes the issues whether transfer learning is necessary for specific learning tasks and whether the source domain data are related to the target domain data. In the scenarios of [156], [171], where training samples are sufficient and impressive performance can be achieved while being constrained in the target domains, including another domain as the source domain becomes superfluous. The divergence across different pairs of source domain and target domain data can be significantly different. For example, when the difference between the target domain and the source domain is only caused by the existence of a small amount of noise, a low divergence level holds; on the other hand, the domain data can be large if the two domains are completely irrelevant (e.g., text data vs. image data). Thus, a brute force transfer of knowledge form from the source domain into the target domain irrespective of their divergence would certainly cause performance degeneration, or, in even worse cases, it would break the original data consistency in the target domain. Secondly, the answer to "what to transfer" can be concluded in three aspects: 1) inductive transfer learning, where all the source domain instances and their corresponding labels are used for knowledge transfer; 2) instance transfer learning, where only the source domain instances are used; 3) parameter transfer learning: in addition to the source domain instances and labels, some parameters of pre-learned models from the source domain are utilized to help improve the performance in

[5] http://research.microsoft.com/∼zliu/ActionRecoRsrc

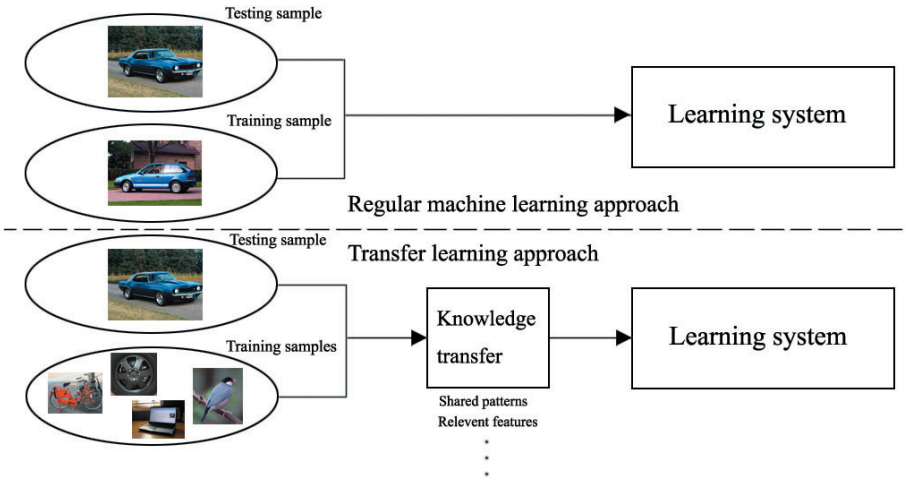


Fig. 1.1 Basic frameworks of traditional machine learning approaches and knowledge transfer approaches. For regular machine learning approaches, the learning system can only handle the situation that testing samples and training samples are sampled from the same distribution. On the other hand, transfer learning approaches have to deal with the data distribution mismatch problem through specific knowledge transfer methods, e.g., mining the shared patterns from data across different domains.

the target domain. Thirdly, "how to transfer" includes all the specific transfer learning techniques, and it's also the most important part that has been studied in the transfer learning literature. Many transfer learning techniques have been proposed, e.g., in [154], [83], [26], where knowledge transfer is based on the Nonnegative Matrix Tri-factorization (NMTF) framework, and in [112], where the transfer learning phase is via dimension reduction. We illustrate the basic frameworks of traditional machine learning approaches and knowledge transfer approaches in Fig. 1.1. For traditional machine learning approaches, the ideal choice of the training set to predict a testing instance 'car' should contain 'cars'. However, in the case of knowledge transfer, the training set can just contain some relevant categories rather than 'cars', e.g., 'wheels', which are similar to the 'wheels' of 'cars'; 'bicycles', which share the knowledge of 'wheels' with the 'car wheels', or even some irrelevant objects, e.g., 'laptops' and 'birds', which seem to have no connections with 'cars' but actually share certain edges or geometrical layouts with local parts of a 'car' image.

As the age of "Big Data" has come, transfer learning can provide more benefits to solve the target problem with more relevant data. Thus, it is believed that more applications on transfer learning will emerge in future research. In this report, transfer learning problems are addressed with several dictionary learning approaches. Along with introducing these approaches, I review existing transfer learning techniques, and discuss how they can be ap-

plied to visual categorization tasks. Some visual characteristics (e.g., appearance, shape, local symmetries and structural information) are considered when designing transfer learning models.

1.3 Dictionary Learning

Dictionary learning for sparse representation has attracted much attention. It has been successfully applied to a variety of computer vision tasks, e.g., face recognition [161] and image denoising [183]. Using an over-complete dictionary, sparse modeling of signals can approximate the input signal by a sparse linear combination of items from the dictionary. Many algorithms [79], [68], [161] have been proposed to learn such a dictionary according to different criteria. The K-Singular Value Decomposition (K-SVD) algorithm [1] is a classical dictionary learning algorithm that generalizes the K-means clustering process for adapting dictionaries to efficiently learn an over-complete dictionary from a set of training signals. The K-SVD method focuses on the reconstructive ability, however, since the learning process is unsupervised, the discriminative capability is not taken into consideration. Consequently, methods that incorporate the discriminative criteria into dictionary learning were proposed in [177], [169], [101], [97], [101], [10]. In addition to the discriminative capability of the learned dictionary, other criteria designed on top of the prototype dictionary learning objective function include multiple dictionary learning [178], category-specific dictionary learning [170], etc. Different from most dictionary learning methods, which learn the dictionary and the classifier separately, the authors of [177] and [66] unified these two learning procedures into a single supervised optimization problem and learned a discriminative dictionary and the corresponding classifier simultaneously. Taking a step further, Qiu et al. [117] and Zheng et al. [179] designed dictionaries for situations in which the present training instances are different from the testing instances. The former presented a general joint optimization function that transforms a dictionary learned from one domain to the other, and applied such a framework to applications such as pose alignment, pose and illumination estimation and face recognition. The latter achieved promising results on the cross-view action recognition problem with pairwise dictionaries constructed using correspondences between the target view and the source view. To make use of some data that may not be relevant to the target domain data, Raina et al. [120] proposed a method that applies sparse coding to unlabeled data to break the large amount of data in the source domain into basic patterns (e.g., edges in the task of image classification) so that knowledge can be

transferred from the bottom level to a high level representation.

In the following part of this subsection, we give more detailed introduction of dictionary learning. Consider a data sample $y \in \mathbb{R}^n$, the coefficient vector $x \in \mathbb{R}^N$ and a projection matrix $D \in \mathbb{R}^{n \times N}$, and suppose that the sample can be reconstructed by the linear transformation of the coefficient vector through the projection matrix plus the reconstruction error as:

$$y = Dx + \textit{reconstruction error}. \tag{1.1}$$

If we define an objective function

$$\mathscr{F}(x) = \|y - Dx\|_2^2, \tag{1.2}$$

the vector x can be estimated by minimizing $\mathscr{F}(x)$ subject to appropriate constraints. If $N > n$, the solution to the unconstrained optimization problem in Equation (1.2) is not unique, thus it leads to the over-fitting problem. In order to give more discriminative solutions when estimating x, additional constraints on x are necessary. The commonly used constraints include regularizing with l_0-norm, l_1-norm and l_2-norm, where the first two are also known as Sparsity-inducing norms and the last one is also known as the Euclidean or Hilbertian norm. In general, l_2-norm has well developed theory and algorithms, and it has been applied to non-linear predictors, non-parametric supervised learning and kernel methods. On the other hand, the developing Sparsity-inducing norms attract more attention recently. Applications that can benefit from the sparsity-inducing compression, regularization in inverse problems, etc. The l_0-norm, which indicates the solution with fewest non-zero entries, is applied in our case. Thus Equation (1.2) can be formulated as:

$$\mathscr{F}(x) = \|y - Dx\|_2^2, s.t. \forall i, \|x\|_0 \leq T, \tag{1.3}$$

T is a sparsity constraint factor that limits the number of non-zero elements in the sparse codes, so that the number of non-zero components of x is less than T. In the case that $T = 1$, i.e., each instance is represented by a single basis in the codebook, Equation (1.3) is equivalent to Vector Quantization (VQ). The construction of the dictionary D is achieved through the K-SVD [1] algorithm, which iteratively minimizes the reconstruction error and learns a reconstructive dictionary for sparse representations. Given D, the computation of sparse code x is generally NP-hard under the sparsity constraint, thus one has to seek alternative methods to approximate the solution, e.g., the greedy algorithms Matching Pursuit (MP)

[99] and Orthogonal Matching Pursuit (OMP) [114], which sequentially select the dictionary atoms, and the Basis Pursuit (BP) [19], which suggests a convexisation by relaxing the l_0-norm to the l_1-norm. More details on optimizing the objective function under the l_0-norm constraint are given in Section 3.3. Since l_1-norm also leads to sparse solutions, an alternative formulation for our problem in Equation (1.3) is to replace l_0-norm regularization with l_1-norm regularization to enforce sparsity:

$$\mathscr{F}(x) = \|y - Dx\|_2^2, s.t. \|x\|_1 \leq T, \tag{1.4}$$

where the optimization problem is convex in D with x fixed, and convex in x with D fixed, but not in both simultaneously. Again, $F(x)$ can be minimized iteratively by alternatingly optimizing D or sparse code x while fixing the other. When dictionary D is fixed, the optimization problem is equivalent to a linear regression problem with l_1-norm regularization on the coefficients, which can be solved with the feature-sign search algorithm [79]. When sparse code x is fixed, the problem reduces to a least square problem with quadratic constraints, so that it can be solved by the Lagrange dual as in [79].

1.4 Optimization

In practise, many optimization problems are solved under the assumption that these problems can be modeled by continuous convex or concave functions, however, not all problems are of this type. On the other hand, submodularity offers another option for solving the optimization problem in a discrete manner. In mathematics, a submodular function is a set function which has the diminishing returns property (i.e., a set function whose value has the property that the difference a single element makes when included in an input set decreases as the size of the input set increases). Submodularity has many applications in economics, game theory, etc. Andreas and Daniel [73] summarized its usage and applications particularly on maximization problems. It is worth noting that different from the relationship between convex and concave functions, minimizing and maximizing a submodular set function require completely different methods.

Submodularity has recently been applied to many computer vision tasks, including clustering [91] and segmentation [65]. Liu et al. [91] presented a method that uses the entropy rate of a random walk on a graph for compact and homogeneous clustering. Jiang and Davis [65] solved a facility location problem [47, 78] for salient region detection. The saliency

of a region is modeled in terms of its appearance and spatial location, and salient region detection is achieved by maximizing a submodular objective function.

1.5 Datasets

1.5.1 Multi-View IXMAS

INRIA Xmas Motion Acquisition Sequences (IXMAS) [159] is a multi-view dataset for view-invariant human action recognition. 13 daily-live motions (in practise, only 11 actions, including 'check watch', 'cross arms', 'scratch head', 'sit down',' get up', 'turn around', 'walk', 'wave', 'kick' and 'pick up', are used for evaluation for state-of-the-art works) performed each 3 times by 9 actors in 5 camera positions. In this report, the IXMAS dataset is used for evaluating the multi-view camera fusion action recognition and cross-view action recognition algorithms. These two tasks induce two experimental settings, where the former utilizes actions captured by a combination set of cameras for training and the same set of cameras for testing, while the latter utilizes training and testing actions which are captured from different cameras.

1.5.2 UCF YouTube

The UCF YouTube dataset [86] (shown in Fig. 1.2) is a realistic dataset that contains camera shaking, cluttered background, variations in actors' scale, variations in illumination and view point changes. There are 11 actions including cycling, diving, golf swinging, soccer juggling, jumping, horse-back riding, basketball shooting, volleyball spiking, swinging, tennis swinging and walking with a dog. These actions are performed by 25 actors.

1.5.3 HMDB51

The HMDB51 dataset (shown in Fig. 1.3) contains video sequences which are extracted from commercial movies as well as YouTube. It represents a fine multifariousness of light conditions, situations and surroundings in which actions can appear, different recording camera types and viewpoint changes.

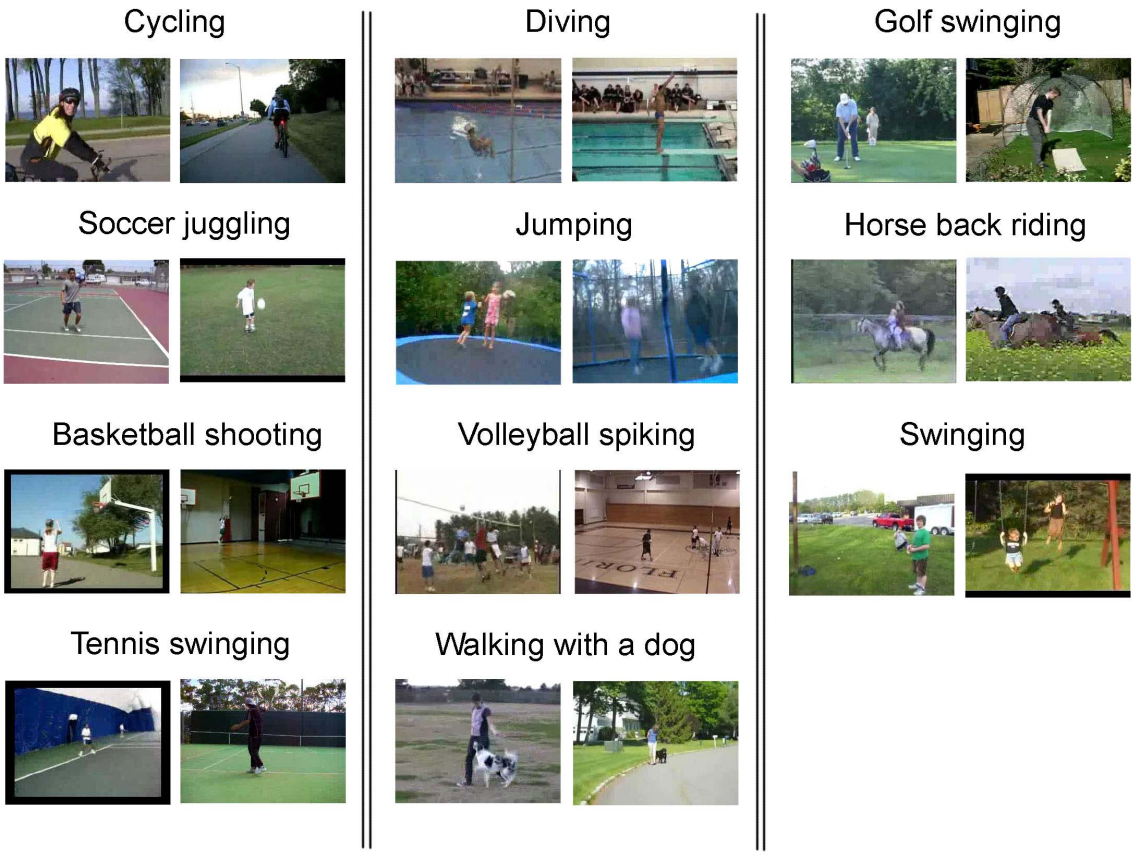


Fig. 1.2 Example images from video sequences in the UCF YouTube dataset.

Fig. 1.3 Example images from video sequences in the selected body movements of the HMDB51 dataset.

1.5.4 Caltech101/256

The Caltech101 image dataset (shown in Fig. 1.4) consists of 101 categories (e.g., accordion, cannon, and chair), and each category contains 30 to 800 images. The Caltech 256 dataset (shown in Fig. 1.5) contains $30,607$ images of 256 categories. Due to the existence of large variations on object location, pose, and size, and also the increased number of categories, the Caltech256 dataset is considered as a more challenging dataset than the Caltech101 dataset.

1.5.5 PASCAL VOC

The PASCAL VOC dataset is created along with the PASCAL VOC Challenge [34]. The VOC challenge ran between 2005-2012, since there are in total 8 releases of the PASCAL VOC dataset. We choose to evaluate on PASCAL VOC 2007, as this is the latest release that provides the ground-truth of the testing data.

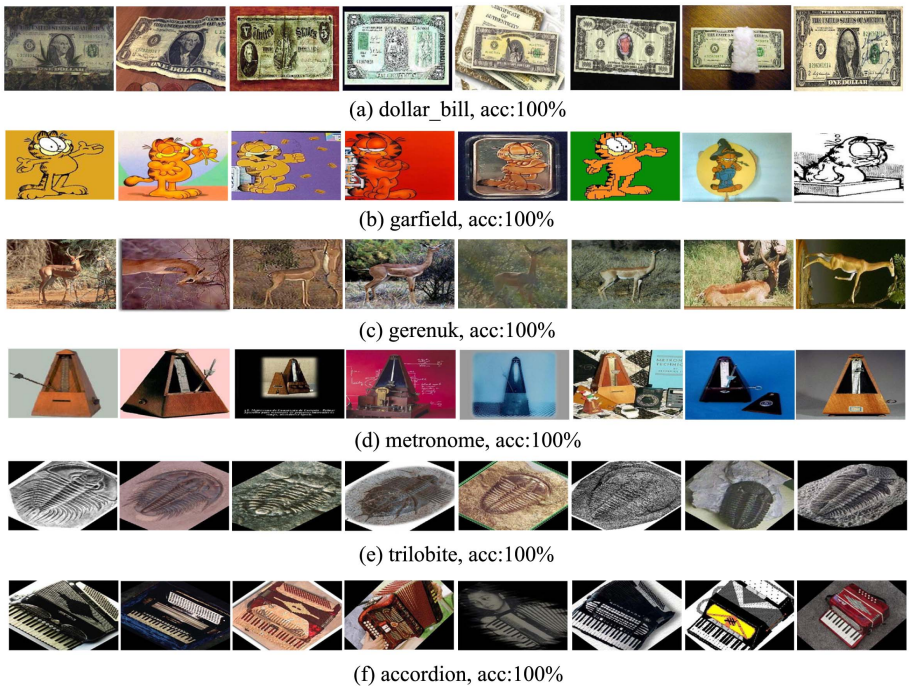


Fig. 1.4 Example images from classes from the Caltech101 dataset.

Fig. 1.5 Example images of the categories from the Caltech256 dataset.

1.5.6 ETHZ-Shape

The ETHZ-Shape dataset [41] contains 255 test images and features five diverse shape-based classes, including ‘apple’ ‘logos’, ‘bottles’, ‘giraffes’, ‘mugs’, and ‘swans’. The ground-truth of both training and testing data are provided.

Chapter 2

Segmentation-based image feature learning[1]

2.1 Preliminaries

Submodularity: Let $\mathscr{V}$ be a finite set, $\mathscr{A} \subseteq S \subseteq \mathscr{V}$ and $a \in \mathscr{V} \backslash S$. A set function $F : 2^{\mathscr{V}} \rightarrow \mathbb{R}$ is submodular if $F(\mathscr{A} \cup a) - F(\mathscr{A}) \geq F(S \cup a) - F(S)$. This property is referred to as diminishing returns, stating that adding an element to a smaller set produces a greater change than a larger set[107].

Monotone submodular functions: A submodular function f is monotone if we have $f(\mathscr{A}) \leq f(S)$ for any $\mathscr{A} \subset S$.

Facility location problem: The facility location problem is an optimal facility placement problem. The optimal solution should maximize the benefits that placed facilities bring to an area while considering issues like hazardous materials, price of opening a facility, etc. An simple example is given in Fig. 2.1. Suppose we wish to place "facilities" 1 and 2 at some locations within the rectangular, and both "facilities" provide services to customers $1-4$. Let w_{ij} (value given in Fig. 2.1) denote the benefit that a facility at location j brings to customer i, and w_{ij} is inversely proportional to the customer-facility distance. Let ϕ denote the cost of placing each facility and S denote the set of placed facilities. Assuming each customer only chooses the facility with highest benefit (i.e., lowest distance in the ex-

[1]The original content of this chapter is published at:
F. Zhu, Z. Jiang and L. Shao, Submodular Object Recognition, IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Jun. 2014.

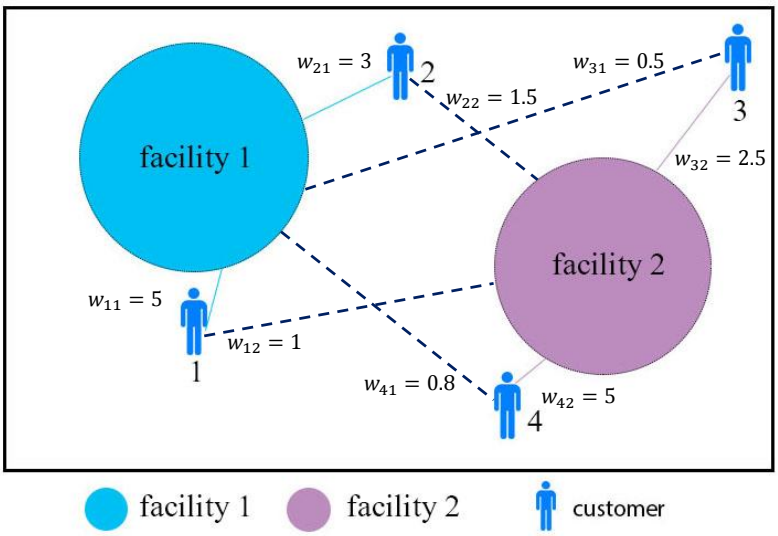


Fig. 2.1 Illustration of the facility location problem. Two facilities 1 and 2 are placed within a rectangle area to provide services to customer $1-4$. Each customer is considered to choose the facility with highest benefit, and the choice of each customer is denoted with a solid line.

ample), the total benefits of building such two facilities within the area is modeled by the set function

$$\begin{aligned} f(S) &= \sum_{i=1}^{4} \max_{j\in\{1,2\}} w_{i,j} - 2\phi \\ &= w_{11} + w_{21} + w_{32} + w_{42} - 2\phi. \end{aligned} \tag{2.1}$$

If all benefit values are non-negative ($w_{ij} > 0$), $f(S)$ is monotone submodular[44]. In our work below, instead of a benefit value, w_{ij} refers to the pairwise relationship between a "customers" segment and a "facility" segment.

2.2 Motivation and Introduction

In recent years, the bag-of-features (BoF) model and its extension, spatial pyramid matching (SPM) [77], have been popular for object recognition. When working with densely sampled pyramid grids and powerful classifiers, BoF and SPM have achieved impressive performance on several object recognition benchmarks including PASCAL VOC 2007 [34]

and Caltech-101 [37]. While these densely sampled grids can retain context information, such as spatial layout, for a specific object category, irrelevant background information is also included. To solve this problem, a lot of efforts have been spent on utilizing image segmentation results for improved recognition performance. Examples of image segmentation results are given in Fig. 2.3 and the difference between our proposed object recognition framework and regular object recognition framework is illustrated in Fig. 2.2. The benefits are two fold: (1) accurate segmentation can enhance the contrast of object boundaries, so that features along the boundaries are more shape-informative; (2) computing features on homogeneous segments improves the signal-to-noise ratio. However, little progress has been achieved due to a lack of reliable segmentation techniques. For example, Nilsback and Zisserman [109] employed segmented images for flower classification. Since only a single segment is considered for an image, and clean segmentations can only be guaranteed for images with simple backgrounds, the performance improvement is not significant when comparing with results from non-segmented images. Unlike approaches that consider only one segment in an image [109], our approach considers multiple segments simultaneously via submodularity. Our approach is based on the recently proposed Constrained Parametric Min Cuts (CPMC) [17] algorithm, which has demonstrated a significant improvement in segmentation. We present a submodular objective function for efficiently selecting salient segments from the set of figure-ground hypotheses for object recognition, and another objective function that additionally considers the discriminative information of segments for the selection. We learn a scoring (regression) function for each object category with the overlapping observations of each pair of the figure-ground hypothesis and the ground-truth segment. The benefit of regression is exploited for discriminating segments' categories and qualities. Our objective function contains a facility-location term and a discriminative term, where the facility-location term measures the similarities between the selected segments and their group elements and the "facility" costs for the selected elements, and the discriminative term is measured by the consistency of categories that obtain the maximum regression values on selected segments. Our main contributions are three-fold:

- ⋆ Object recognition is modeled as a facility location problem with the constraint of segments' class consistency of selected segments (facility locations), which can be solved by maximizing a submodular function. We provide a new perspective by applying submodularity to the object recognition problem.

- ⋆ By adding an extra discriminative term to the objective function, though the solution is ad hoc, we further improve the recognition performance.

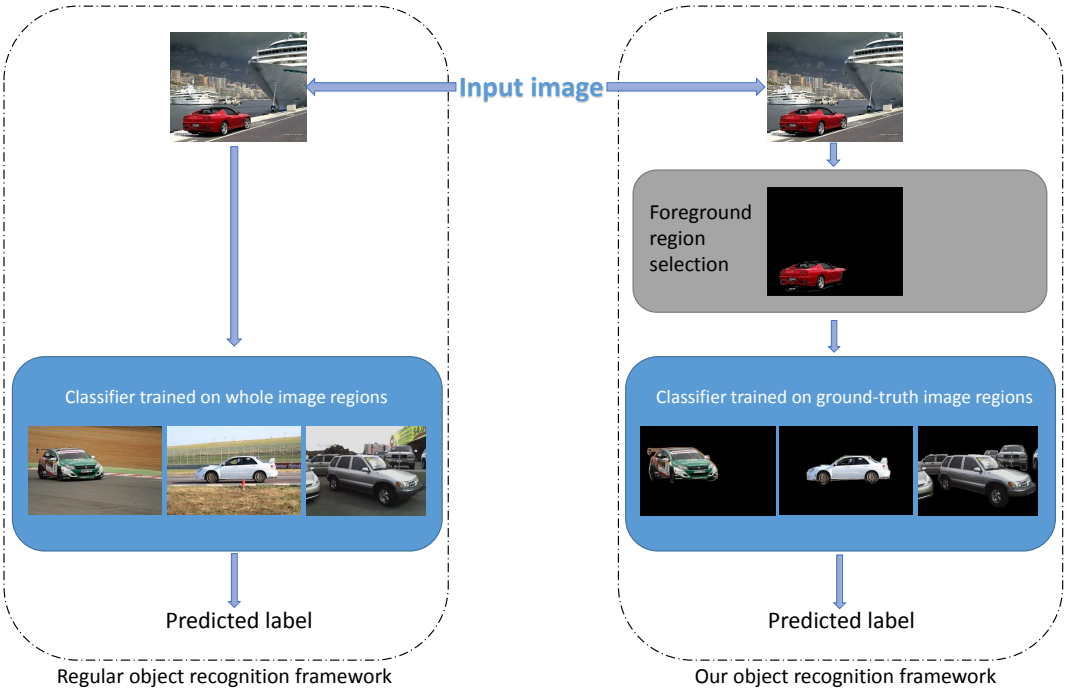


Fig. 2.2 Illustration of the difference between our proposed object recognition framework (right) and the regular object recognition framework (left). The foreground region selection part (within the gray region) is the focus of this work.

⋆ Our submodular recognition approach achieves state-of-the-art performance on three popular object recognition benchmarks.

2.3 Related Work

Many recent bottom-up object recognition approaches attempt to use the spatial layouts of objects for better performance. He et al. [57] constructed a Conditional Random Field (CRF) framework on image pixels, where each pixel is assigned to one of a finite set of labels. Both image features and image labels are incorporated into the probabilistic framework. Shotton et al. [141] proposed Textonboost, which incorporates texture, layout and context information for unary classification. By incorporating the unary classifier into a CRF, the spatial interactions between class labels of neighboring pixels are captured to guarantee the smoothness. A major limitation of pixel-level methods is their weak capability for segmenting nearby objects of the same category. Gould et al. [53] and Ladicky et al. [76] addressed such a limitation using rectangular bounding box detection constraints. Rather

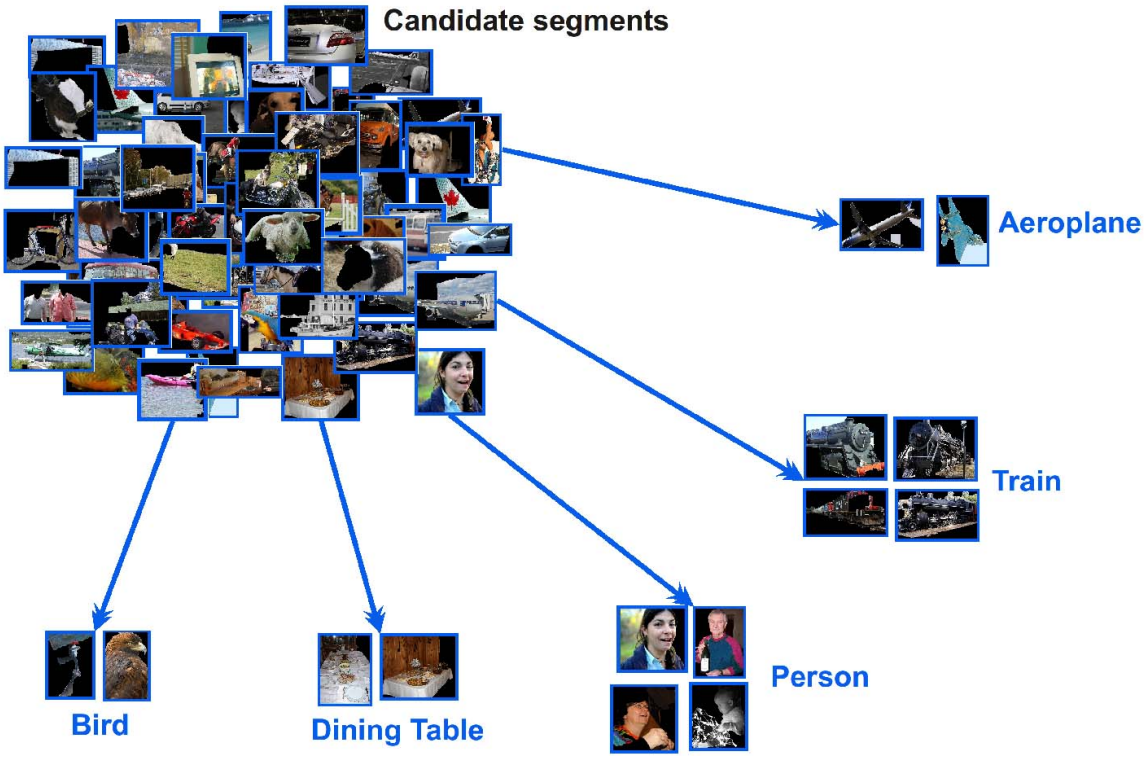


Fig. 2.3 The pool of image segments.

than using bounding boxes, segment-based or superpixel-based approaches are closer to the ground-truth spatial support. Fulkerson et al. [45] used superpixels as basic units in the recognition framework. To this end, the histogram of local features within each superpixel is used to construct a classifier, which is regularized by aggregating histograms of neighboring superpixels. For segment-level recognition methods, Rabinovich et al. [119] applied a stability heuristic to select a reduced list of segmentations obtained from normalized cuts [140]. For an image I, each segment in the list is regarded as a stand-alone image, and labels from all segments are used to vote for the category of image I. By using a collection of segments for recognition rather than a single segment, more object boundary information can be captured. However, they do not provide a reliable segment selection mechanism for filtering out erroneous segments, and treating the whole collection of segments as a new set of images is too computationally expensive. Li et al. [81] presented an object recognition framework based on multiple figure-ground segmentations generated by CPMC, which is the most similar approach to our work.

2.4 Segmentation-based object recognition

Our method solves the object recognition problem through the selection of a subset of segments so as to best discover the target object in a query image. Firstly, we apply the CPMC segmentation [17] on each image to produce a set of figure-ground hypotheses in an unsupervised manner. Then we construct a graph G based on the generated figure-ground hypotheses. Since using all segment hypotheses is too computationally expensive and probably produces misleading predictions, we aim to discover the representative and visually salient segment subset $\mathscr{A}$ of S by iteratively adding elements of S into $\mathscr{A}$. Object masks are obtained by overlaying selected segments for extracting foreground objects. Finally, a linear classifier is applied for recognizing objects.

2.4.1 Graph-Construction

For an image I (in gray-scale), N segments $\mathscr{V} = \{v_1, v_2, \cdots, v_N\}$[2] are generated by CPMC and the ground-truth segment G_I^k of object category k is provided for the training data. Note that each segment v_i here is in the form of a regular gray-scale image, which has the same size as the image I. Only a region within v_i has identical pixel values as I, while the remaining pixels have the value 0. A subset of segments is shown in Fig. 2.1(b)$\sim$(f). We construct a graph $G = (\mathscr{V}, E)$ on the segment hypotheses in image I, where the vertices $v \in \mathscr{V}$ are segments while $e \in E$ are edges between segments. The weight w_{ij} assigned to the edge e_{ij} is computed using Equation 2.3.

2.4.2 Salient Segment Selection based on Submodularity

Objective Function

In order to select visually salient image regions from backgrounds, we model the segment selection and recognition as a facility location problem [47, 78]. We consider that a foreground region consists a set of image segments, which are computed by the CMPC segmentation technique, so that our task becomes that of selecting the set of visually salient segments. We consider each selection task to be equivalent to placing a "facility" in an area, and we define the number of selected segments as no more than K. Let $N_{\mathscr{A}}$ denote the number of selected segments (i.e., placed "facilities") and $H(\mathscr{A})$ denote the benefit value when

[2] N is constrained within 100 to limit the computational cost in our work.

the segment set $\mathscr{A} \in \mathscr{V}$ is selected, the combinatorial formulation of the segment selection problem can be applied:

$$\begin{aligned} \max_{\mathscr{A}} H(\mathscr{A}) &= \sum_{i \in \mathscr{V}} \max_{j \in \mathscr{A}} w_{ij} - \sum_{j \in \mathscr{A}} \phi_j \\ s.t. \quad & \mathscr{A} \subseteq \mathscr{V}, N_{\mathscr{A}} \leq K \end{aligned} \tag{2.2}$$

where w_{ij} denotes the weight between a unselected segment ("customer") v_i and a selected segment ("facility") v_j (considered as facilities), and the cost ϕ_j of selecting a segment into $\mathscr{A}$ is fixed to δ. Submodularity of the objective function H has been proved in [47, 78].

The first term in (2.2) encourages the "customer" segment v_i has the largest weight with its assigned "facility" segment (i.e., v_j). It favors the selected segment v_j ("facility") to well represent or be similar to the "customer" segments it associates to, so that the final selected set $\mathscr{A}$ can cover a visually salient region [65]. Let x_i denote the Spatial Pyramid Matching (SPM) [168] feature computed from v_i, then the weight w_{ij} is computed as:

$$w_{ij} = K(x_i, x_j) + O(v_i, v_j), \tag{2.3}$$

where $K(x_i, x_j)$ denotes the chi-squared distance on histogram features of any pair of segments:

$$K(x_i, x_j) = \sum_{i \in \mathscr{V}, j \in \mathscr{V}} \sum_{n=1}^{N_d} \frac{(x_i^n - x_j^n)^2}{2(x_i^n + x_j^n)}, \tag{2.4}$$

where N_d is the dimension of x_i, and $O(v_i, v_j)$ denotes the 'intersection-over-union' overlap measurement of the same pair of segments:

$$O(v_i, v_j) = \frac{|v_i \bigcap v_j|}{|v_i \bigcup v_j|}. \tag{2.5}$$

If w_{ij} is computed only based on the overlap measurement, the facility location term will pursue segments that have highest overlap values with neighbouring segments, so that segments with large background coverage are preferably selected. Including the chi-squared distance on segments' histogram features can effectively avoid such a problem. The second term in (2.2) penalizes on extraneous facilities. When the gain obtained by introducing a new segment to the $\mathscr{A}$ is offset by the cost of selecting a new segment (i.e., "placing a new facility"), $\mathscr{A}$ will stop growing. Hence this selected $\mathscr{A}$ is representative and visually salient.

Algorithm 1 Submodular Salient Segment Selection

Input: $\mathscr{V}$, $N_{\mathscr{A}}$ and K.
Output: $\mathscr{A}$, Q, L.
Initialization: $\mathscr{A} = \emptyset$, $L =$ a zero matrix, $N_{\mathscr{A}} = 0$.
loop
    if $N_{\mathscr{A}} > K$ **then**
        break;
    end if
        $$v_i^* = \arg\max_{v_i \in \mathscr{V} \setminus \mathscr{A}} H(\mathscr{A} \cup \{v_i\}) - H(\mathscr{A})$$
    if $H(\mathscr{A} \cup \{v_i^*\}) \leq H(\mathscr{A})$
        break;
    end if
    $\mathscr{A} = \mathscr{A} \cup \{v_i^*\}$
    $L = L + v_i^*$
    $N_{\mathscr{A}} = N_{\mathscr{A}} + 1$
end loop
Obtain the final mask Q

Optimization

Direct maximization of $H(\mathscr{A})$ is an NP-hard problem [47]. The monotonicity and submodularity of (2.2) has been proved in [47, 78]. Utilizing this property, (2.2) can be efficiently solved with a greedy-based approximate solution [47] [107]. The segment set $\mathscr{A}$ is initialized with $\emptyset$, and a segment $v_i \in \mathscr{A} \setminus S$ that leads to the largest marginal gain $H(\mathscr{A} \bigcup v_i) - H(\mathscr{A})$ at each iteration is iteratively added to $\mathscr{A}$. $\mathscr{A}$ stops absorbing new segments when the desired number of segments is reached or the gain decreases. Given a finite segment set S and its subset $\mathscr{A} \subseteq S$, a simple uniform matroid is induced towards the number of selected segments $N_{\mathscr{A}}$, which is less than K. Maximization of a submodular function with a uniform matroid constraint yields a $(1 - 1/e)$-approximation [107], Hence our approach provides a performance-guarantee solution. The pseudocode of submodular salient segment selection framework is given in Algorithm 1, where K denotes the limit of segment number and $N_{\mathscr{A}}$ denotes the number of selected segments in $\mathscr{A}$.

The optimization process can be accelerated by using the submodularity property of the objective function. Instead of recomputing the gain for adding every segment $v_i \in \mathscr{V} \setminus \mathscr{A}$, which requires $|\mathscr{V}| - |\mathscr{A}|$ evaluations for the gain $H(\mathscr{A})$, we use the lazy evaluation form from [80].

2.4.3 Salient and Discriminative Segment Selection

While the salient segment selection method introduced in Section 2.4.2 is an unsupervised selection method, we include a discriminative term which can discriminate segments' category information into the objective function for selecting both representative and discriminative segments. Based on the regressor trained for each category, the category label of each segment can be estimated. Our intention is to force the majority of selected segments of a query image are assigned the same category label.

Discriminative Term

We enforce a consistency of segments' labels to boost the discriminative power of the selected set $\mathscr{A}$. The discriminative term is based on the learned segment regressor for each category (e.g., cars or planes). More descriptions on how these regressors are trained are give below.

These segments $V = \{v_1, v_2, \cdots, v_N\}$ are represented by the spatial pyramid descriptors [168] $x = \{x_1, x_2, \cdots, x_N\}$. The object category contained in image I is $k \in \{1, 2, \cdots, m\}$, and we need to learn m scoring functions $f_1(x), f_2(x), \cdots, f_m(x)$ for each object category. Each function is defined on the score set O, which is computed by the overlaps between a segment v_i and the ground-truth segment (a matrix which has binary scores, where the values 1 correspond to the foreground regions of image I and the values 0 correspond to the background region) G_I^k of category k in an image I using the 'intersection-over-union' measurement. Specifically, each O_i is computed as:

$$O_i(v_i, G_I^k) = \frac{|v_i \bigcap G_I^k|}{|v_i \bigcup G_I^k|}. \tag{2.6}$$

Thus, each segment v_i is associated with its regression score $O_i(v_i, G_I^k)$. Since a segment usually overlaps with more than one ground-truth segments when training each $f_k(x)$, it can have different regression values for different categories when training the scoring function of different categories. If category k does not appear in image I, all the segments in image I are considered as having no overlap with category k. A simple linear Support Vector Regression is applied to learn each scoring function $f_k(x)$ by regressing on the score set $\{O\}$ against S for all images in the training set[3]. During testing, the scoring function which

[3]The regressors are first trained only using the ground-truth segments, after which all candidate segments are fed into the regressors for classification. The miss-classified segments are then added to the training segments for re-training the regressors. Considering the high computational cost caused by large numbers of

results in the highest regression value determines the category of a query segment v_i, i.e., the category of v_i is computed by $y_i = \arg\max_k f_k(x_i)$.

The entropy is governed by the probability distribution of category labels that exist in $\mathscr{A}$, and it measures the consistency of the labels of selected segments. Note that the probability $p(j)$ is not calculated by counting the number of segments that contribute to each category, but by directly using the category label of each selected segment. The definition of the entropy term is given by:

$$Q(\mathscr{A}) = -\sum_{j \in \mathscr{A}} p(j) \log p(j), \tag{2.7}$$

with

$$p(j) = \frac{\arg\max_k f_k(x_j)}{\sum_{i \in \mathscr{A}} \arg\max_k f_k(x_i)}, \tag{2.8}$$

where the numerator denotes the object category of segment j, and the denominator sums all values on numerators to guarantee that the sum of $p(j)$ is one. To each candidate segment, its category is assigned by the scoring function that achieves the highest regression value. By maximizing $Q(\mathscr{A})$, we encourage the selected segment set $\mathscr{A}$ to possess homogeneous category labels, which reduce negative effects from erroneous regressors. We can obtain a maximum $Q(\mathscr{A})$ when all segments within $\mathscr{A}$ are assigned the same category. Note that different ways for including the entropy term into the objective function are used for the multi-category classification task (i.e., the Caltech-101 dataset) and the presence/absence classification[4] task (i.e., the PASCAL 2007 and the ETHZ-shape dataset dataset). For multi-category classification, regressors of all categories are used during the segment selection process, and a segment's category is allocated by the regressor that possesses the highest regression value. For the presence/absence classification task, only a single regressor of the query category is considered, and a segment's category is allocated as '1' if the regression value is above 0.5, and '2' otherwise. When the query category changes, different segments are selected with respect to a different regressor. Thus, our method can well handle the presence/absence classification problem, where images contain more than one object.

Fig. 2.5 show the segment selection example on the presence/absence classification task. Specifically, a small subset of segment hypothesis generated by CPMC is given in Fig. 2.5(b)∼(f). Aggregated confidence of selected segments are shown in Fig. 2.5(g)∼(i), where Fig. 2.5(g) is based on the facility location term only, Fig. 2.5(h) and Fig. 2.5(i) are based on both the facility location term and the entropy term. Without the entropy term as in

segments, we adopt the hard negative example mining strategy to refine the training as in [16].

[4]Following [6, 18, 28, 56, 115, 168], the presence/absence classification is to predict presence/absence of an example of that class in the test image.

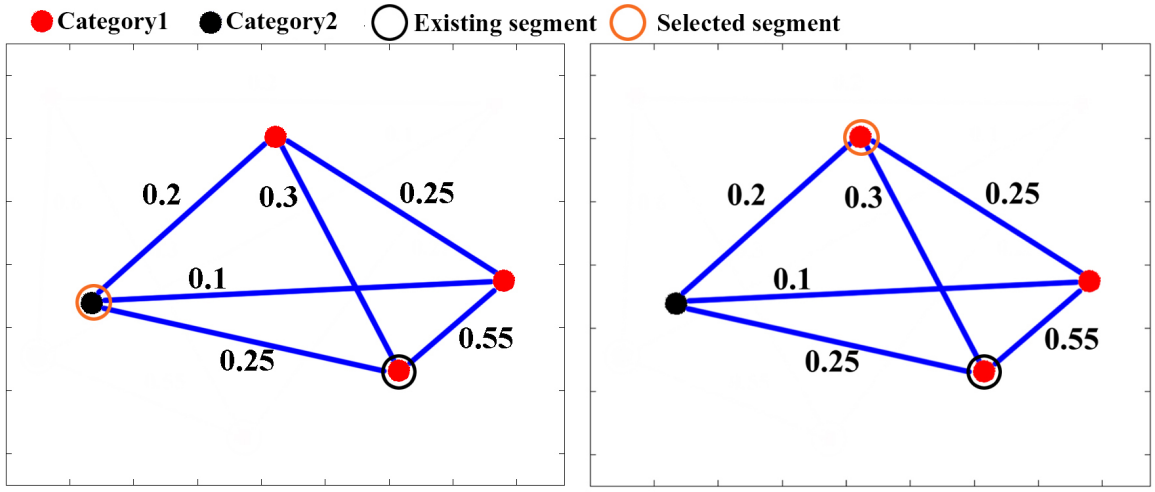


Fig. 2.4 Points with different colors denote vertices from different categories. The black circle denotes an existing segment which is selected in the first round selection. The orange circle denotes a selected segment at the second round selection. (a) shows the selection result based on the facility location term. According to Equation 2.3 and 2.5, if we assume $\phi_j = 1$, a maximum marginal gain $H(\mathscr{A}+a) - H(\mathscr{A})$ of 0.75 is reached when the black point is selected. (b) shows the selection result when integrating the entropy term together with the facility location term. According to Equation 2.2, 2.7 2.8 and 2.9, if we assume the tradeoff parameter $\lambda = 2$, the top red point is selected instead with a maximum marginal gain of 2.1.

Fig. 2.5(j), the facility location term only favors representative segments. When the entropy term is included, category-specific segments are selected. When detecting object category 'car' segments with high regression scores of the "car" regressor are prefered (shown in Fig. 2.5(k)). We can combine the facility-location term and the discriminative term into a unified objective function:

$$\begin{aligned} \max_{\mathscr{A}} C(\mathscr{A}) &= \max_{\mathscr{A}} H(\mathscr{A}) + \lambda Q(\mathscr{A}) \\ &= \max_{\mathscr{A}} \sum_{i \in \mathscr{V}} \max_{j \in \mathscr{A}} w_{ij} - \sum_{j \in \mathscr{A}} \phi_j \\ &\quad -\lambda \sum_{j \in \mathscr{A}} p(j) \log p(j), \end{aligned} \tag{2.9}$$

where $C(\mathscr{A})$ is the overall benefit value of selecting segment set $\mathscr{A}$. Fig. 2.4 illustrates how the facility location term and the entropy term contribute to a selection. A brief explanation of Fig. 2.4 is given in the following paragraph.

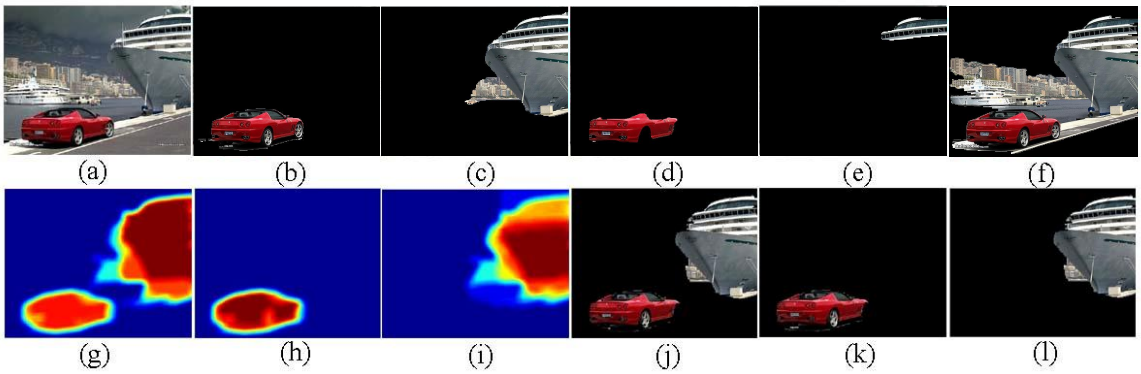


Fig. 2.5 An example of submodular segment selection for the presence/absence classification task. (a): Image I; (b)$\sim$(f): A small subset of segment hypothesis generated by CPMC; (g)$\sim$(i): Aggregated confidence of selected segments. (j)$\sim$(l): Illustrations of masked images L.

For subfigure 2.4(a), we consider the candidate facilities in the left to right order:

- $H(\mathscr{A}+a)-H(a) = (2+2+0.3+0.55-2\times\delta)-(2+0.25+0.3+0.55-\delta) = 1.75-\delta$, (the black point at the left side in Fig. 2.4)
- $H(\mathscr{A}+a)-H(a) = (2+2+0.25+0.55-2\times\delta)-(2+0.25+0.3+0.55-\delta) = 1.7-\delta$, (the red point at the top side in Fig. 2.4)
- $H(\mathscr{A}+a)-H(a) = (2+2+0.25+0.3-2\times\delta)-(2+0.25+0.3+0.55-\delta) = 1.45-\delta$, (the red point at the right side in Fig. 2.4)

If we set $\delta = 1$, the largest marginal gain equals to 0.75. For subfigure 2.4(b), we use a balancing parameter between the facility location term and the entropy term of $\lambda = 2$. When only one facility is selected, $Q(A) = 0$, then:

- $Q(\mathscr{A}+a) = -(\frac{1}{3}\log\frac{1}{3}+\frac{2}{3}\log\frac{2}{3}) \approx 0.6365$, (the black point at the left side in Fig. 2.4)
- $Q(\mathscr{A}+a) = -(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2}) \approx 0.6931$, (the red point at the top side in Fig. 2.4)

- $Q(\mathscr{A}+a) = -(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}) \approx 0.6931,$ (the red point at the right side in Fig. 2.4)

The largest marginal gain is when the red point at the top side is selected, where $H(A+a) - H(a) + \lambda \times E(A+a) = 1.7 - 1 + 2 \times 0.6931 \approx 2.1$, thus the case (b) is preferred.

Optimization

The optimization of the objective function defined in (2.9) follows the same greedy solution, however, the entropy term is neither monotonic nor submodular, which means the above solution is ad-hoc and it does not hold any theoretical performance guarantee based on the objective function given in (2.9). We claim that including the discriminative term still improves the overall recognition performance over the unsupervised segment selection method even though the solution may not be a global optimum. At the initialization stage when the algorithm selects the first segment into $\mathscr{A}$, the segment with the highest regression score is selected. For example, if we are detecting the presence/absence of a car within a query image, the segment that has the largest regression score, which is returned by the "car" regressor, is selected.

2.4.4 Segmentation Mask Construction

The final mask M is obtained by overlaying all segments in $\mathscr{A}$ while taking account of the confidence score $f_k(a_j)$ of each segment $a_j \in \mathscr{A}$. An adaptive threshold $\tau = 0.6 \times N_{\mathscr{A}}$ is applied to M to filter out pixels with low confidence scores.

2.4.5 Image Representation and Classification

Each mask M is in the form of a binary matrix, which has an identical size to the original image I. By performing element-wise multiplication on each mask M and the image I, a new matrix L can be obtained. In L, entries that correspond to non-zero elements in M are identical to pixel values of I, while the remaining entries equal to zero. Illustrations of L are given in Fig. 2.5(j)$\sim$(l). The SPM feature $x_I \in X$ is computed from L, and is used as the representation of image I, where X denotes the features for the whole training set. We can obtain the classifier parameters $\hat{W}$ of a linear classifier [52] through:

$$\hat{W} = \arg\max_{W} \|H - WX\|_2^2 + \varphi\|W\|_2^2, \tag{2.10}$$

Algorithm 2 Submodular Object Recognition based on Segment Selection Results

Input: Q, $\hat{W}$, I and τ.
Output: M and k^*.
Initialization:$M \leftarrow 0$.
for each pixel $Q(i,j)$ **do**
    if $Q(i,j) > \tau$ **then**
        $M(i,j) = 1$
    else
        $M(i,j) = 0$
    end if
end for
Integrate the final mask with the SPM framework by computing a global representation x_I based on $M \cdot I$ instead of I, where $\cdot$ means dot product.
Obtain the category $k^* = \arg\max_k k| \longrightarrow (\hat{W}x_I)_k$.

where H is the class label matrix of X, and W denotes classifier parameters. This yields the solution $\hat{W} = HX^T(XX^T + \varphi\mathscr{L})^{-1}$, with $\mathscr{L}$ being an identity matrix. For a test image I, we first compute its representation x_I and then estimate its class label vector $l = \hat{W}x_I$, where $l \in R^m$. Its label is the index i corresponding to the largest element in l. The pseudocode of how object recognition can be achieved based on segment selection result is given in Algorithm 2.

2.5 Experiments

We evaluate our submodular object recognition approach on three popular benchmarks, including Caltech-101 [37], PASCAL VOC 2007 [Everingham et al.] and ETHZ-shape [41]. For all three datasets, we compute the dense SIFT features on each image. Regressors are trained based the ground-truth segmentations provided with the training data. For all the experiments, we evaluate our approach by either using the facility location term ("FL") only or using both the facility location term and the entropy term ("FL"+"EN").

2.5.1 PASCAL VOC 2007

We extensively evaluate the effectiveness of our approach on the PASCAL VOC 2007 dataset, as the ground-truth of the testing data is released. The PASCAL VOC 2007 dataset contains $9,963$ images from 20 visual object categories, and the dataset is evenly split to

Table 2.1 Average precisions (APS) of each object category achieved by the baseline method and our proposed methods on the PASCAL VOC 2007 dataset.

Methods	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	Avg
Yang [168]	74.8	65.2	50.7	70.9	28.7	68.8	78.5	61.7	54.3	48.6	51.8	44.1	76.6	66.9	83.5	30.8	44.6	53.4	78.2	53.5	59.3
Florent [115]	75.7	64.8	52.8	70.6	30.0	64.1	77.5	55.5	55.6	41.8	56.3	41.7	76.3	64.4	82.7	28.3	39.7	56.6	79.7	51.5	58.3
Harzallah [56]	77.2	69.3	56.2	66.6	45.5	68.1	83.4	53.6	58.3	51.1	62.2	45.2	78.4	69.7	86.1	52.4	54.4	54.3	75.8	62.1	63.5
Qiang [18]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	66.5	52.5	57.5	62.8	51.1	81.4	71.5	86.5	36.4	55.3	60.6	80.6	57.8	64.7
Dong [28]	82.2	**83.0**	58.4	76.1	56.4	77.5	88.8	69.1	62.2	61.8	64.2	51.3	85.4	80.2	**91.1**	48.1	61.7	67.7	86.3	70.9	71.1
FL	81.2	82.2	56.7	73.5	56.2	76.5	88.5	67.8	58.0	60.1	61.7	48.1	85.1	77.8	89.3	45.5	60.6	64.4	84.3	69.2	69.3
FL+EN	**83.7**	82.5	**63.3**	**77.3**	**58.0**	**80.2**	**89.4**	**68.8**	**63.1**	**63.7**	**67.4**	**53.5**	**86.4**	**82.7**	90.5	**48.4**	**62.0**	**67.9**	**87.2**	**71.5**	**72.4**

"trainval" and "test" parts. Following typical settings in [18, 28, 56, 115, 168], we conduct experiments on the "trainval" and "test" splits. In our algorithm, we train the regressors according to the overlap observations between each figure-ground hypothesis and the ground-truth segmentation of an object category. Since the ground-truth segmentations are only available for those images provided in the segmentations challenge, we train the regressors only based on images with provided ground-truth segmentation in the "trainval" split. We show the results achieved by both "FL" and "FL+EN" in Table 2.1. We calculate the average precisions (APs) for each object category using both approaches, and compare with state-of-the-art approaches [18, 28, 56, 115, 168]. As can be observed, the "FL+EN" approach outperforms all other approaches.

2.5.2 Caltech-101 Dataset

The Caltech-101 dataset [37] contains 9,144 images from 102 classes (101 object classes and a 'background' class). The ground-truth segmentations are provided in this dataset. We train a codebook with 2048 bases, and choose 4×4, 2×2 and 1×1 sub-regions for SPM. Following the common experimental protocol, randomly selected 5, 10, 15, 20, 25, 30 samples per category are used for training, and remaining samples are for testing. We repeat the experiments 10 times and the final results are reported as the average of each run. We compare our results with state-of-the-art approaches [16, 67, 131, 168] in Table 2.2. We also show the results of "BS" and "GT", which denote results produced by using only the best segments[5] and ground-truth segments, respectively. The high performance of "BS" and "GT" proves our motivation that recognition performance can be improved by segmentation.

We randomly select 30 images as training data, and evaluate our approach when different values of the entropy term weight λ and the penalty cost δ are selected. As shown in

[5]For an image I, the best segment is a segment that has the largest overlap with the ground-truth segment G_I.

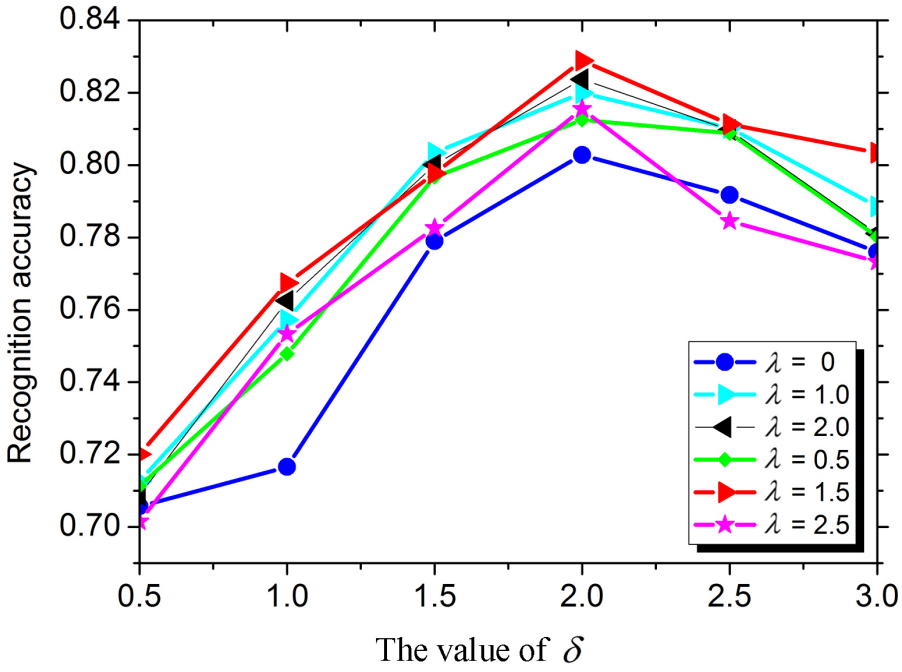


Fig. 2.6 Effects of parameter selection of λ and δ on the recognition performance on the Caltech-101 dataset when using 30 training examples per category. The horizontal axis denotes different values of δ, while lines with different colors denote different λ values.

Fig. 2.6, the best performance is achieved when $\lambda = 1.5$ and $\delta = 2$. If λ is set to 0, the performance degrades since segments' label consistency is not considered. On the other hand, if λ is too large, pursuing segments'label consistency while considering less their visual saliency is harmful to the performance. The performance is more sensitive to the penalty cost δ. If δ is large, the cost of selecting a new segment can easily exceed the gain that the selected segment can contribute to the objective function. Consequently, only one or a few facilities can be selected. In general, there does not exist a single correct segmentation for an image, so that the performance is weakened when recognition is performed on too few segments within an image (as shown in Fig. 2.7(c) and Fig. 2.7(d)). A small δ can lead to a large collection of segments being selected. Thus, the intersection of selected segments is concentrated on a small image region (as shown in Fig. 2.7(f)), and much object information is discarded. As a result, recognition performance significantly degrades. Fig. 2.8 demonstrates results of aggregated confidence maps of selected segments and resulting foreground objects, and Fig. 2.9 shows example images from classes with high classification accuracy of the Caltech-101 dataset.

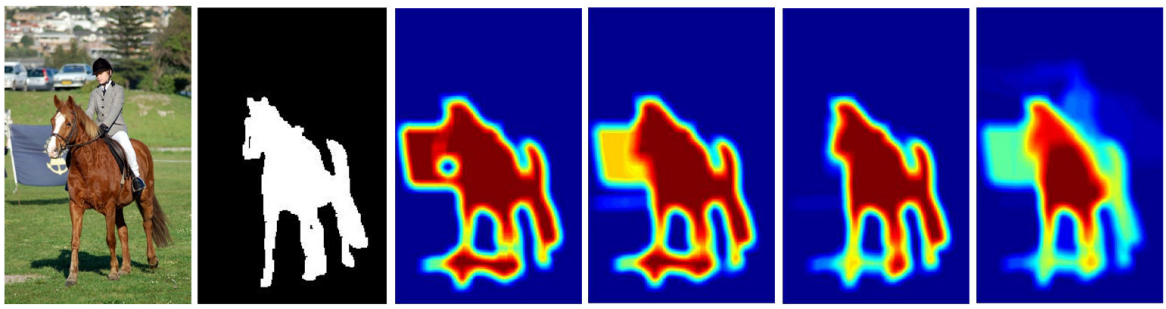


Fig. 2.7 Effects of parameter selection of δ on the aggregated confidence of selected segments $\mathscr{M}$. (a): Input image; (b): Ground truth segment; (c)$\sim$(f): The aggregated confidence of selected segments when the penalty cost $\delta = 3, 2.5, 2, 0.5$, respectively. The color denotes different confidence values (red: high, blue: low). In case of too few segments are selected as in (c), the aggregated confidence does not have accurate coverage of the object. The coverage of the aggregated confidence is improved in (d) when more segments are selected. (e) has the most accurate coverage. $\mathscr{A}$ can "over-select" segments if we reduce the penalty term. In (f), the aggregated confidence focuses on a small central region of the object as too many segments are selected.

2.5.3 ETHZ Shape Classes

The ETHZ Shape Classes dataset [118] contains 255 images from 5 shape categories, including "Applelogo", "Bottles", "Giraffes", "Mugs", "Swans", and object ground-truth outlines are provided for all images. Following the PASCAL classification criterion, for each of the 5 categories, we predict presence/absence of an example of that class in the test image. The dataset is evenly split into training and testing sets and performance is averaged over 5 random splits. Performance comparisons between our approaches ("FL" and "FL" +"EN") and approaches in [67, 168] are given in Table 2.3. It can be observed that the proposed "FL"+"EN" significantly outperforms other methods. The ROC curves of our approaches and approaches in [67, 168] for the all five categories are shown in Fig. 2.10.

2.6 Conclusions

We have proposed a segmentation-based object recognition approach based on submodularity. Segment selection methods based on both saliency and discriminativity are proposed to suppress background information in images through selecting representative and discriminative segments from image segmentation results. Salient segments are selected by maximizing a submodular function, which can be viewed as a facility location problem, and

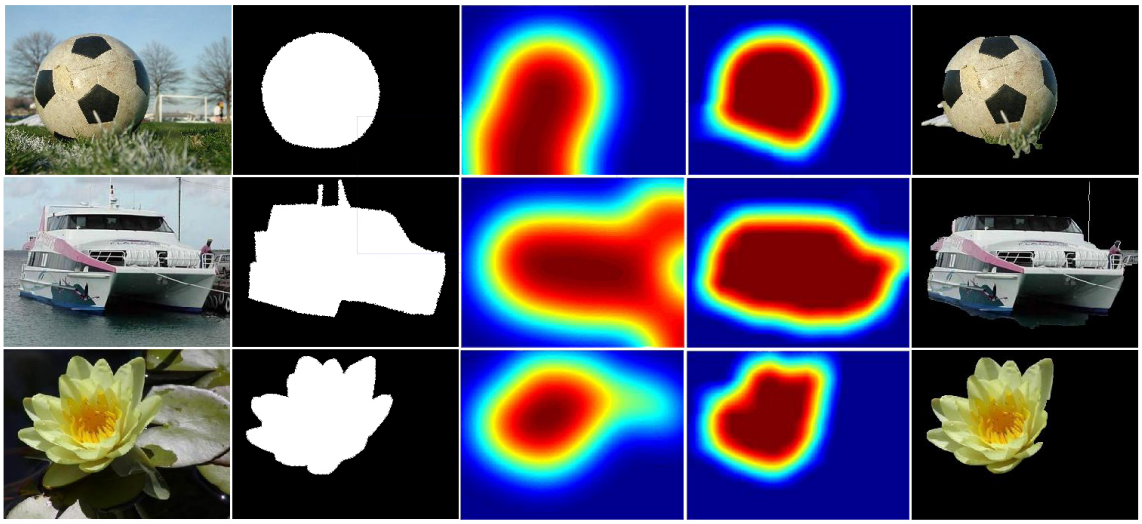


Fig. 2.8 Examples of aggregated confidence maps of selected segments on images from Caltech-101 dataset. (a): Input images; (b): Ground truth object segmentations; (c): Aggregated confidence of selected segments using "FL" method only; (d): Aggregated confidence of selected segments using "FL"+"EN" method; (e): Foreground objects based on masks generated by the results of (d) through adaptive thresholds.

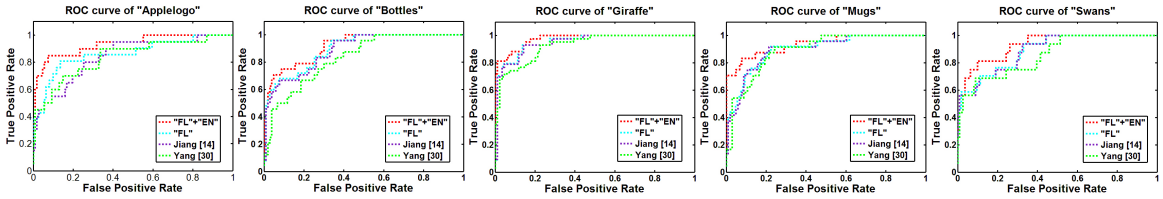


Fig. 2.9 ROC curves of our approaches ("FL" and "FL" +" EN") and state-of-the-art approaches on the all five categories of the ETHZ Shape Classes dataset.

discriminative segments are selected by applying pre-learned category-specific regressors to segments in the query image. The discriminative term is also governed by entropy, which favors the consistency of assigned labels of selected segments.Experimental results on three public benchmarks indicate that our method outperforms state-of-the-art recognition techniques.

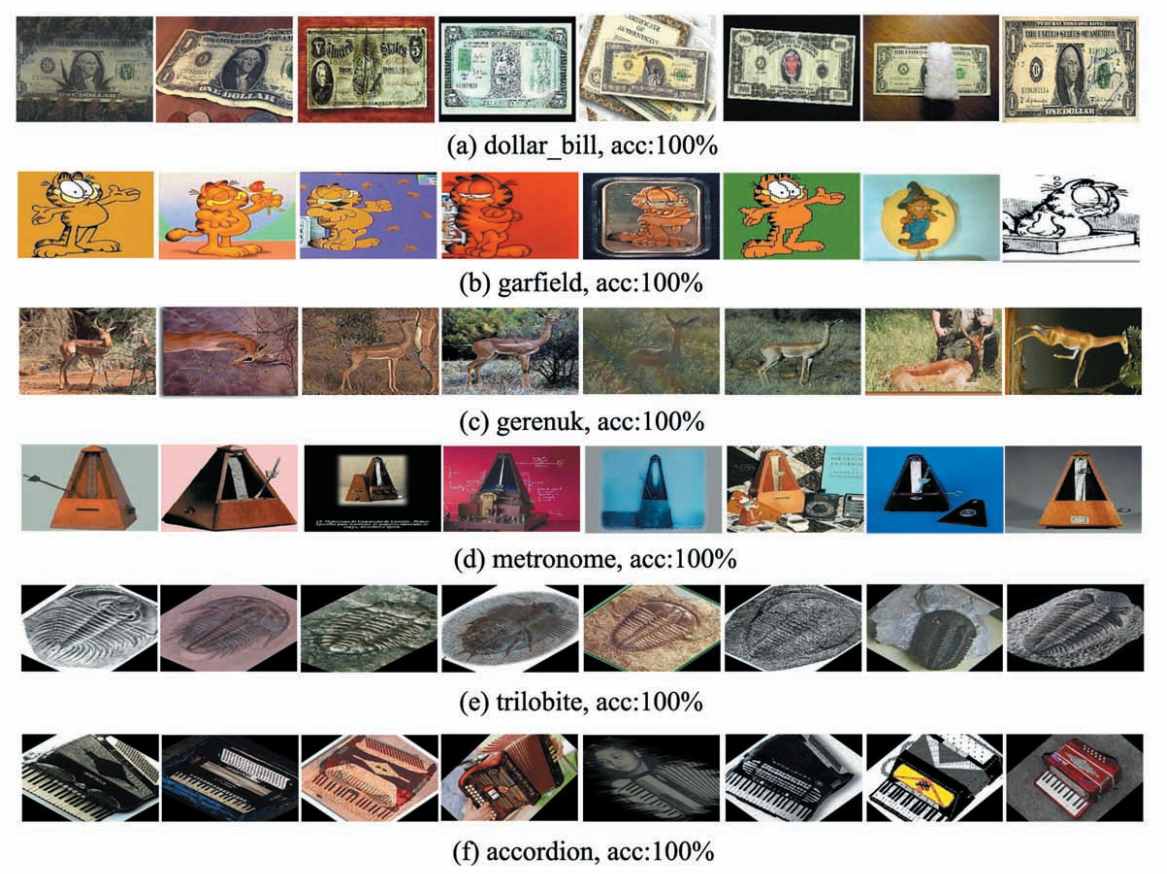


Fig. 2.10 Example images from classes with high classification accuracy of the Caltech-101 dataset.

Table 2.2 Recognition accuracies using spatial pyramid features on the Caltech-101 dataset. "BS" and "GT" denote results produced by using only the best ranked segments and ground-truth segments, respectively.

Method	5	10	15	20	25	30
Yang [168]	49.84%	57.26%	62.75%	68.78%	71.12%	73.72%
Jiang [67]	54.00%	63.10%	67.70%	70.50%	72.30%	73.60%
Shaban [131]	54.01%	63.86%	68.70%	71.58%	73.73%	75.07%
Carreira [16]	60.90%	–	74.70%	–	–	81.90%
GT	68.71%	77.67%	81.33%	84.49%	86.73%	88.34%
BS	63.95%	72.03%	77.66%	79.69%	82.24%	83.27%
FL	59.81%	68.45%	73.90%	76.98%	78.96%	80.28%
FL+EN	63.29%	71.47%	76.43%	78.26%	81.03%	83.18%

Table 2.3 Average precisions (APS) of each object category on the ETHZ shape classes dataset.

Methods	Apple	Bottles	Gira	Mugs	Swans	Avg
Yang [168]	83.79	83.13	92.77	89.16	85.75	86.92
Jiang [67]	84.11	88.71	94.74	89.64	88.91	89.22
FL	86.40	89.50	95.04	89.72	89.16	89.96
FL+EN	**93.18**	**91.71**	**97.43**	**93.61**	**92.96**	**93.78**

Chapter 3

Transfer Feature Learning

3.1 Cross-Domain Dictionary Learning[1]

3.1.1 Motivation and Introduction

In the past few years, along with the explosion of online image and video data (Flickr [2], YouTube [3]), the computer vision community has witnessed a significant amount of applications in content-based image/video search and retrieval, human-computer interaction, sport events analysis, etc. These applications are built upon the development of several aspects of classical computer vision tasks, such as human action recognition, object localization and image classification, which, however, remain challenging in real-world scenarios due to cluttered background, view point changes, occlusion, and geometric and photometric variations of the target [145], [172], [157], [63], [70], [156], [31], [102]. These issues result in either imposing irrelevant information to the target introduced by, e.g., cluttered background, or producing very different representations for the same target caused by, e.g., geometric and photometric changes. Many previous methods that manage to deal with these issues are proposed and some state-of-the-art approaches include semantic attributes [145], estimated pose features [172], and mined hierarchical features [50]. The conventional framework ap-

[1] The content of this section is published at:
F. Zhu and L. Shao, Weakly-Supervised Cross-Domain Dictionary Learning for Visual Recognition, International Journal of Computer Vision, vol. 109, no. 1-2, pp.42-59, Aug. 2014
F. Zhu and L. Shao, Enhancing Action Recognition by Cross-Domain Dictionary Learning, British Machine Vision Conference, Bristol, UK, Sep. 2013.

[2] http://www.flickr.com/

[3] http://www.yoututbe.com/

plies a robust classifier using human annotated training data, and makes the assumption that the testing data stay in the same feature space or share the same distribution with the training data. However, in real-world applications, due to the high price of human manual annotation and environmental restrictions, sufficient training data that stay in the same feature space or share the same distribution with the testing data cannot always be guaranteed, in which case insufficient training data can limit the potential discriminability of the trained model. Typical examples are [15], [48], [111], where only one action template is provided for each action class for training, and [88], where training samples are captured from a different viewpoint. In these situations, obtaining more labeled data is either impossible or expensive, while seeking for an alternative way of using data from other domains as compensation can be seen as a possible and economic solution.

We introduce a new visual categorization framework that utilizes weakly labeled data obtained from online resources as the source data to span the intra-class diversity of the original learning system. In addition to the manually labeled training data in the target domain, the source domain data are utilized as extensions of category prototypes in the target domain. Fig 3.1 illustrates a simple example of how auxiliary data can help with the classification tasks. We let purple triangles, the orange circles and the red squares denote the training samples from Classes 1, 2 and 3 respectively, and the corresponding hollow shapes denote the auxiliary training samples from Classes 1, 2 and 3. Original decision boundaries are drawn according to the target domain data and are represented by the solid lines, and the new decision boundaries are drawn according to the updated data and are represented by the dashed lines. The testing sample, which is denoted as a red square with black borders, is misclassified as Class 1 according to the original decision boundaries because the original target domain training data are insufficient and cannot provide a wide enough coverage. Thus, the aim of including auxiliary domain data is to span the intra-class diversity of the original training data, and thus help with the classification task.

Based on the recent success of dictionary learning methods in solving computer vision problems, we present a weakly-supervised cross-domain dictionary learning method to learn a reconstructive[4], discriminative and domain-adaptive dictionary pair and an optimal linear classifier simultaneously. In order to demonstrate the effectiveness of our method, we gather supportive evidence by evaluating our method on action recognition, image classification and event recognition tasks. The UCF YouTube dataset [86], the Caltech101 dataset [38], the Caltech 256 dataset [54] and the Kodak consumer video dataset [93] are used as the

[4] A reconstructive dictionary means that the error between the original signals and the reconstructed signals, which are obtained based on corresponding dictionary atoms and sparse codes, is small.

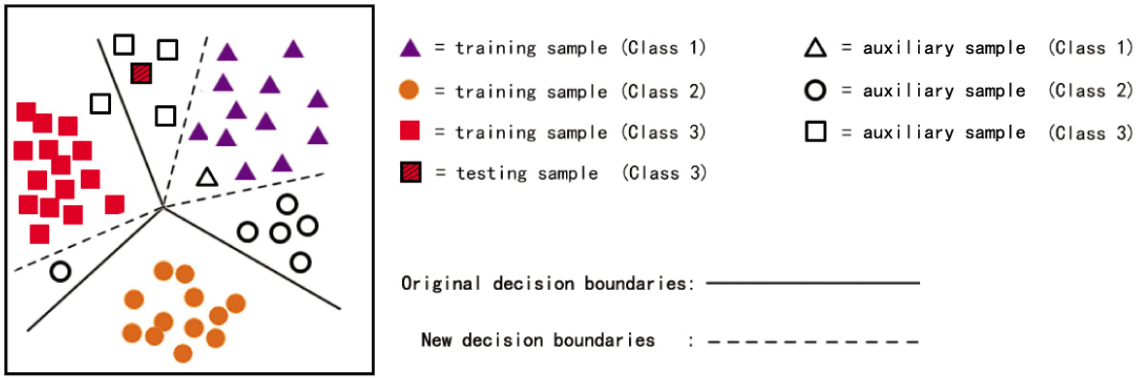


Fig. 3.1 Illustration of how the auxiliary data help with the classification task. Original decision boundaries are represented by the solid lines and the new decision boundaries are represented by the dashed lines. By adding the auxiliary data, the new decision boundaries are drawn according to the updated data, which provide a better coverage.

target domain data in our experiments, while selected actions in the HMDB51 dataset [55] and some indexed Web images or YouTube videos are used as the source domain data in our experiments.

The dictionary learning framework of the proposed method is illustrated in Fig. 3.1 and it offers the following two main contributions. Firstly, it attempts to make use of as much as possible existing knowledge by a novel weakly-supervised visual categorization framework. An efficient manifold ranking method is applied to the source domain for the selection of a pre-defined number of most relevant instances per category according to the target domain training data, following which correspondences connecting the source domain and the target domain are established based on the selected source domain data and the target domain training data. Secondly, we propose a new cross-domain dictionary learning method to cope with the feature distribution mismatch problem across the source domain and the target domain. Specifically, we perform dictionary learning upon the correspondences built from both domains so that the projections of data from different domains can obey the same distribution when limited by the learning function. In addition to the dictionary, classifier parameters are learned jointly during the discriminative dictionary function learning process. Thus, knowledge transfer of the proposed framework is accomplished through both the feature level and the classifier level. As the samples from the source domains are weakly labeled rather than being manually (correctly) labeled, we call our algorithm “Weakly-Supervised Cross-Domain Dictionary Learning”(WSCDDL).

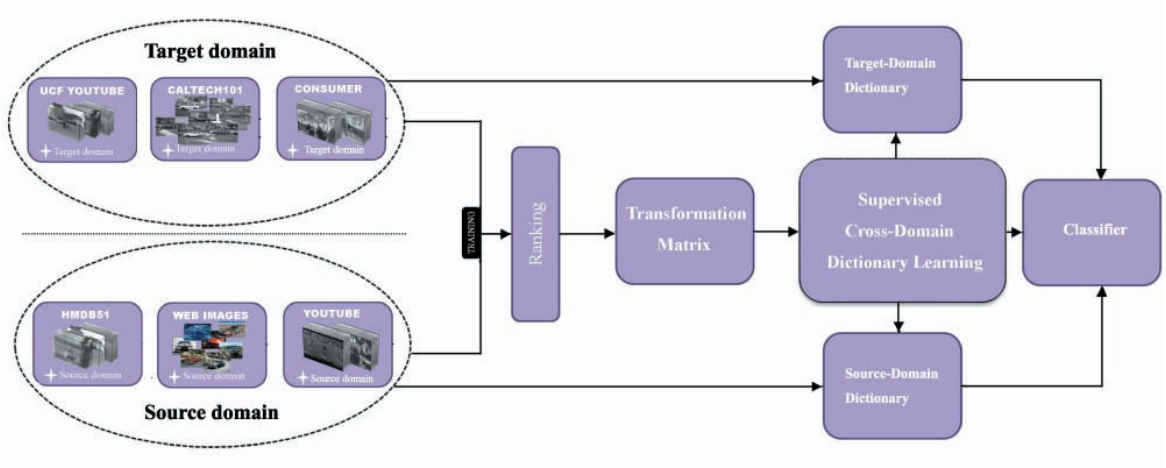


Fig. 3.2 Illustration of the cross-domain dictionary learning framework. By applying manifold ranking to the source domain data, pruned and virtually labeled source domain data are obtained. The cross-domain dictionary learning method is applied to both the target domain data and the pruned source domain data, then the learned target domain dictionary, source domain dictionary and parameters of the linear classifier are obtained for the testing stage.

3.1.2 Related Work

A considerable number of methods have been proposed to address visual categorization problems [98], [64], [126], [124], [125], [90]. Reasonable results are achieved using traditional machine learning approaches without considering the data distribution mismatch among the training data and the testing data when training data are abundant. Transfer learning (a.k.a., cross-domain learning, domain transfer, domain adaptation) approaches begin to attract increasing interests in the computer vision community in recent years due to the data explosion on the Internet and the growing demands for visual computational tasks. In [13], action detection is conducted across datasets from different visual domains, where the KTH dataset [129], which has a clean background and limited viewpoint and scale changes, is set as the source domain, and the Microsoft Research Action Dataset [5] and the TRECVID surveillance data [25], which are captured from realistic scenarios, are used as the target domain. [167] and [29] addressed the problem of video concept detection using domain transfer approaches. The former one utilized the Adaptive Support Vector Machine (A-SVM) to adapt one or more existing classifiers of any type to a new dataset, and the latter proposed a Domain Transfer Multiple Kernel Learning (DTMKL) method to simultaneously learn a kernel function and a robust SVM classifier by minimizing both the structural risk function of SVM and the distribution mismatch of labeled and unlabeled data

[5] http://research.microsoft.com/∼zliu/ActionRecoRsrc

in different domains. Authors of [88] and [82] constructed cross-domain representations to cope with the cross-view action recognition problem, where the divergences across domains are caused by view-point changes. Liu et al. [88] built a bipartite graph via unsupervised co-clustering to measure the visual-word to visual-word relationship across the target view and the source view so that a high-level semantic feature that bridges the semantic gap between the two vocabularies can be filled. Similarly, Li et al. [82] captured the conceptual idea of "virtual views"to represent an action descriptor continuously from an observer's view-point to another. Duan et al. [75] considered to leverage large amounts of loosely labeled web videos for visual event recognition using the Adaptive Multiple Kernel Learning (A-MKL) to fuse the information from multiple pyramid levels and features and cope with the considerable variation in feature distributions between videos across two domains.

Recently, dictionary learning for sparse representation has attracted much attention. It has been successfully applied to a variety of computer vision tasks, e.g., face recognition [161] and image denoising [183]. Using an over-complete dictionary, sparse modeling of signals can approximate the input signal by a sparse linear combination of items from the dictionary. Many algorithms [79], [68], [161] have been proposed to learn such a dictionary according to different criteria. The K-Singular Value Decomposition (K-SVD) algorithm [1] is a classical dictionary learning algorithm that generalizes the K-means clustering process for adapting dictionaries to efficiently learn an over-complete dictionary from a set of training signals. The K-SVD method focuses on the reconstructive ability, however, since the learning process is unsupervised, the discriminative capability is not taken into consideration. Consequently, methods that incorporate the discriminative criteria into dictionary learning were proposed in [177], [169], [101], [97], [101], [10]. In addition to the discriminative capability of the learned dictionary, other criteria designed on top of the prototype dictionary learning objective function include multiple dictionary learning [178], category-specific dictionary learning [170], etc. Different from most dictionary learning methods, which learned the dictionary and the classifier separately, the authors of [177] and [66] unified these two learning procedures into a single supervised optimization problem and learned a discriminative dictionary and the corresponding classifier simultaneously. Taking a step further, Qiu et al. [117] and Zheng et al. [62] designed dictionaries for the situations that the present training instances are different from the testing instances. The former presented a general joint optimization function that transforms a dictionary learned from one domain to the other, and applied such a framework to applications such as pose alignment, pose and illumination estimation and face recognition. The latter achieved promising results on the cross-view action recognition problem with pairwise dictionaries constructed using

correspondences between the target view and the source view. To make use of some data that may not be relevant to the target domain data, Raina et al. [120] proposed a method that applies sparse coding to unlabeled data to break the huge amount of data in the source domain into basic patterns (e.g., edges in the task of image classification) so that knowledge can be transferred through the bottom level to a high level representation.

Our approach differs from the above approaches in such aspects that it more comprehensively learns pairwise dictionaries and a classifier while considering the capacity of the dictionaries in terms of reconstructability, discriminability and domain adaptability. Additionally, corresponding observations across the domains are not required in our framework. While most previous knowledge transfer algorithms focus on the situations where the target domain is incomplete, but have not attempted to utilize other domain data as an aid for enhancing present categorization systems, in our approach, the learned classifier in the target domain becomes more discriminative against intra-class variations as a result of the learning process that integrates with source domain data. Our work makes the following contributions:

⋆ We present a novel cross-domain action recognition framework that attempts to enhance the performance of the original recognition system by spanning the intra-class diversities of the target domain training actions using actions from the source domain.

⋆ The proposed discriminative cross-domain dictionary learning technique copes with the feature distribution mismatch problem across different domains by learning a domain-adaptive dictionary pair that transfers data under different distributions into the same feature space.

⋆ Our approach does not require correspondence annotations across different domains, so that it can be adapted to solve many real-world transfer learning problems.

3.1.3 Knowledge Transfer via Discriminative Dictionary Learning

3.1.4 Problem Formulation

Given a collection of partially labeled images/videos, our goal is to learn a classifier that automatically specifies different classes for the unlabeled instances using the labeled ones. The source domain datasets are constructed according to the given image/video categories. Specifically, when treating the UCF YouTube dataset as the target domain data, the source domain data are composed of corresponding or similar action categories (ride bike, dive, golf, jump, kick ball, ride horse, shoot ball) in the HMDB51 dataset. Compared to the UCF

YouTube dataset, the HMDB51 dataset contains more severe camera motions, viewpoint changes, video quality variations and occlusions, and is thus more realistic (a detailed comparison between the UCF YouTube dataset and the HMDB51 dataset is given in [55]). On the other hand, when setting the Caltech101/Caltech256 dataset as the target domain data, we choose the N_s image categories (chosen according to the ascending alphabetic order) and use the first N_g results returned from Google Image Search for each chosen category as the source domain data, where the indexing procedure is performed by simply searching the category names. Following the terminology from prior literature, we denote $\mathscr{D}_t$ as the target domain, and $\mathscr{D}_t = \mathscr{D}_t^l \cup \mathscr{D}_t^u$, where $\mathscr{D}_t^l$ and $\mathscr{D}_t^u$ denote labeled target domain data and unlabeled target domain data respectively. $\mathscr{D}_s$ denotes the source domain data.

Manifold Ranking

The manifold ranking [181], [182] algorithm is applied as a pre-processing stage of knowledge transfer by filtering out irrelevant data in the source domain data X_s and assigning virtual labels to unlabeled source domain data. An illustration of the manifold ranking procedure is given in Fig. 3.3. For each category in X_s, 5 samples are manually labeled so that the ranking is conducted in a semi-supervised manner. We define the labeled data points in X_s as positive samples, and the remaining points as negative. Assume the initial number of auxiliary samples (i.e., source domain samples) is M, a vector $y^* = [y_1^*, y_2^*, \cdots, y_M^*]$ is defined, where $y_i^* = 1$ if the i-th data point X_s^i is a positive sample, otherwise $y_i^* = 0$. We also define a weight vector $r = [r_1, r_2, \cdots, r_M]^T$ which indicates the overall importance of each data point. Similar as in[182], the ranking procedure can be described as follows:

- Construct a connected graph based on X_s. Sort the pairwise distance between any pair of data points within X_s, and connect the data pairs in a ascending order until a connected graph is obtained. If two points are connected, assign the weight $w_{ij}^* = exp[-d^2(x_i, x_j)/2\sigma^2]$ to the weight matrix W^*, otherwise $w_{ij}^* = 0$.

- The affinity matrix W^* is then symmetrically normalized by $S^* = D^{*-1/2} W^* D^{*-1/2}$, where D^* is a diagonal matrix and $D^*(i,i)$ equals to the sum of the elements in i-th row of W^*.

- The weight vector is iteratively updated by $r^{\text{new}} = \alpha^* S^* r + (1 - \alpha^*) y^*$ until convergence, where α^* is a parameter in $[0, 1)$.

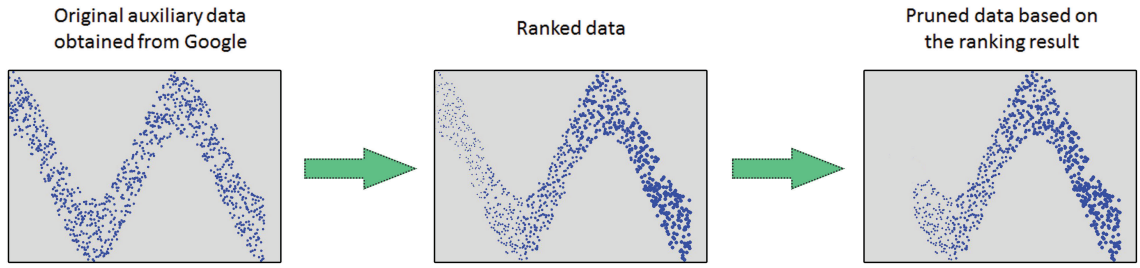


Fig. 3.3 Illustration of how the online data are preprocessed with the manifold ranking method. Manifold ranking assigns weights to all the auxiliary data, where the marker size of each data point is proportional to its overall importance (shown in the middle window). Finally, data with low weights are pruned (shown in the right window).

- When the ranking scores r^* are obtained, we keep 80% data points with the highest ranking scores for each category in the image classification and event recognition tasks.

This particular weight updating method is chosen because the aim of this ranking algorithm is to spread the ranking scores of all points to their nearby neighbors. When the weight W^* is normalized to S^*, the ranking scores can be updated with respect to the network connections. The vector y^* indicates confidence of each instance within the network. By assigning '1's in y^* to query instances, strong confidences are given to the queries. Rigorous convergence proof of this weight updating algorithm is given in [182]. By conducting the manifold ranking procedure, the weakly-labeled source domain data are assigned virtual labels, and noisy data are pruned for image classification and event recognition tasks.

3.1.5 Dictionary Learning

We denote Y_t as the target domain n-dimensional low level image/video data with N training instances and Y_s as the corresponding source domain n-dimensional data with M training instances, i.e., $Y_t = [y_t^1, ..., y_t^N] \in \mathbb{R}^{n \times N}$ and $Y_s = [y_s^1, ..., y_s^M] \in \mathbb{R}^{n \times M}$. Learning a reconstructive dictionary for obtaining the sparse representation of Y_t and Y_s can be accomplished by solving the following optimization problems:

$$< D_t', X_t >= arg \min_{D_t, X_t} \|Y_t - D_t X_t\|_2^2 \qquad s.t. \forall i, \|x_t^i\|_0 \leq T, \tag{3.1}$$

$$< D'_s, X_s >= arg \min_{D_s, X_s} \|Y_s - D_s X_s\|_2^2 \qquad s.t. \forall i, \|x_s^i\|_0 \leq T, \tag{3.2}$$

where $D_t = [d_t^1, ...d_t^{K_t}] \in \mathbb{R}^{n \times K_t}$ is the learned dictionary in the target domain, $X_t = [x_t^1, ...x_t^N] \in \mathbb{R}^{K_t \times N}$ is the target domain sparse signal, $D_s = [d_s^1, ...d_s^{K_s}] \in \mathbb{R}^{n \times K_s}$ is the learned dictionary in the source domain and $X_s = [x_s^1, ...x_s^M] \in \mathbb{R}^{K_s \times M}$ is the source domain sparse signal, respectively. The numbers of dictionary items K_t and K_s are set to be greater than both N and M to ensure that the dictionaries are over-complete.

By minimizing the terms $\|Y_t - D_t X_t\|_2^2$ and $\|Y_s - D_s X_s\|_2^2$ in both error functions while satisfying the sparsity constraint, two optimized reconstructive dictionaries D'_t and D'_s along with the sparse representations X'_t and X'_s are obtained. As the two energy minimization procedures in Equation (3.1) and Equation (3.2) are performed separately, the encoded data still obey to different distributions. In order to minimize the cross-domain divergence, we propose to build correspondences across both domains and minimize the distances between each corresponded pair. As stated in Section 3.1.4, the source domain samples are assigned virtual labels, however, the correspondence information is not available in the stated scenario. We build approximate correspondences by connecting each target domain training instance to its most similar source domain instance, which shares the same label with the target domain instance, based on Euclidean distance. Note that even Euclidean distance is not a precise estimation for cross-domain data, each correspondence is at least established between instances of the same category. The correspondences of category c are stored in the transformation matrix $\mathbb{A}_c$, in which $\mathbb{A}_c(i, j) = 1$ if the source domain training instance i is corresponded to the target domain instance j, otherwise $\mathbb{A}_c(i, j) = 0$. Since each target domain training instance is connected to a single source domain instance, each column of $\mathbb{A}_c$ allows only one non-zero entry. Specifically, for each column j, $\mathbb{A}_c(i, j)$ can be computed as:

$$\mathbb{A}_c(i^*, j) = \begin{cases} 1, & if \quad i^* = \arg\max_{i=1:M} (G_c(i, j)) \\ \\ 0, & otherwise, \end{cases} \tag{3.3}$$

where G_c is a $M \times N$ dimensional matrix, in which each $G_c(i, j)$ is the Euclidean distance between the ith source domain instance and the jth target domain instance. Consequently, the global transformation matrix $\mathbb{A}$ for all categories can be obtained by filling each $\mathbb{A}_c$ into

a single matrix:

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_1 & & & \\ & \mathbb{A}_2 & & \\ & & \ddots & \\ & & & \mathbb{A}_C \end{pmatrix}, \tag{3.4}$$

So far, the cross-domain dictionary learning function can be written as:

$$\begin{aligned} < D_t', D_s', X_t', X_s' > = arg \min_{D_t, D_s, X_s, X_t} & \|Y_t - D_t X_t\|_2^2 + \|Y_s \mathbb{A}^T - D_s X_s \mathbb{A}^T\|_2^2 + \|X_t - X_s \mathbb{A}^T\|_2^2 \\ s.t. \forall i, \ & \|x_t^i\|_0 \leq T, \end{aligned} \tag{3.5}$$

where $\|X_t - X_s \mathbb{A}^T\|_2^2$ estimates the cross-domain divergence. We add an additional constraint to the learning function to allow corresponded cross-domain instances possess identical representations in the projected feature space, i.e., $X_t = X_s \mathbb{A}^T$. Thus, the objective function can be formulated as:

$$\begin{aligned} < D_t', D_s', X_t', X_s' > = arg \min_{D_t, D_s, X_t} & \|Y_t - D_t X_t\|_2^2 + \|Y_s \mathbb{A}^T - D_s X_t\|_2^2 \\ s.t. \forall i, \ & \|x_t^i\|_0 \leq T, \end{aligned} \tag{3.6}$$

Since the learned sparse representation can be directly fed into the classifier, separating the dictionary learning stage from the classification procedure might lead suboptimal D_t and D_s. Thus we attempt to jointly learn the dictionaries and the classifier by including the discriminative term in dictionary learning. Let the model parameters W of the classifier $\mathscr{F}(x)$ satisfy Equation (3.7):

$$W' = arg \min_W \sum_i \mathscr{L}\{h_i \mathscr{F}(x_t^i, W)\} + \lambda_1 \|W\|_2^2, \tag{3.7}$$

where $\mathscr{L}$ is the classification loss function (we use the logistic loss function $\mathscr{L}(z) = \log(1 + e^{-z})$ as in [97]), h_i indicates the target domain labels of x_t^i, $\mathscr{F}(x_t^i, W)$ is a positive value for any signal in the positive class and a negative value otherwise, W denotes the classifier parameters and λ_1 is a regularization parameter. Intuitively, the original objective function

for dictionary learning can then be updated as:

$$\begin{aligned} < D_t', D_s', X_t' > = arg \min_{D_t, D_s, X_t} & \|Y_t - D_t X_t\|_2^2 + \|Y_s \mathbb{A}^T - D_s X_t\|_2^2 \\ & + \sum_i \mathscr{L}\{h_i, \mathscr{F}(x_t^i)\} + \lambda_1 \|W\|_2^2 \\ & s.t. \forall i,\ \|x_t^i\|_0 \leq T. \end{aligned} \tag{3.8}$$

As in previous work [97], [169], [66], [177], the classification error of a linear predictive classifier is included in the objective function:

$$\begin{aligned} < D_t', D_s', X_t', \mathbb{A}', W' > = arg \min_{D_t, D_s, X_t, A, W} & \|Y_t - D_t X_t\|_2^2 + \|Y_s \mathbb{A}^T - D_s X_t\|_2^2 \\ & + \alpha \|Q - \vartheta X_t\|^2 + \beta \|H - W X_t\|_2^2 \\ & s.t. \forall i,\ \|x_t^i\|_0 \leq T, \end{aligned} \tag{3.9}$$

where ϑ is a linear transformation matrix that maps the original sparse codes to be in correspondence with the target discriminative sparse codes $Q = [q_1, q_2, \cdots, q_N] \in \mathbb{R}^{K \times N}$ of the input signal Y_t. Specifically, $q_i = [q_i^1, q_i^2, \cdots, q_i^K] = [0, \cdots, 1, 1, \cdots, 0] \in \mathbb{R}^K$, and 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:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{3.10}$$

and $H = [h_1, h_2, \cdots, h_N] \in \mathbb{R}^{C \times N}$ are the class labels of Y_t, where the non-zero element indicates the class of an input signal within each column $h_i = [0, \cdots, 1, \cdots, 0]^T \in \mathbb{R}^C$. Following the same example in (3.10), H can be defined as:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3.11}$$

Scalars α and β are set to control the relative contribution of the terms $\|Q-\vartheta X_t\|^2$ and $\|H-WX_t\|_2^2$. By solving the optimization problem in Equation (3.8), the reconstructive, discriminative and domain-adaptive dictionary pair D_t and D_s as well as an optimal classifier $\mathscr{F}(x)$ can be obtained.

3.1.6 Optimization

Solving WSCDD with the K-SVD algorithm

We rewrite Equation (3.9) as:

$$< D_t', D_s', X_t', A', W' >= \\ arg \min_{D_t, D_s, X_t, A, W} \| \begin{pmatrix} Y_t \\ Y_s \mathbb{A}^T \\ \sqrt{\alpha} Q \\ \sqrt{\beta} H \end{pmatrix} - \begin{pmatrix} D_t \\ D_s \\ \sqrt{\alpha}\vartheta \\ \sqrt{\beta} W \end{pmatrix} X_t \|_2^2, \qquad s.t. \forall i,\ \|x_t^i\|_0 \leq T, \tag{3.12}$$

We further define $Y = (Y_t^T, (Y_s\mathbb{A}^T)^T, \sqrt{\alpha}Q^T, \sqrt{\beta}H^T)^T$, $X = X_t$ and $D = (D_t^T, D_s^T, \sqrt{(}\alpha)\vartheta^T, \sqrt{(}\beta)W^T)^T$, where column-wise l_2 normalization is applied to D, so that optimizing Equation (3.12) is equivalent to optimizing Equation (3.13):

$$< D', X' >= arg \min_{D,X} \|Y - DX\|_F^2, \quad s.t. \forall i,\ \|x_t^i\|_0 \leq T. \tag{3.13}$$

The optimization problem in Equation (3.13) is NP-hard. We apply an approximate solution, K-SVD, [1], which solves exactly the same problem, to Equation (3.13). The dictionary learning problem is solved by iteratively conducting the following two steps until convergence:

- **Computing sparse codes:** by fixing each dictionary D, each sparse code x_i can be computed by

$$x_i = \arg\max_{x_i} \|y_i - Dx_i\|_2^2, \quad \text{s.t.} \forall i, \|x_i\|_0 \leq T. \tag{3.14}$$

  Exact determination of x_i is also NP-hard, thus the greedy-based OMP is applied to approximate the solution.

- **Updating dictionary**: the dictionary D is updated column by column. Define each

row of X as x_T^k (this is to be distinguished from the kth column of X, x_k), a vector w_k, which indicates whether the dictionary atom d_k is used for representing the signal y_i, can then be represented as $w_k = \left\{i | 1 \leq i \leq N, x_T^k(i) \neq 0\right\}$. When updating the dictionary atom d_k, only dictionary atoms, which are from the remaining dictionary $D_{j \neq k}(:,k)$ and with non-zero entries of w_k, are considered[6]. For each column $k = 1, 2, \cdots, K$ in D, a overall reconstruction error matrix E_k can be computed as:

$$E_k = Y - \sum_{j \neq k, w_k^j \neq 0} d_j x_j. \tag{3.15}$$

By applying SVD decomposition to $E_k = U \sum V^T$, the dictionary column $\tilde{d_k}$ can be updated by the first column of U and the coefficient vector $\tilde{x_k}$ can be updated by the first column of V multiplied by $\sum(1,1)$. The dictionary updating step is finished when all columns of the dictionary D are updated.

Convergence Analysis

The convergence proof of the proposed WSCDD method can be given similarly as the K-SVD algorithm [1]. In the dictionary updating stage, each dictionary atom and its corresponding coefficients are updated by minimizing quadratic functions, and the remaining dictionary atoms are updated upon the previous updates. Consequently, the MSE of the overall reconstruction error is monotonically decreasing with respect to the dictionary updating iterations. In the sparse coding stage, computation of the "best matched" coefficients under the l_0-norm constraint also leads to a reduction in MSE conditioned on the success of the OMP algorithm. Finally, since MSE is non-negative, the optimization procedure is monotonically decreasing and bounded by zero from below, thus the convergence of the proposed dictionary learning method is guaranteed. The typical strategy to avoid the optimization procedure getting stuck in a local minimum is to initialize the dictionary with a few different random matrices in several runs. Such a strategy is applied in our approach.

[6] If all remaining dictionary atoms are considered, i.e., all entries of w_k equal to 1, the corresponding new vector x_T^k is very likely to be filled, so that the sparsity constraint does not hold.

3.1.7 Classification

Since D_t, D_s, ϑ and W are jointly normalized in the optimization procedure, they cannot be directly applied to the testing phase. Utilizing the same strategy as in [177], the desired $\tilde{D}_t$, $\tilde{D}_s$, $\tilde{\vartheta}$ and $\tilde{W}$ can be computed as:

$$\begin{aligned}
\tilde{D}_t &= \{\frac{d_t^1}{\|d_t^1\|_2}, \frac{d_t^2}{\|d_t^2\|_2}, \cdots, \frac{d_t^K}{\|d_t^K\|_2}\} \\
\tilde{D}_s &= \{\frac{d_s^1}{\|d_t^1\|_2}, \frac{d_s^2}{\|d_t^2\|_2}, \cdots, \frac{d_s^K}{\|d_t^K\|_2}\} \\
\tilde{\vartheta} &= \{\frac{\vartheta^1}{\|d_t^1\|_2}, \frac{\vartheta^2}{\|d_t^2\|_2}, \cdots, \frac{\vartheta^K}{\|d_t^K\|_2}\} \\
\tilde{W} &= \{\frac{w^1}{\|d_t^1\|_2}, \frac{w^2}{\|d_t^2\|_2}, \cdots, \frac{w^K}{\|d_t^K\|_2}\}
\end{aligned} \tag{3.16}$$

Given a target domain query sample y_t^i, its sparse representation x_t^i can be computed by applying OMP to the original signal y_t^i and the target domain dictionary $\tilde{D}_t$ via Equation 3.14. Then, a simply linear classifier $l = \tilde{W}_t x_t^i$ is applied to x_t^i for classification, where the label l_i^* is the index that corresponds to the largest element in l.

3.1.8 Experiments

Action recognition

The UCF YouTube dataset and the HMDB51 dataset are used for the action recognition task, where the UCF YouTube dataset is used as the target domain and the HMDB51 dataset is used as the source domain. The UCF YouTube dataset is a realistic dataset that contains camera shaking, cluttered background, variations in actors' scale, variations in illumination and view point changes. There are 11 actions including cycling, diving, golf swinging, soccer juggling, jumping, horse-back riding, basketball shooting, volleyball spiking, swinging, tennis swinging and walking with a dog, and these actions are performed by 25 actors. The HMDB51 dataset contains video sequences which are extracted from commercial movies as well as YouTube, and it represents a fine multifariousness of light conditions, situations and surroundings in which actions can appear, different recording camera types and viewpoint changes. Since the HMDB51 dataset is a more challenging dataset, our case closely resembles real-world scenarios, where the source domain data can contain a wide range of noise levels. In correspondence with the target domain action categories, we choose 7 body

movements from the HMDB51 dataset, including ride bike, dive, golf, jump, kick ball, ride horse and shoot ball. Both scenarios of the proposed WSCDDL method when manifold ranking is utilized or not utilized are compared in all the experiments, and are denoted as WSCDDL-MR and WSCDDL-EU respectively.

We adopt the dense trajectories [153], which is extended from the motion coding scheme based on motion boundaries, as the low-level action video representation to distinguish the motion of interest. To leverage the motion information in the dense trajectories, a set of local descriptors are computed within space-time volumes around the trajectories at multiple spatial and temporal scales, and these features include the HOGHOF [60], the optical flow [61] and the Motion Boundary Histogram (MBH) [105]. Specifically, the HOGHOF feature is a combination of appearance information (captured by HOG [22]) and local motion probabilities (captured by Histogram of Optical Flow (HOF)). Since motion is the most important cue for analyzing actions, the optical flow works effectively by computing the relative motion between the observer and the scene. MBH represents the gradient of the optical flow by separately computing the derivatives for the horizontal and vertical components of the optical flow, so that relative motion between pixels is encoded. Changes in the optical flow field being preserved and constant motion information being suppressed, the MBH descriptor can effectively eliminate noise caused by background motion compared with video stabilization [61] and motion compensation [151] approaches [153]. Despite its powerful capability for describing action motions, the dense trajectories come with two weaknesses: 1) trajectories tend to drift from their initial locations during motion tracking, which is a common problem in tracking; 2) the large quantity of local trajectory descriptors leads to high computational and memory complexity for the coding methods, such as VQ and SC. To cope with the first issue, the length of a trajectory is limited to a pre-defined number of frames. Taking the second issue into account, a Locality-constrained Linear Coding (LLC) [68] scheme is adopted instead of VQ and SC. LLC represents the low-level dense trajectories by multiple bases. In addition to achieving less quantization error, the explicit locality adaptor in LLC guarantees the local smooth sparsity.

Dense trajectories are extracted from raw action video sequences with 8 spatial scales spaced by a factor of $1/\sqrt{2}$, and feature points are sampled on a grid spaced by 5 pixels and tracked in each scale, separately. Each point at frame t is tracked to the next frame $t+1$ by median filtering in a dense optical flow field. To avoid the drifting problem, the length of a trajectory is limited to 15 frames. HOGHOF and MBH are computed within a $32 \times 32 \times 15$ volume along the dense trajectories, where each volume is sub-divided into a spatio-temporal grid of size $2 \times 2 \times 3$ to impose more structural information in the represen-

tation. Considering both efficiency and the construction error, LLC coding scheme is applied to the low-level local dense trajectories features with 30 local bases, and the codebook size is set to be 4000 for all training-testing partitions. To limit the complexity, only 200 local dense trajectories features are randomly selected from each video sequence when constructing the codebook. We run our method on five different partitions of the UCF YouTube dataset, where we randomly choosing all action categories performed by the number of 5/9/16/20/24 actors as the training actions while using the remaining actions as the testing actions for each partition. The 30 most relevant actions are chosen from each of the 7 source domain categories using manifold ranking, and they are represented in the same manner as the target domain actions and coded with the same codebook. The weight α on the label constraint term and the weight β on the classification error term are set as 4 and 2 respectively, and 50 iterations of SVD decomposition are executed during optimization. To avoid over-fitting, the dictionary size is set to be larger when more training data are available at the training stage. The results are demonstrated in Table 3.1 for all five partitions, where we use the size of 200, 300, 500, 700 and 900 for each partition. We compare the performance of the baseline LLC, sparse coding methods K-SVD [1] and LC-SVD [66], and transfer learning methods FR [24] and A-SVM [167] with the proposed WSCDDL method. Results are reported on both scenarios where the source domain data are included or excluded in Table 3.1 and Table 3.2 respectively. By comparing Table 3.1 and Table 3.2, we can discover that for many cases, brute-forcing the knowledge from the source domain into the target domain irrespective of their divergence can cause certain performance degeneration. On the other hand, the proposed WSCDDL method consistently leads to the best performance over all the partitions. Fig. 3.5 shows the convergence analysis and performance of varying dictionary size of the WSCDDL-MR method. Fig. 3.6 shows the confusion matrix comparisons between the LLC method and the WSCDDL method for all five partitions. In order to compare the WSCDDL method with state-of-the-art methods, we further demonstrate its performance under the leave-one-actor-out setting in Table 3.3.

Image Classification

We utilize the Caltech101 dataset as the target domain and some collected Web images as the source domain for the image classification task. The Caltech101 image dataset (shown in Fig. 3.4) consists of 101 categories (e.g., accordion, cannon, chair), and each category contains 30 to 800 images. The source domain data of the Caltech101 dataset are constructed by a set of images returned by Google Image Search (shown in Fig. 3.4). For each

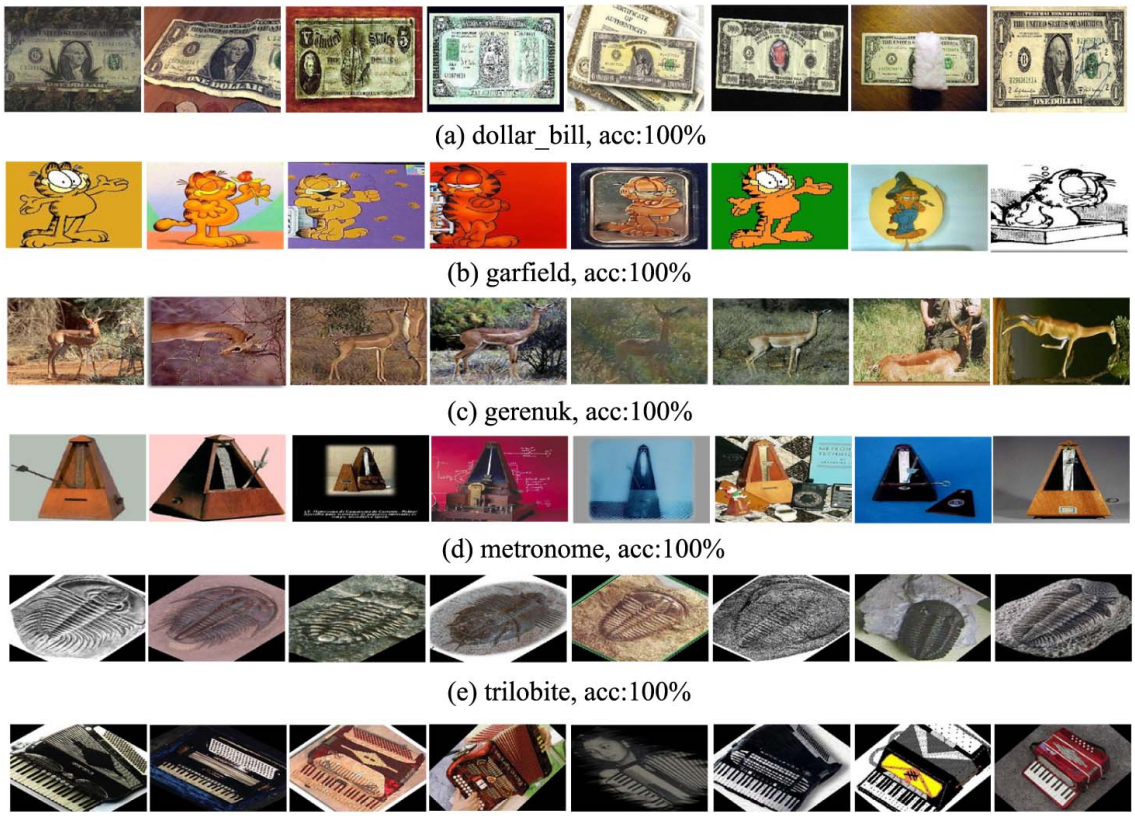


Fig. 3.4 Example images from classes with high classification accuracy from the Caltech101 dataset.

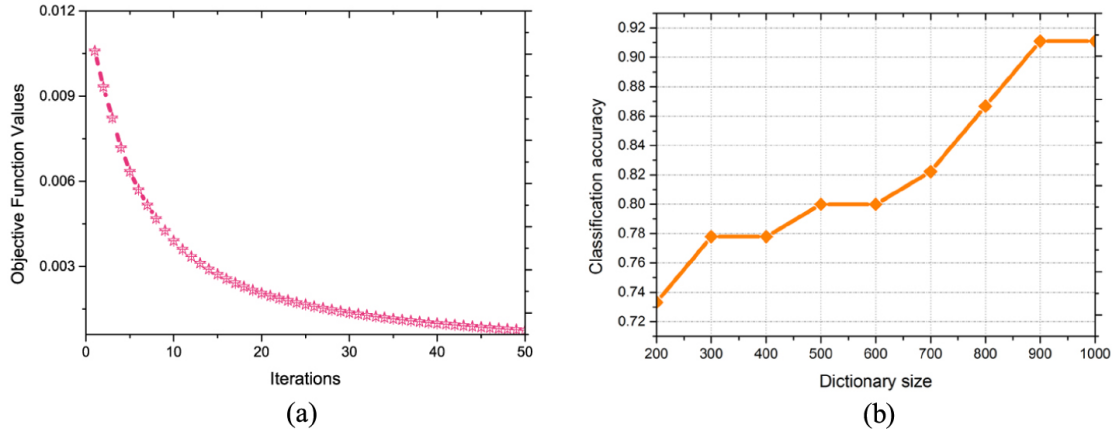


Fig. 3.5 Performance analysis on the UCF YouTube dataset when actions performed by 24 actors are used in the training data. (a) The optimization process of the objective function for WSCDDL-MR with 50 iterations. (b) Performance when varying the dictionary size.

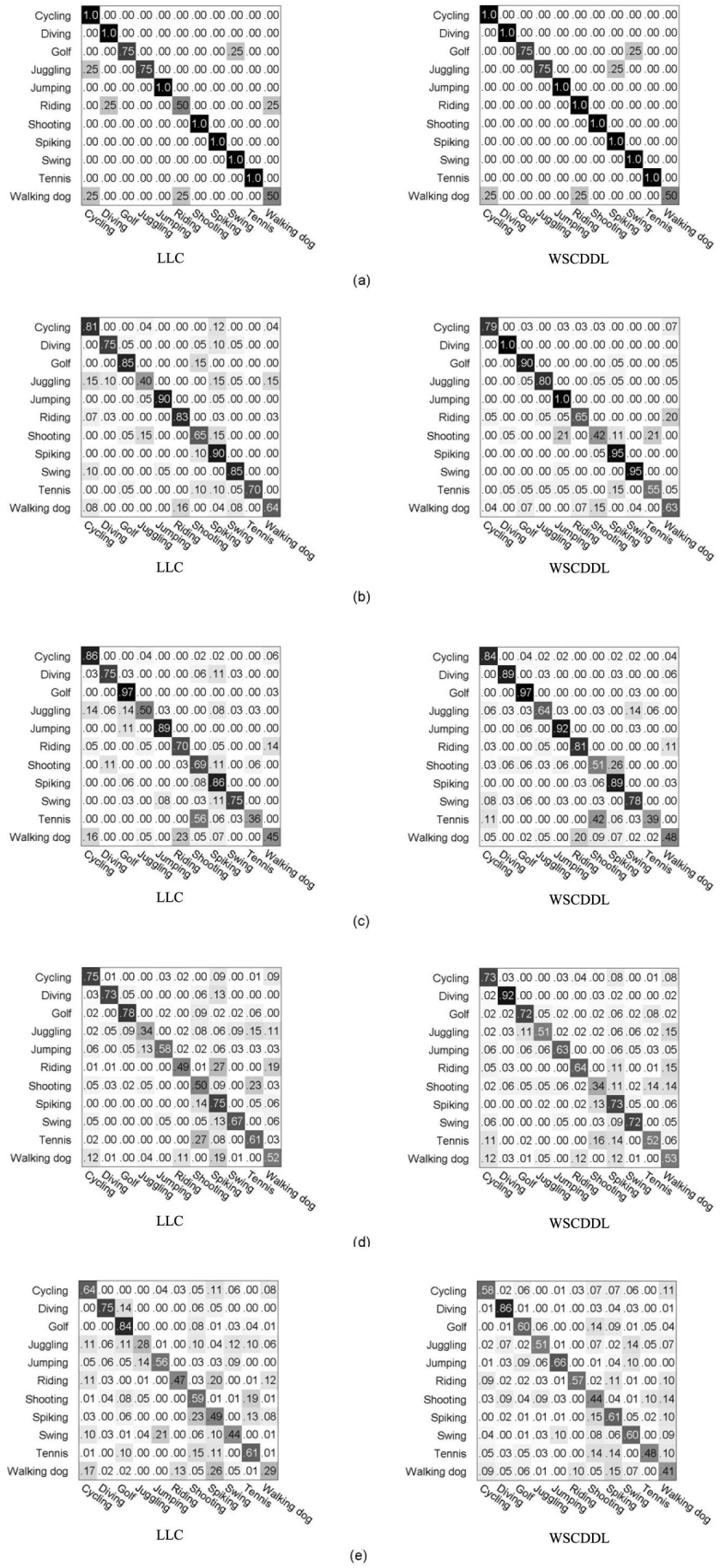


Fig. 3.6 Comparison of the confusion matrixes between the baseline ScSPM and the WSCDDL on five different data partitions of the UCF YouTube dataset.

Table 3.1 Performance comparison between the WSCDDL and other methods on the UCF YouTube dataset when the source domain data are only used by the WSCDDL.

Algorithm	LLC [68]	K-SVD [1]	LC-KSVD [66]	WSCDDL-EU	WSCDDL-MR
Dictionary Learning	*N/A*	*Unsupervised*	*Supervised*	*Supervised*	*Supervised*
Source data	*No*	*No*	*No*	*Yes*	*Yes*
24 actors	86.67%	82.22%	86.67%	88.89%	**91.11**%
20 actors	75.42%	68.75%	75.42%	77.50%	**78.30**%
16 actors	70.88%	63.96%	72.08%	73.03%	**73.03**%
09 actors	61.41%	55.70%	65.25%	66.31%	**66.05**%
05 actors	54.10%	50.05%	56.55%	56.66%	**57.19**%

Table 3.2 Recognition results on the UCF YouTube dataset when using the HMDB dataset as the source domain.

Algorithm	LLC [68]	K-SVD [1]	LC-KSVD [66]	FR [24]	A-SVM [167]	WSCDDL-EU	WSCDDL-MR
Dictionary Learning	*N/A*	*Unsupervised*	*Supervised*	*Supervised*	*Supervised*	*Supervised*	*Supervised*
Source data	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
24 actors	86.67%	77.78%	82.22%	83.74%	82.51%	88.89%	**91.11**%
20 actors	70.21%	72.08%	75.42%	74.88%	79.05%	77.50%	**78.30**%
16 actors	70.17%	67.54%	72.08%	71.56%	72.46%	73.03%	**73.03**%
09 actors	61.80%	59.15%	64.72%	62.77%	61.65%	66.31%	**66.05**%
05 actors	53.35%	48.88%	54.10%	54.09%	51.54%	56.66%	**57.19**%

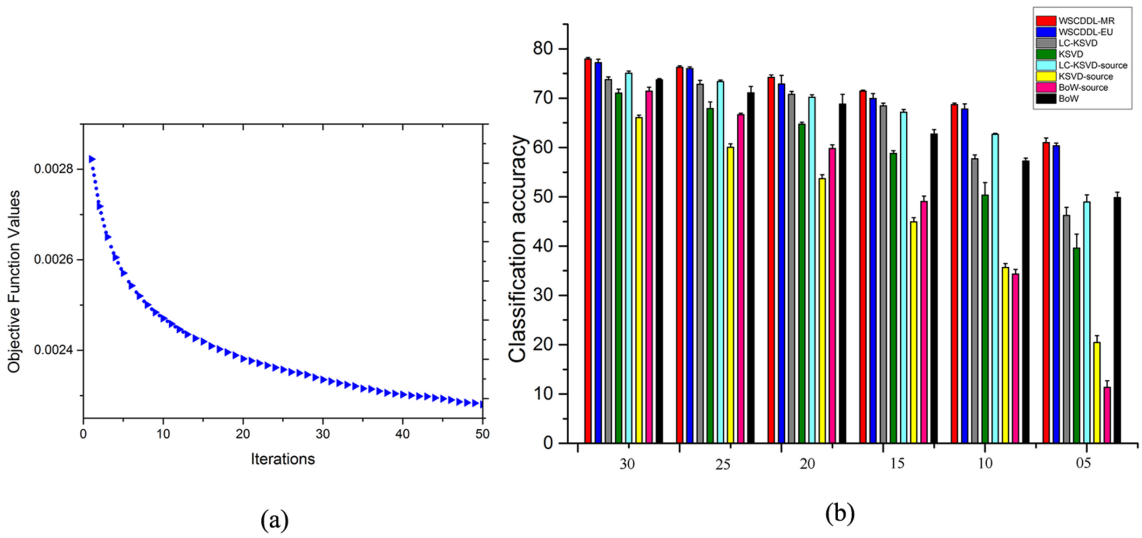
Table 3.3 Performance comparison of the WSCDDL with state-of-the-art methods under the leave-one-actor-out setting on the UCF YouTube dataset.

Methods	[86]	[61]	BoF	WSCDDL-EU	WSCDDL-MR
Results	71.2%	75.21%	80.02%	81.13%	**82.32**%

of the 20 selected categories, we retrieve a set of images by querying the category name into Google Image Search, and randomly choose 20-30 images from the first 100 retrieved images. Such a procedure is done manually, and only 5 retrieved images are considered as labeled for each category.

The combination of local features and the Bag-of-Words (BoW) model has demonstrated its effectiveness in previous art and is a major component of many state-of-the-art systems. To overcome the shortcoming that structural relationships among local descriptors are discarded by the BoW model, the Spatial Pyramid Matching (SPM) method was proposed for image representations in [77]. However, in order to obtain good performance, both BoW and SPM must be applied along with a particular type of nonlinear Mercer kernels, which lead to high computational complexity $O(n^3)$ and memory usage $O(n^2)$. We represent images in both the target domain and the source domain with the sparse coding based spatial-pyramid image representation [168], which can be seen as an extension of the SPM. The SIFT descriptors [94] extracted from different spatial scales of an image are first encoded according to an overcomplete codebook. With a unit $l2$-norm constraint on the cluster centers, the restrictive cardinality constraint of K-means Vector Quantization (VQ) in the traditional SPM is relaxed. Instead of performing spatial pooling by computing histograms in the original SPM, the *max* spatial pooling method, which is more biologically meaningful and more robust in representing local spatial relationships, is applied. Such a sparse coding based SPM (ScSPM) image representation captures more salient properties of visual patterns and leads to promising results when working with linear SVMs, so that the training complexity can be reduced to $O(n)$.

Following the settings in [168], the SIFT descriptors are extracted from 16×16 pixel patches and densely sampled from each image on a grid with the step size of 8 pixels. The codebook is trained using sparse coding with the codebook size of 1024. Through the ranking procedure, 10 most relevant images are chosen to build the source domain of each image category. The same values of the weights α, β and K-SVD iterations are adopted as in the action recognition task. Similarly, we compare the performance of the baseline ScSPM [168], K-SVD [1] and LC-SVD [66] with the proposed WSCDDL method in Table 3.4 and Table 3.5. Fig. 3.4 shows samples of 6 categories with high classification accuracies when using 30 training images per category. Results on six different numbers of training data are reported, and all the results are obtained from 5 iterations of different randomly selected training and testing images to guarantee the reliability. As shown in Fig. 3.7, the proposed WSCDDL method results in larger improvements over others when fewer samples are used for training, which demonstrates its effectiveness in terms of utilizing the source



(a) (b)

Fig. 3.7 Performance analysis on the Caltech101 dataset. (a) The optimization process of the objective function for WSCDDL-MR with 50 iterations. (b) Means and standard deviations of different methods when the number of training samples per class varies from 5 to 30.

domain data. Fig. 3.8 demonstrates the performance of all the 101 image categories. We further compare our approach with state-of-the-art methods in Table 3.6. For all scenarios, our approach consistently yields the best performance.

We additionally evaluate our methods on the more challenging Caltech 256 dataset [54], which contains $30,607$ images of 256 categories. Compared to the Caltech101 dataet, it is much more difficult due to the large variations on object location, pose, and size. Similar to the strategy we adopt in constructing the source domain for the Caltech101 dataset, 400 images from 20 categories indexed by Google Images are used as the source domain. We evaluate our approach on both 15 and 30 training images per class and compare with K-SVD [1], SRC [161], LLC [68] and state-of-the-art approaches [54], [168]. As shown in Table 3.7, our approach consistently leads the best performance. Fig. 3.9 shows samples from 5 categories with high classification accuracies when using 30 images per category.

Event Recognition

We compare our proposed method WSCDDL with state-of-the-art transfer learning methods on the event recognition task using the Kodak Consumer Videos and a set of additional videos. The Kodak consumer video benchmark dataset was collected by Kodak from about 100 real users over the period of one year, and it includes two video subsets from two dif-

Table 3.4 Performance comparison between different dictionary learning methods on the Caltech101 dataset.

Algorithm	ScSPM [168]	K-SVD [1]	LC-KSVD [66]	WSCDDL-EU	WSCDDL-MR
Dictionary Learning	*N/A*	*Unsupervised*	*Supervised*	*Supervised*	*Supervised*
Source data	*No*	*No*	*No*	*Yes*	*Yes*
05 training	$49.84 \pm 1.13\%$	$39.63 \pm 2.81\%$	$46.25 \pm 1.63\%$	$60.33 \pm 0.58\%$	$\mathbf{61.01 \pm 0.94}\%$
10 training	$57.26 \pm 0.63\%$	$50.3a \pm 2.54\%$	$57.73 \pm 0.77\%$	$67.79 \pm 1.07\%$	$\mathbf{68.69 \pm 0.32}\%$
15 training	$62.72 \pm 0.93\%$	$58.82 \pm 0.54\%$	$68.45 \pm 0.53\%$	$69.95 \pm 0.97\%$	$\mathbf{71.44 \pm 0.18}\%$
20 training	$68.78 \pm 2.00\%$	$64.73 \pm 0.38\%$	$70.79 \pm 0.58\%$	$72.88 \pm 1.76\%$	$\mathbf{74.24 \pm 0.50}\%$
25 training	$71.12 \pm 1.29\%$	$67.92 \pm 1.31\%$	$72.83 \pm 0.80\%$	$76.03 \pm 0.34\%$	$\mathbf{76.27 \pm 0.31}\%$
30 training	$73.72 \pm 0.26\%$	$71.04 \pm 0.79\%$	$73.75 \pm 0.55\%$	$77.23 \pm 0.67\%$	$\mathbf{78.04 \pm 0.26}\%$

Table 3.5 Classification results on the Caltech101 dataset when using web images as the source domain.

Algorithm	ScSPM [168]	K-SVD [1]	LC-KSVD [66]	WSCDDL-EU	WSCDDL-MR
Dictionary Learning	*N/A*	*Unsupervised*	*Supervised*	*Supervised*	*Supervised*
Source data	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
05 training	$11.33 \pm 1.37\%$	$20.42 \pm 1.40\%$	$48.95 \pm 1.45\%$	$60.33 \pm 0.58\%$	$\mathbf{61.01 \pm 0.94}\%$
10 training	$34.31 \pm 0.93\%$	$35.64 \pm 0.82\%$	$62.71 \pm 0.20\%$	$67.79 \pm 1.07\%$	$\mathbf{68.69 \pm 0.32}\%$
15 training	$49.08 \pm 1.06\%$	$44.93 \pm 0.86\%$	$67.14 \pm 0.59\%$	$69.95 \pm 0.97\%$	$\mathbf{71.44 \pm 0.18}\%$
20 training	$59.80 \pm 0.73\%$	$53.69 \pm 0.77\%$	$70.17 \pm 0.50\%$	$72.88 \pm 1.76\%$	$\mathbf{74.24 \pm 0.50}\%$
25 training	$66.68 \pm 0.28\%$	$60.07 \pm 0.70\%$	$73.39 \pm 0.27\%$	$76.03 \pm 0.34\%$	$\mathbf{76.27 \pm 0.31}\%$
30 training	$71.46 \pm 0.78\%$	$66.07 \pm 0.50\%$	$75.05 \pm 0.47\%$	$77.23 \pm 0.67\%$	$\mathbf{78.04 \pm 0.26}\%$

Table 3.6 Comparison with the state-of-the-art methods on the Caltech101 dataset.

Number of training samples	5	10	15	20	25	30
Malik [176]	46.6%	55.8%	59.1%	62.0%	–	66.2%
Griffin [54]	44.2%	54.5%	59.0%	63.3%	65.8%	67.6%
SRC [161]	48.8%	60.1%	64.9%	67.7%	69.2%	70.7%
Wang [68]	51.15%	59.77%	65.43%	67.74%	70.16%	73.44%
WSCDDL-EU	**60.33**	**67.79**%	**69.95**%	**72.88**%	**76.03**%	**77.23**%
WSCDDL-MR	**61.01**	**68.69**%	**71.44**%	**74.24**%	**76.27**%	**78.04**%

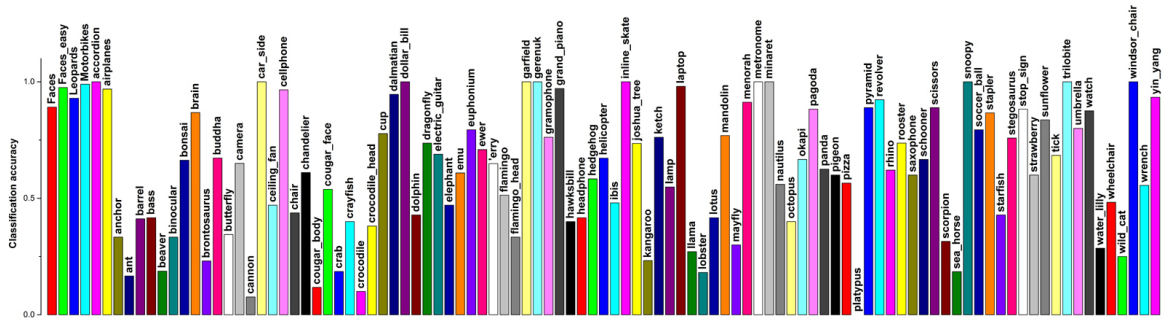


Fig. 3.8 Performance on all the categories of the Caltech101 dataset achieved by the WSCDDL-MR method when using 30 training images per category.

Table 3.7 Recognition results on the Caltech256 dataset.

Number of training samples	15	30
Griffin [54]	28.3%	34.10%
Yang [168]	27.73%	34.02%
K-SVD [1]	25.33%	30.62%
SRC [161]	27.86%	33.33%
LLC [68]	25.61%	30.43%
LC-KSVD [66]	28.9%	34.32%
WSCDDL-EU	29.68%	35.78%
WSCDDL-MR	30.14%	36.07%

ferent sources, where the first part contains Kodak's video data which includes 1,358 video clips contributed by involved users and the second part contains 1,873 clips downloaded from the YouTube website after removing TV commercial videos and low-quality videos. Similarly, the additional videos collected by Duan et al. in [75] also contain two parts, which are the self-collected consumer videos and downloaded YouTube videos. To resemble the real-world scenario, the downloaded YouTube videos are not additionally annotated so that they can remain in a loosely labeled setting. Thus, only the self-collected consumer videos from the dataset used in [75] possess precise labels. The total numbers of consumer videos and YouTube videos are 195 and 906, respectively, and each video belongs to only one event category. Following the settings in [75], six events, namely "birthday", "picnic", "parade", "show", "sports" and "wedding" are chosen for experiments. The target domain is constructed using both the consumer videos from the Kodak dataset and additional self-collected consumer videos in [75]. On the other hand, the second part of the Kodak dataset and the loosely labeled YouTube videos used in [75] constitute the source domain. In the target domain, three consumer videos from each event (18 videos in total) are randomly chosen as the labeled training videos and the remaining videos are used as the test data. In order to set up a fair comparison in correspondence with the experimental results in [75], we

Fig. 3.9 Example images of the categories with high classification accuracy from the Caltech256 dataset.

use the same low-level features, which are SIFT features and ST features. For each sampled frame, which is sampled at the sampling rate of 2 frames per second, the 128-dimensional SIFT features are extracted from the salient regions, which are detected by the Difference-of-Gaussians (DoG) interest point detector [94]. The 162-dimensional local ST feature is the concatenation of the 72-dimensional HOG feature and the 90-dimensional HOF feature. We also conduct experiments in the same three cases as in [75]: a) dictionaries and classifiers are learned based on SIFT features, b) dictionaries and classifiers are learned based on ST features and c) dictionaries and classifiers are learned on both SIFT and ST features. Based on the same experimental settings as in [75], we compare our method WSCDDL with SVM-AT, SVM-T, FR [24], A-SVM [167], MKL [30], DTSVM [30] and A-MKL [75], where SVM-AT denotes the case that labeled training samples are obtained from both the target domain and the source domain, and correspondingly SVM-T denotes the case that labeled training samples are only obtained from the target domain. Table 3.8 demonstrates the recognition results of the proposed WSCDDL method and other cross-domain methods. We can observe that SVM-T consistently outperforms SVM-AT in both scenarios of (b) and (c), which indicates that abruptly including the ST features of source domain videos may degrade the recognition performance. The proposed WSCDDL method consistently

Table 3.8 Comparison with the state-of-the-art methods on the Kodak and YouTube dataset.

	SVM-T	SVM-AT	FR [24]	A-SVM [167]	MKL [30]	DTSVM [30]	A-MKL [75]	WSCDDL-Eu	WSCDDL-MR
(a)	42.32 ± 5.50	53.93 ± 5.58	49.98 ± 5.63	38.42 ± 7.93	47.19 ± 2.59	52.36 ± 1.88	47.14 ± 2.34	57.18 ± 0.84	58.42 ± 2.25
(b)	32.56 ± 2.08	24.73 ± 2.22	28.44 ± 2.61	24.95 ± 1.25	35.34 ± 1.55	31.07 ± 2.60	37.24 ± 1.58	37.80 ± 1.77	39.11 ± 2.76
(c)	42.00 ± 4.94	36.23 ± 3.37	44.11 ± 3.57	32.40 ± 4.99	46.92 ± 2.53	53.78 ± 2.99	58.20 ± 1.87	61.92 ± 2.89	62.60 ± 1.76

outperforms other cross-domain methods in all three cases.

3.1.9 Conclusion

In this work, we have presented a novel visual categorization framework using the weakly-supervised cross-domain dictionary learning algorithm. Auxiliary domain knowledge is utilized to span the intra-class diversities, so that the overall performance of the original system can be improved. The proposed framework only requires a small set of labeled samples in the source domain. Through a transformation matrix, dictionary learning is performed on both the source domain data and the target domain data while no correspondence annotations between the two domains are required. Promising results are achieved on action recognition, image classification and event recognition tasks, where knowledge from either the Web or a related dataset is transferred to standard benchmark datasets. The proposed framework leads to an interesting topic for future investigation when large scale source and target domain data are available.

3.2 Boosted Cross-Domain Dictionary Learning[7]

3.2.1 Motivation

Based on the recent success of dictionary learning methods in solving computer vision problems, we present a cross-domain discriminative dictionary learning technique to learn a reconstructive, discriminative and domain-adaptive dictionary pair for data under different distributions. In addition, a boosted classification framework is introduced to work in conjunction with the proposed dictionary learning method. Through iteratively updating both the source domain data representations and their distribution, the source domain training

[7]The content of this section is published at:
F. Zhu, L. Shao and J. Tang, Boosted Cross-Domain Categorization, British Machine Vision Conference, Nottingham, UK, Sep. 2014.

instances can be optimized, and thus can help improve the visual categorization tasks in the target domain. A weakly-supervised cross-domain visual categorization framework that unifies the discriminative cross-domain dictionary learning method and the boosting-based classification method is proposed in this work. Our goals are two folds: 1) we aim to learn robust domain-adaptive representations for cross-domain image and video data; 2) we aim to learn powerful classifiers for accurate classification.

3.2.2 Related Work

Adaptive Boosting (AdaBoost) [42] is a popular boosting algorithm, which has been used in conjunction with a wide range of other machine learning algorithms to enhance their performance, e.g., Shen et al. [139] applied AdaBoost to Gabor wavelet features and Fathi et al. [35] used AdaBoost to construct mid-level shape features from low-level gradient features. The Transfer Learnning AdaBoost (TrAdaBoost) was introduced in [21] to extend AdaBoost for transfer learning by weighting less on the different-distribution data which are considered as "dissimilar" to the same-distribution data in each boosting iteration. Fig. 3.10 illustrates how TrAdaBoost deals with cross-domain data. Given a set of 2-class target domain data and source domain data, the left sub-figure shows the decision boundary of a traditional linear classifier, and the right sub-figure shows both the updated weights allocated to incorrectly labeled samples and the new decision boundary according to the updated weights. In each iteration of TrAdaBoost optimization process, more weights are allocated to those incorrectly labeled (according to the classifier in the previous iteration) target domain samples, so that the updated classifier is tuned to pay more attentions to the target domain "hard" samples, while on the other hand, the source domain incorrectly labeled samples are allocated less weights or simply removed (for clearer illustration, we use black crosses to denote that these samples are removed) to avoid "bad" source domain samples.In comparison with TrAdaBoost, our basic motivation is that rather than removing those unsmooth data, we prefer moving them to more appropriate positions. Thus, we aim to utilize the above data correctness information to learn better representations that are closer to the intrinsic data partition. Moreover, different from the majority transfer learning frameworks, our model also does not assumes that the target domain data are perfectly smooth.

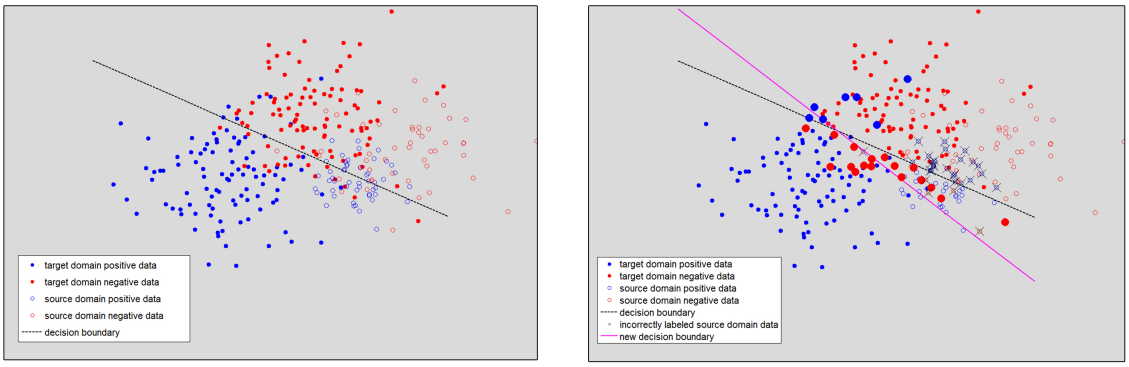


Fig. 3.10 Illustration of how TrAdaBoost deals with cross-domain data. Given a set of 2-class target domain data and source domain data, the left sub-figure shows the decision boundary of a traditional linear classifier, and the right sub-figure shows both the updated weights allocated to incorrectly labeled samples and the new decision boundary according to the updated weights, where the increased marker size denotes increased weight assigned to a target domain sample, and the black cross denotes that an incorrectly labeled source domain sample is removed.

3.2.3 Boosted Cross-Domain Dictionary Learning

We firstly define some notations according to the problems we dealing with. Given raw input image and video data, SIFT and dense trajectories features are extracted from images and videos, respectively, followed by which the SPM model and the BoW model are applied to the extracted low-level features for global representations. Without loss of generality, we denote such global representations for both images and videos as Y. Specifically, we denote $Y_t = [y_t^1, y_t^2, ..., y_t^N] \in \mathbb{R}^{n \times N}$ as n-dimensional target domain training data, $Y_s = [y_s^1, y_s^2, ..., y_s^M] \in \mathbb{R}^{n \times M}$ as n-dimensional source domain data, $X_t = [x_t^1, ...x_t^N] \in \mathbb{R}^{K \times N}$ as the target domain sparse coefficients, $X_s = [x_s^1, ...x_s^N] \in \mathbb{R}^{K \times N}$ as the source domain sparse coefficients, $D_t = [d_t^1, ...d_t^K] \in \mathbb{R}^{n \times K_t}$ as the target domain dictionary and $D_s = [d_s^1, ...d_s^K] \in \mathbb{R}^{n \times K_s}$ as the source domain dictionary, where K is the number of dictionary atoms for dictionaries in both domains. Since the dictionary learning problem in this work is essentially the same as the dictionary learning problem in Section 3.1.5, we use the same objective function as

in Equation 3.9 to describe the learning problem:

$$\begin{aligned} < D_t', D_s', X_t', A', P' > = arg \min_{D_t, D_s, X_t, A, P} & \|Y_t - D_t X_t\|_2^2 + \|Y_s \mathbb{A}^T - D_s X_t\|_2^2 \\ & + \alpha \|Q - \vartheta X_t\|^2 + \beta \|H - P X_t\|_2^2 \\ & s.t. \forall i, \ \|x_t^i\|_0 \leq T, \end{aligned} \tag{3.17}$$

where both $\mathbb{A}$ and ϑ are the transformation matrices and P contains the parameters of a linear classifier. More detailed explanations of these notations can be found in Section 3.1.5. Q is the target discriminative sparse codes, where $q_i = [q_i^1, q_i^2, \cdots, q_i^K]^T = [0, \cdots, w_i, w_i, \cdots, 0]^T \in \mathbb{R}^K$, and 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. The definition of Q can be written as:

$$\begin{pmatrix} w_1 & w_2 & 0 & 0 & 0 & 0 \\ w_1 & w_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & w_4 & 0 & 0 \\ 0 & 0 & w_3 & w_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & w_6 \\ 0 & 0 & 0 & 0 & w_5 & w_6 \end{pmatrix}, \tag{3.18}$$

and $H = [h_1, h_2, \cdots, h_N] \in \mathbb{R}^{C \times N}$ are the class labels of Y_t, where the non-zero element indicates the class of an input signal within each column $h_i = [0, \cdots, 1, \cdots, 0]^T \in \mathbb{R}^C$. Since predictions are made with respect to the data distribution of X_t, w_i is included in each item of H. Thus H can be defined as follows according to the same example in Equation (3.18)

$$\begin{pmatrix} w_1 & w_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & w_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & w_6 \end{pmatrix}. \tag{3.19}$$

The definitions of $W = [w_1, w_2, \cdots, w_N]^T$ are given in the following subsection. Since the optimization problem of Equation 3.17 can be solved using the method as in Equation 3.9, we do not repeat here. Please refer to Section 3.1.6 for details of the optimization stage.

3.2.4 Boosted Classification

Boosting Algorithms

AdaBoost [42] is a classical machine learning algorithm which 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:

$$\mathscr{F}(x) = \sum_{i=1}^{T} \gamma_i f_i(x), \tag{3.20}$$

where each $f_i(\cdot)$ represents a weak classifier. Any $f_i(\cdot)$ 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 weight updating mechanism, Zhang et al. [178] attempted to capture more discriminative information by learning a set of codebooks in sequence. As an extension to AdaBoost, Dai et al. [21] 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, while TrAdaBoost stays at the classifier level, it fails to update the data towards more robust and discriminative representations through the learning process.

We propose a cross-domain learning framework to effectively utilize data under a different feature distribution for the classification task 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 in addition to updating their weights.

Boosted Classification

Similar as TrAdaBoost [21], we consider the similarities between the source domain training instances Y_s and the target domain training instances Y_t according to the present distri-

Algorithm 3 Boosted Cross-Domain Dictionary Learning

Input the labeled target domain data Y_t and the source domain data Y_s , the maximum number of iterations M_a and the Weak Learner.

Output a "strong" classifier $\mathscr{F}(\cdot)$ 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 Y_t and Y_s can be represented by X_t and X_s^1 respectively.

for $j = 1$ to M_a **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 Y_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 $h_s^j : X_s^j \to l(X_s)$, providing that p^j is over both $\mathscr{D}_l^t$ and $\hat{\mathscr{D}}^s$.
4. Calculate the error ε^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 ε^j is required to be less than 0.5.
5. Set $\beta_t^j = \frac{\varepsilon^j}{1-\varepsilon^j}$ and $\beta_s = \frac{1}{1+\sqrt{2\ln M/M_a}}$
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

bution. 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 source domain 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 binary label (either 0 or 1) of instance x_i and $h_s^j(x_i)$ is the binary output (either 0 or 1) of the weak classifier at iteration j, 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" instances towards more appropriate representations. The confidence of the discriminative term is measured by allocating weights to different training instances, 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 source domain training instances in 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 consistent with the original AdaBoost [42] by multiplying the factor $\beta_t^{j-|h_t^j(x_i)-l(x_i)|}$, so that the weights of those incorrectly classified target domain instances in are increased in order to make the new classifier focus on those instances in the next iteration. The base of source domain exponent weight factor, β_s, is a constant, and is determined by the number of source domain instances M and the maximum number of iterations M_a. 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.

The Classification and Regression Trees (CART) [12] 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. Let the "error of leaf" be the probability of a instance being misclassified at a leaf, the construction of the whole tree follows the following steps:

1. Construct a root node.

2. Choose the leaf with the largest error.

3. Construct a node using only those training instances associated with the chosen leaf.

4. Replace the chosen leaf with the constructed node.

5. Repeat steps 2-4 until the total error is zero, or the maximum iteration is reached.

All the errors are evaluated according to the updated weights, so that the training instances can be learned with respect to their present distribution.

3.2.5 Experiments

3.2.6 Parameter settings

The experiments are conducted on both image classification (Caltech 101 dataset) and action recognition (UCF YouTube dataset) tasks. For the image classification task, the SIFT+SPM model is utilized as the initial image presentation, with the codebook size 1024 and 3 pyramid scales, which results in a $(4 \times 4 + 2 \times 2 + 1) \times 1024 = 21504$ dimensional feature for each image, i.e., $N = 21504$. For the action recognition task, local dense trajectories features are projected to a learned codebook (using K-means clustering) using LLC, so that the global feature length equals to the codebook size, which is 1024, i.e., $N = 1024$. For both tasks, the dictionary size K_t and K_s are fixed to 300, and the sparsity T is fixed to 10.

Image classification

We adopt the dense SIFT descriptors plus the sparse coding approach [168] 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) [168], K-Singular Value Decomposition (K-SVD) [1], Label Consistent-Singular Value Decomposition (LC-KSVD) [67], AdaBoost [42], and Weakly Supervised Cross-Domain Dictionary Learning (WSCDDL) [184] [8] and Transfer AdaBoost (TrAdaBoost) [21]. Experimental results are reported in Table 3.9 and Table 3.10 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 3.9 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

[8] 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 report. On the other hand, results are reported for all image categories in [184].

domain data are used, which further validates the importance of our boosted cross-domain categorization method. Fig. 3.11 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.

Action recognition

We extract the dense trajectories [153] as local features from raw action videos and project local features on a codebook using Locality Constrained Linear Coding (LLC) [68]. We run 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[9] in Table 3.11 and Table 3.12 when source domain data are included or not respectively. The reported results of LLC, K-SVD and LC-KSVD in Table 3.11 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 3.12.

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.

[9]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 [184].

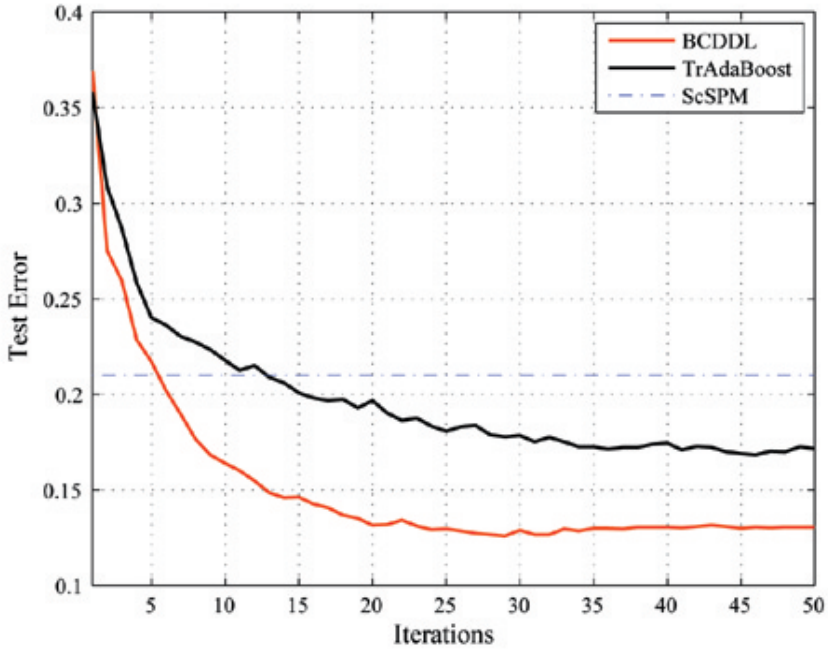


Fig. 3.11 Error rate comparison of the proposed method with TrAdBoost and ScSPM on the Caltech 101 dataset.

Table 3.9 Performance comparison between the BCDC and state-of-the-art methods on the Caltech-101 dataset with source domain data.

Algorithm	ScSPM [168]	K-SVD [1]	LC-KSVD [67]	TrAdaBoost [21]	WSCDDL [184]	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**%

Table 3.10 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 [168]	K-SVD [1]	LC-KSVD [67]	AdaBoost [42]	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**%

Table 3.11 Performance comparison between the BCDC and state-of-the-art methods on the UCF YouTube dataset with source domain data.

Algorithm	LLC [68]	KSVD [1]	LC-KSVD [67]	TrAdaBoost [21]	WSCDDL [184]	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**%

Table 3.12 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 [68]	KSVD [1]	LC-KSVD [67]	AdaBoost [42]	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**%

3.2.7 Conclusion

In this work we have presented a novel cross-domain learning framework for visual categorization tasks. A cross-domain discriminative dictionary learning method is proposed to work in conjunction with a boosted cross-domain classification algorithm, so that the source domain data are adapted to the target categorization tasks through both their feature representation and distribution update. 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.

3.3 Cross-Modality Neural Network

3.3.1 Motivation and Introduction

Over the last decade, social networking services (e.g., Facebook, YouTube and Flickr) are getting wildly popularized. As a consequence, we have witnessed the explosion of multimedia data on the Internet. Information retrieval techniques are applied to multimedia data to collect information in response to people's interests. While classical approaches [134], [147], [149], [92], [148] , which only utilize a unimodal representation, e.g., text, image, audio, etc., are considered somewhat outdated, multimodal systems [142], [23], [165] become popular for information retrieval. The majority of these works only consider the scenario where multimodal data are available for both training and testing, which, however, cannot well address the problem when training and query instances come from different media forms. Thus, as a sub-topic of transfer learning [136], the cross-modality scenario is proposed to fill such a gap, where the database contains data in one modality and the query instances are expressed in the other modality/modalities. The cross-modality scenario can find many real-world applications, including enabling the machine to allocate a description of a few sentences to a query image, or to search over an image repository for a set of relevant matches in response to the query text description. From another research perspective, we may mine useful knowledge from relevant data in a different media form. Thus, by utilizing the mined information, which can be presented in a more discriminative manner, we look forward to improved capability for understanding the target queries. To this end, either labeled or unlabeled data from a relevant media domain can be utilized for enhancing an existing learning system.

In order to provide readers a straightforward understanding of how the data that we are

dealing with is presented in different media forms, examples are given in the following part. We show three examples with one-to-one image/text correspondence from the Wikipedia dataset [20] in Fig. 3.12. All these three examples are obtained from the "sport" category. (a) the first image describes that the famous NBA basketball player Michael Jordan attempts dunking in a game, and the text paragraph attached at the side of the image introduces Michael Jordan's career achievements; (b) the middle image contains the head and shoulders of the professional cricket player Chris Morris, and similar as (a), the text paragraph attached beside the image introduces Chris Morris's career achievements; (c) the last image contains the upper body part of the American football player Merio Danelo, and the attached text paragraph introduces stories of the USC Trojans team. It can be observed that the three image examples are distributed with high intra-class variations, in particular the latter two images can hardly be categorized as "sports", if the observer have never watched games played by these athletes. However, on the other hand, when the same data present in the text domain, more representative characteristics can be extracted based on shared pivot words, e.g., "championship", "season", "won", which have relatively higher probabilities of belonging to sports-related descriptions. The majority of existing approaches that deal with image-text retrieval tasks treat such a cross-modality task in a rough manner by associating a whole image and a whole paragraph of text descriptions as a cross-modality data pair, so that retrieval is achieved at such a global level. In this work, our approach also follows such determinations of how image and text data are associated.

In this work, we propose a neural network-based approach to address the cross-modality problem. Specifically, we train two cross-modality autoencoders (CMAE) to map image representations and text representations to a unified feature space, and conduct cross-modality retrieval within the new feature space. By setting identical random vectors for data of the class as the outputs of the neural network, when data pass through the learned network, insignificant intra-class distances can be obtained. Though the training of these neural networks are simple, experimental suggest that the proposed method can achieve state-of-the-art performance on the Wikipedia dataset.

3.3.2 Related Work

Content-based information retrieval has been an important subject in multimedia, and it has received much attention in the last decade. Many previous content-based retrieval techniques are based on single-modality data, which can be images [40], [173], texts [128], or other media forms. In the task of single-modality information retrieval, the query data

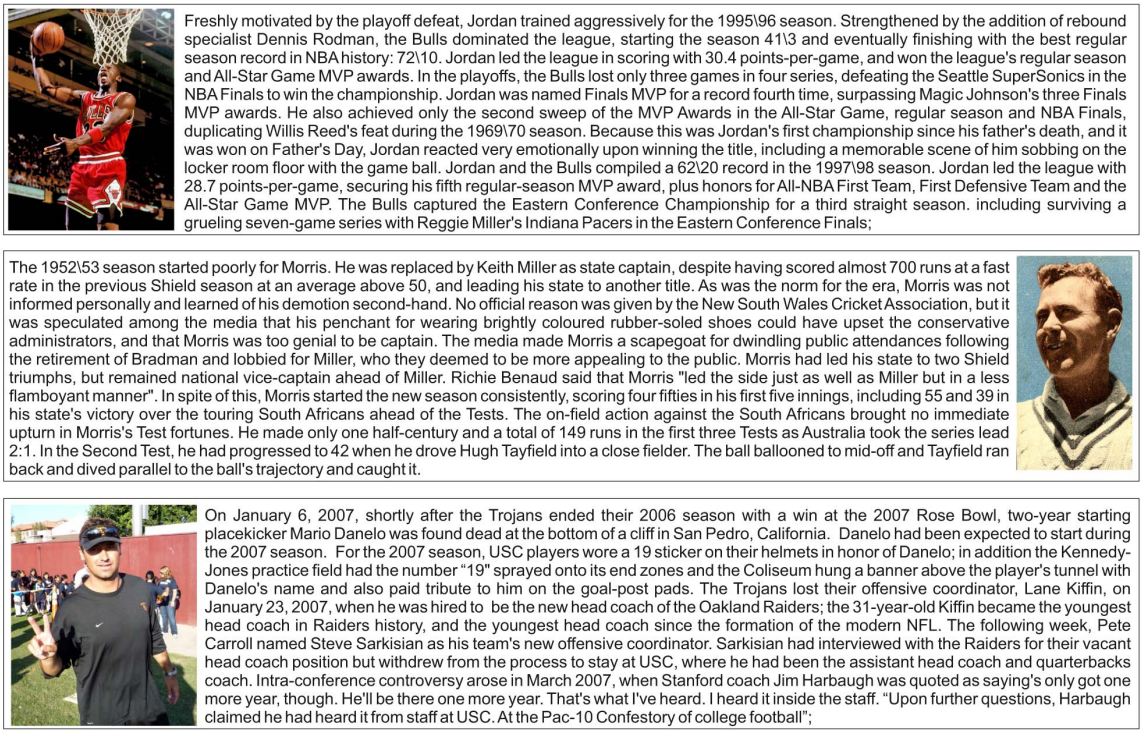


Fig. 3.12 Examples of image-text pairs from the “sport” category: (a) the NBA basketball player Michael Jordan and his career achievements (b) the cricket player Chris Morris and his achievements. (c) American football player Mario Danelo and stories of the USC Trojans team.

and the training data are matched, thus single-modality information retrieval systems are constructed upon low-level features, e.g., gradients for images and ‘pivot’ words for texts. However, some real-world scenarios require that the training and query data come from different modalities.

The study of neural networks can be traced back to 1969 in Marvin Minsky and Seymour Papert’s work [104]. Due to the significant increase of computational power in recent years, neural network-based deep learning has demonstrated its superiority over other machine learning algorithms, and has been widely applied in both academic researches and industrial projects. Autoencoder [58] is an unsupervised neural network-based learning method, and is typically used for dimension reduction. The previous work, correspondence autoencoder for cross-modality information retrieval [39], is close to our work, where a correspondence antoencoder is learned based on image-text pairs for extracting modality-invariant representations at the output layers of the autoencoder. On the other hand, our

proposed cross-modality autoencoder method does not require the expensive cross-modality correspondence information, and extracts features from the hidden layer.

In this work, we consider the supervised cross-modality retrieval problem. The neural networks are trained on the training data, where training labels are used to allocate identical random vectors at the output layers for data of the same class. Rasiwasia et al. [122] addressed the cross-modality retrieval problem by investigating the correlations between two modalities and the effectiveness of abstraction, where the canonical correlation analysis (CCA) and the use of abstraction are all proved to be effective. In order to validate the contributions of each separate component, three approaches correlation matching (CM), semantic matching (SM) and semantic correlations matching are proposed for the correlation modeling, the abstraction method and the joint working mode of both approaches respectively. Sharma et al. [137] proposed Generalized Multiview Analysis (GMA) to extract features from different views. GMA solves a joint, relaxed quadratic constrained quadratic program (QCQP) over different feature spaces to obtain a single linear/non-linear subspace, thus it affords an efficient eigenvalue based solution, and it is applicable to be extended to the cross-modality scenario. Sharma et al. [137] built the working environment for GMA with both Linear Discriminant Analysis (LDA) [4] and Marginal Fisher Analysis (MFA) [166], which result in approaches GMLDA and GMMFA respectively. A similar scenario to the supervised cross-modality problem was proposed in [184], where the 'same-distribution' annotated training data are utilized along with the annotated auxiliary domain training data. In order to train a learning system on two parts of data with different distributions, transforming original representations from different sources into a smooth feature space is necessary. Zhu and Shao [184] achieved such a transformation through a cross-domain dictionary learning, where the 'matched instances', which are associated through a fuzzy search procedure, are assumed to possess the same representations after being projected onto the learned dictionary pair. In the works of both [122] and [137], modality adaptation techniques aim at finding the linear combinations of the data from both the source modality and the target modality that possess maximum correlation with each other through CCA.

3.3.3 Cross-Modality Autoencoder

Problem Definition

Though the cross-modality information retrieval framework can be easily generalized to any mismatched pair of media forms, we restrict our discussions to the information retrieval

problem across text documents and image documents in this report. We consider the image training features as $Y_I = \{Y_I^1, Y_I^2, \cdots, Y_T^P\}\mathbb{R}^{d_I \times P}$, where d_I is the dimension of each original image feature (obtained by the SIFT+Bag-of-Words model) and P is the number of training images, and the text training features as $Y_T = \{Y_T^1, Y_T^2, \cdots, Y_T^P\}\mathbb{R}^{d_T \times P}$, where d_T is the dimension of each original text feature (obtained by the Latent Dirichlet Allocation model). Note that we consider the training numbers of data from both modalities are equal. We train two neural networks based on the training data of two modalities, where we allocate identical random vectors at the output layers for data of the same class. When data pass through the trained networks, features in the middle hidden layers are extracted as cross-modality representations. Let d be the feature dimension of the middle layers in both neural networks, then the two trained neural networks can in general be seen as two non-linear mappings, $F_I : \mathbb{R}^{d_I \times P} \rightarrow \mathbb{R}^{d \times P}$ and $F_T : \mathbb{R}^{d_T \times P} \rightarrow \mathbb{R}^{d \times P}$. When the testing data from both modalities are input to the image network F_I and the text network F_T respectively, pairwise distances are computed for the evaluation of retrieval performance.

Single-Modality Neural Networks

We begin by introducing a basic neuron of the neural network [108]. An example of the simplest neuron is shown in Fig. 3.13. We consider $\{x_1, x_2, \cdots, x_K\}$ as the input K-dimensional feature, $+1$ as an intercept term and $\{w_1, w_2, \cdots, w_K\}$ and b as neuron parameters. The output of such a neuron $h_{W,b}(x)$ can be computed as:

$$h_{W,b}(x) = f(\sum_{i=1}^{K} w_i x_i + b), \tag{3.21}$$

where the function f is chosen as the sigmoid function:

$$f(z) = \frac{1}{1 + exp(-z)}, \tag{3.22}$$

which scales the output $f(z)$ to in the range $[0, 1]$.

Many-to-One Autoencoder

An autoencoder neural network can be constructed by putting many simple neurons together. An example of an autoencoder neural network is shown in Fig. 3.13. The autoencoder neural network is normally used as an unsupervised learning method, and its target values at the

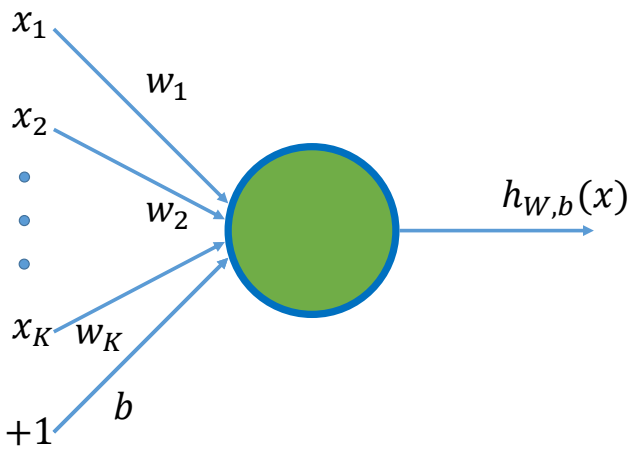


Fig. 3.13 An example of the simplest neuron.

output layer should set equal to the input values. Let $X = \{x^1, x^2, \cdots, x^P\} \in \mathbb{R}^{K \times P}$ be the K-dimensional input feature and $\hat{X} = \{\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^P\} \in \mathbb{R}^{K \times P}$ be the K-dimensional target values at the output layer, where P is the number of instances. The hidden layer values $Y = \{y^1, y^2, \cdots, y^P\} \in \mathbb{R}^{N \times P}$ are extracted as the N-dimensional feature of the input sample.

In this work, we design a supervised autoencoder learning algorithm, Many-to-One Autoencoder (MOAE). By setting identical target values at the output layer for training data that share the same class labels, the learned network can guarantee a low intraclass distance, so that encoded data become more discriminative. Experimental results suggest that these targets $\hat{X}$ can be set as random vectors. By enforcing the sparsity constraint to MOAE, the objective function can by formulated by the square-loss function with sparsity constraint on the weights:

$$\arg\max_{W,b} \frac{1}{P} \sum_{i=1}^{P} \|\hat{x}^i - h_{W,b}(x^i)\|_2^2 + \lambda \sum_{l=1}^{L} \|W^l\|_1, \tag{3.23}$$

where $W = \{W^1, W^2, \cdots, W^L\}\mathbb{R}^{K \times L}$ is the neuron parameters of MOAE and L is the number of layers, λ is the balancing parameter and W^l is the weight vector at layer l.

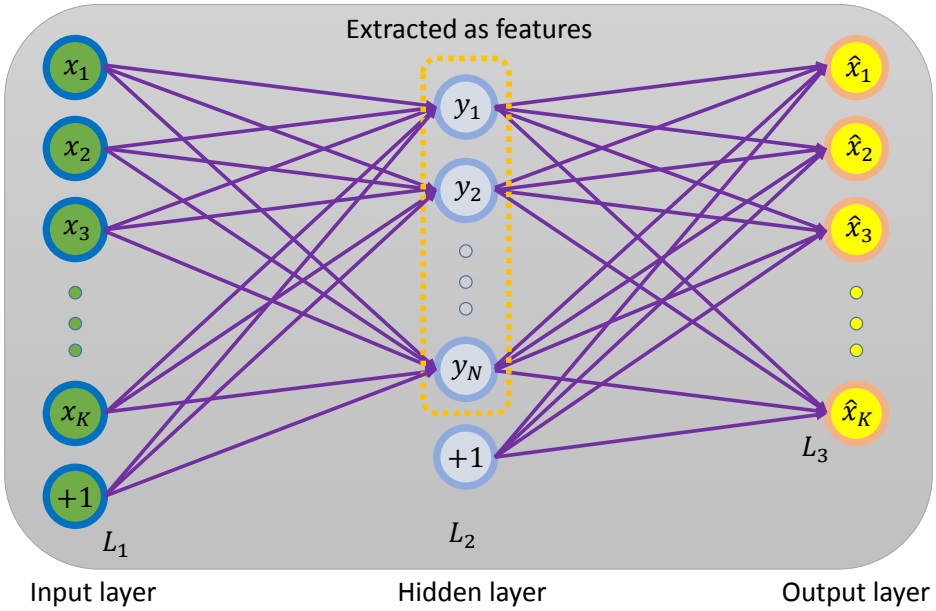


Fig. 3.14 An example of the Single-Modality Autoencoder.

Cross-Modality Autoencoder

In order to deal with data from two modalities, we learn a Cross-Modality Autoencoder (CMAE) by separately learning two MOAEs, while forcing identical target vectors at both output layers of the two MOAEs. The structure of CMAE is illustrated in Fig. 3.15. We further define $X_I = \{x_I^1, x_I^2, \cdots, x_I^P\} \in \mathbb{R}^{K_I \times P}$ as the K_I-dimensional image input features, and $X_T = \{x_T^1, x_T^2, \cdots, x_T^P\} \in \mathbb{R}^{K_T \times P}$ as the K_T-dimensional text input features. Correspondingly, $\hat{X}_I = \{\hat{x_I}^1, \hat{x_I}^2, \cdots, \hat{x_I}^P\} \in \mathbb{R}^{K \times P}$ and $\hat{X}_T = \{\hat{x_T}^1, \hat{x_T}^2, \cdots, \hat{x_T}^P\} \in \mathbb{R}^{K \times P}$ are the outputs of the two MOAE networks. Note that the number of neurons at the output layer (i.e., L_3) can be defined by the user, and we simply set an identical number K for both networks. As shown in Fig. 3.15, the two MOAEs can be driven towards learning a unified representations for both image and text data by linking both MOAE networks at the output layer, i.e., $\hat{X}_I = \hat{X}_T$. Thus, we can use the random vectors $\hat{X}$ to denote the outputs by setting $\hat{X} = \hat{X}_I = \hat{X}_T$. Based on 3.23, the objective function for learning the CMAE network can be formulated as:

$$\begin{aligned} &\arg\max_{W_I, b_I} \frac{1}{P} \sum_{i=1}^{P} \|\hat{x}^i - h_{W_I, b_I}(x_I^i)\|_2^2 + \lambda \sum_{l=1}^{L} \|W_I^l\|_1, \\ &\arg\max_{W_T, b_T} \frac{1}{P} \sum_{i=1}^{P} \|\hat{x}^i - h_{W_T, b_T}(x_T^i)\|_2^2 + \lambda \sum_{l=1}^{L} \|W_T^l\|_1, \end{aligned} \tag{3.24}$$

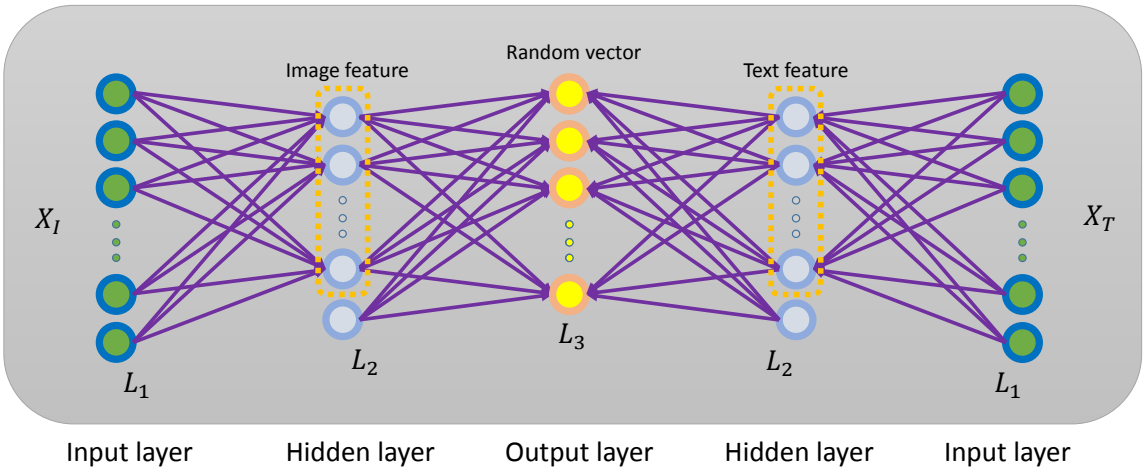


Fig. 3.15 An example of the Cross-Modality Autoencoder.

where, W_I, b_I, W_T and b_T are parameters of the image MOAE and the text MOAE respectively. The optimization of the CMAE neural network can be seen as separately optimizing two MOAE neural networks. The backpropagation algorithm [58], [5] is applied for optimizing above equations. Once we obtain the optimum $\hat{W}_I$, $\hat{b}_I$, $\hat{W}_T$ and $\hat{b}_T$, testing image and text data can be encoded by the CMAE. When inputing new features at two input layers of CMAE, neuron values in both L_2 layers are extracted as the image feature and text feature respectively.

3.3.4 Experiments

Dataset and Experimental Settings

Popular cross-modality datasets include the TVGraz dataset [72] and the Wikipedia dataset [122], however, the former is no longer maintained. We evaluate the propose CMAE approach on the Wikipedia dataset, which is a challenging image-text dataset with large intra-class variations and small inter-class discrepancies. The Wikipedia dataset consists of 2866 image-text pairs. The context of each text article describes people, places or some events, which are closely relevant to the content of the corresponding image document. There are 10 semantic categories in the Wikipedia dataset, including art & architecture, geography &

places, history, literature & theatre, biology, media, music, sports & recreation, royalty & nobility, warfare. We follow the data partition adopted in [122] to split the dataset into a training set of 2173 pairs and a testing set of 693 pairs. The text representation Y^T is derived from the latent Dirichlet allocation (LDA) model [8] (implemented with the Python Natural Language Toolkit[10]), which summarizes the semantic content or "gist" of a text document as a mixture of topics. The image representation is based on the scale invariant feature transformation (SIFT) [94] and the Bag-of-Words (BoW) representation using 128 codewords. We consider ground-truth labeling are provided in the training data, but the correspondence between each image-text pair as unavailable. For each MOAE neural network, 3 layers are used, which are the input layer L_1, the hidden layer L_2 and the output layer L_3. The numbers of neurons in the input layer is equal to the input image feature dimension or the text feature dimension. The number of neurons in the hidden layers and the output layers are set as 300 and 10, respectively.

Results on the Wikipedia Dataset

Table 3.13 Cross-Modality Retrieval Performance Comparison (MAP scores).

Methods	Image query	Text query	Average
PCA [69]	0.112	0.173	0.143
BLM [150]	0.202	0.256	0.229
CM [122]	0.193	0.245	0.219
SM [122]	0.218	0.226	0.222
GMMFA [137]	0.214	0.275	0.245
GMLDA [137]	0.210	0.275	0.243
LCFS [155]	0.214	0.279	0.247
SCM [122]	0.277	0.226	0.252
CMAE	0.270	0.226	0.248

Results on the Wikipedia Dataset

We compare the proposed CMAE approach with state-of-the-art approaches[11]. For the supervised cross-modality scenario, CMAE is compared with the non-knowledge transfer method (PCA), correlation matching (CM) [122], semantic matching (SM) [122], semantic

[10] www.nltk.org

[11] Note that the results reported in the extended version [20] are better than the results reported in the original version [122] using the same techniques. The improved performance is due to the merging of relevant classes, which reduces the total class number from 10 to 4. In this work, we follow the strategy in [122].

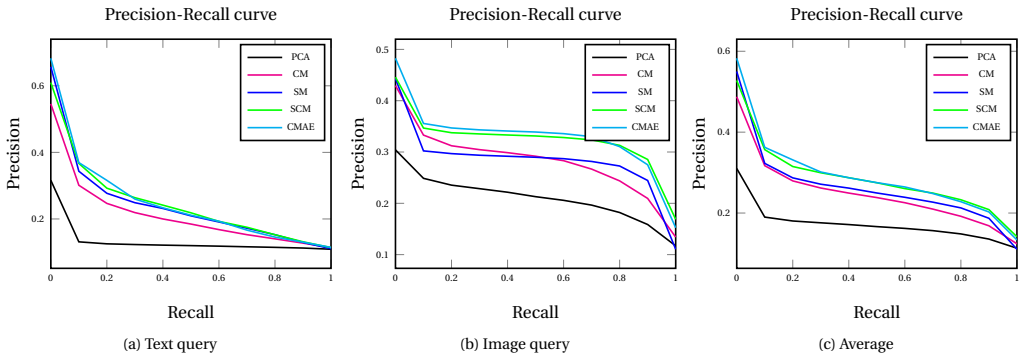


Fig. 3.16 Precision recall curves for different cross-modality retrieval methods.

correlation matching (SCM) [20], a bilinear model (BLM) [150], generalized multiview linear discriminant analysis (GMLDA) [137] and generalized multiview marginal fisher analysis (GMMFA) [137]. The non-knowledge transfer approach is achieved by directly applying PCA to the image modality in order to unify the feature dimension across both modalities. CMAE is compared with classical methods based on a single-modality (UNI) and an exhaustive search of mixing data from both media forms for training. The abrupt approach is realized through enforcing dimensionality reduction (PCA) on the image representations. We compare the performance of different approaches through 11-point interpolated precision-recall (PR) curves [100]. The PR curves for different cross-modalities methods are given in Fig. 3.16. MAP scores, which are calculated based on the under curve area of PR curves, are given in Table 3.13. The proposed CMAE can achieve state-of-the-art performance for either image queries or text queries. Also, from the results, we can conclude that abruptly introducing miss-matched data to a target domain can break the data smoothness in the original domain, and thus lead to weak performance.

3.3.5 Conclusion

In this work, we have proposed a supervised many-to-one autoencoder neural network approach, and extend this approach to address the cross-modality image/text retrieval problem. Specifically, two many-to-one autoencoders are trained for both image and text data. When data pass through the trained networks, intra-class distances can be guaranteed at a low level in the new feature spaces. By forcing the output layers of these two many-to-one

autoencoders to be equal to each other, cross-modality data can be associated, so that missmatched image and text data can be mapped to an isomorphic feature space when encoded by the cross-modality autoencoder. The proposed method is evaluated on the Wikipedia dataset, and experimental results suggest that the cross-modality autoencoder can achieve state-of-the-arts performance.

3.4 Cross-View Action Recognition[12]

3.4.1 Motivation and Introduction

In the past few years, along with the explosion of online image and video data, computer vision based applications in image/video retrieval, human-computer interaction, sports events analysis, .etc are receiving significant attention. Also, as can be anticipated, future products, such as the Google Glasses, which can essentially revolutionize traditional human-computer interaction , will bring more requirements and challenges to computer vision algorithms. As an important topic in computer vision, human action recognition plays a key role in a wide range of applications. Many approaches [143], [185], [172], [50], [157], [70], [180], [138] are proposed, however, some challenges still remain in real-world scenarios due to cluttered background, view point changes, occlusion and geometric variations of the target.

Recently, novel strategies have been proposed to represent human actions more discriminatively. These representations include optical flow patterns [32],[3], 2D shape matching [95], [164], [85], spatio-temporal interest points [27], [89], trajectory-based representation [121], .etc. Many state-of-the-art action recognition systems [130], [153], [84] are based on the bag-of-features (BoF) model, which represents an action video as a histogram of its local features. When cooperating with informative low-level features on detected spatio-temporal interest points or densely sampled 3D blocks, the BoF model and its variants yield encouraging performance in many challenging and realistic scenarios [153]. However, such results are achieved under a fixed viewpoint or within limited view point variations, i.e., the discriminative capability of such representations tends to significantly degrade when the view point variations are increased. Thus, we aim to seek a high-level feature representation that brings action videos captured from different view points to the same feature space, while keeping its discriminative power and allowing the data to satisfy the smoothness assump-

[12]The content of this section is published at:
F. Zhu and L. Shao, Correspondence-Free Dictionary Learning for Cross-View Action Recognition, International Conference on Pattern Recognition, Stockholm, Sweden, Aug. 2014.

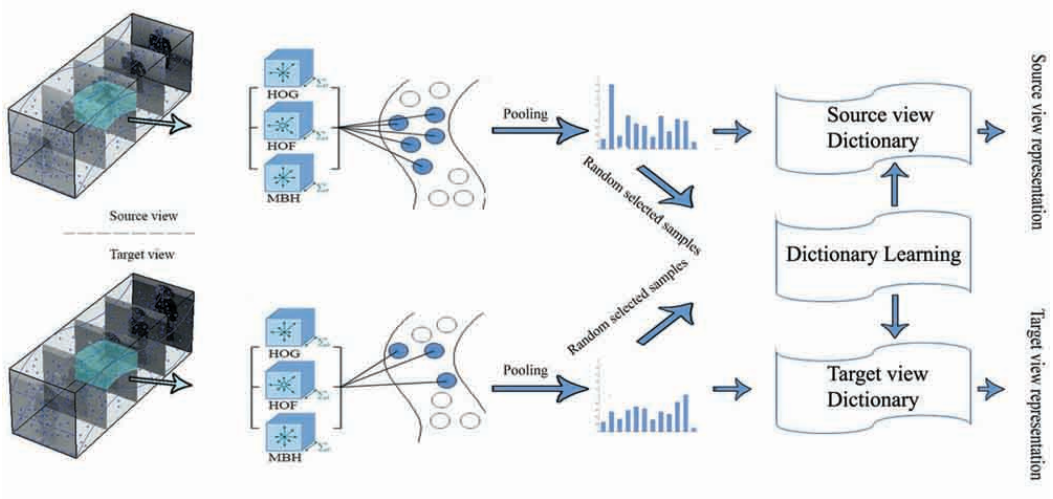


Fig. 3.17 The flowchart of our framework. Low-level dense trajectories are first coded with LLC to derive a set of coding descriptors. By pooling the peak values of each dimension of all local coding descriptors, a histogram that captures the local structure of each action is obtained. Dictionary learning is conducted utilizing randomly selected actions from both views, then source view training actions and target view testing actions are coded with the learned dictionary pair to obtain the cross-view sparse representations.

tion (which implies that data points which are close to each other are more likely to share the same label.) in supervised learning. Many recent efforts have been paid towards this direction. One typical line of attack is to infer the three-dimensional scene structure based on the given cross-view actions, where the derived features can be adapted from one view to another utilizing geometric reasoning [49]. Junejo *et al.* [70] applied a temporal Self-Similarity Matrix to store distances between different pairs of actions for a view-invariant representation. In [88], a bipartite graph is built via unsupervised co-clustering to measure visual-word to visual word relationship across different views so that a high-level semantic feature that bridges the semantic gap between the two Bag-of-Visual-Words (BoVW) vocabularies can be generated. Li *et al.* [82] adopted the conceptual idea of 'virtual views' to represent an action sequence continuously from one observer's viewpoint to another, and similarly, Zhang *et al.* [180] utilized a kernel-based method to capture all the virtual views on the virtual path instead of sampled views to keep all the visual information on the virtual path and eliminate the parameter tuning procedure. Zheng *et al.* [62] adopted the K-SVD algorithm [1] to construct an over-complete transferable dictionary pair, which encourages actions taken from different viewpoints to possess the same representation. These methods require either labeled samples in the target view or correspondence annotations, which, however, are expensive or impossible to obtain in many scenarios. Our approach is most

similar to [62], however, there is one significant difference in terms of the training data requirement between our approach and the one adopted in [62] that we learn the cross-view action representation in an unsupervised manner and action correspondences across the source view and the target view are not required in our learning phase. Such elimination of the strict training data requirement is very useful and can be seen as a significant progress in cross-view action recognition since neither the labeled training data in the target view nor the correspondences across the source view and the target view are handy to obtain in most real-world applications.

As an attempt towards real-world applications, our approach addresses the cross-view action recognition problem utilizing only labeled source view actions and unlabeled target view actions. In order to capture the local structure of actions from each view individually, the dense trajectories features [153] are first coded by the Locality-constrained Linear Coding (LLC) [68] layer. The view knowledge transfer is performed by an efficient dictionary learning method [96], which brings the query action in the target view into the same feature space of actions in the source view. The construction flowchart of the cross-view sparse representation is shown in Fig. 3.17 This work makes the following contributions:

⋆ By captureing both local action structures and the cross-view knowledge, the proposed representation guarantees its discriminative capability over different action categories as well as different observation viewpoints.

⋆ In accordance with the initial intention of transfer learning, the proposed approach is unsupervised, and only requires action labels from the source view.

⋆ We give an in-depth review of dictionary learning and coding methods under different constraints.

⋆ The experimental results are promising, and they lead to a new setting towards real-world applications.

3.4.2 Cross-View Dictionary Learning

Problem Statement

We define $Y_s = [y_s^1, \cdots, y_s^n] \in \mathbb{R}^{d \times n}$ as n d-dimensional features extracted from source view actions and $Y_t = [y_t^1, y_t^2, ..., y_t^m] \in \mathbb{R}^{d \times m}$ as m d-dimensional features extracted from target view actions, where Y_t are unlabeled and Y_s are labeled. Based on the fact that action videos from two different viewpoints must contain the same action, we assume there exists a high-level action representation shared between the two videos captured from different view-

points. Given any Y_t, we can always find corresponding actions in Y_s. The aim is to discover the connections between Y_t and Y_s, and find a projection that can map Y_t and Y_s into a unified feature space based on such connections.

Locality-Constrained Linear Coding

In aforementioned image classification tasks of this report, local image patches are always encoded with sparse coding. In this work, local action descriptors are encoded with LLC instead of sparse coding. Comparisons between sparse coding (SC) and vector quantization (VQ) are given in Section 1.3, and SC has demonstrated its effectiveness and superiority over VQ in many tasks [168], [184]. Here, we quote the discussion in [68] regarding the advantage of LLC over SC. While keeping the sparsity, LLC also favors each of the set of dictionary atoms that a signal associates to be close to the input signal respectively. Given the low-level action representation $V = [v_1, v_2, ..., v_N] \in \mathbb{R}^{d_1 \times (N)}$ and the codebook B with M atoms, the objective function of LLC is given as:

$$\mathscr{F}_{\text{LLC}}(y_i) = \sum_{i=1}^{N} \|v_i - By_i\|^2 + \lambda \|q_i \odot y_i\|^2, \quad \text{s.t.} \quad 1^\top y_i = 1, \forall i \tag{3.25}$$

where the notation $\odot$ denotes element-wise multiplication, and

$$q_i = \exp\left(\frac{\text{dist}(v_i, B) - \max(\text{dist}(v_i, B))}{\sigma}\right) \tag{3.26}$$

measures the normalised distances between each input signal v_i to all dictionary atoms $B = [b_1, b_2, \cdots, b_M]$ with $\text{dist}(v_i, B) = [\text{dist}(v_i, b_1), \cdots, \text{dist}(v_i, b_M)]^T$ and $\text{dist}(v_i, b_j)$ is the Euclidean distance between the input signal v_i and dictionary atom b_j. σ is the parameter that controls the weight decay speed for each v_i. After being encoded according to Equation 3.25, the set of local descriptors that locate within the same region are unified to a single vector using max pooling. Specifically, Z local descriptors within the same region are put into a single $Z \times M$ matrix, followed by which only the maximum value at each of the M bins is kept as the value for the global feature at such a bin. Fig. 3.18 shows the overall structure of how the global feature vector for a action sequence is generated.

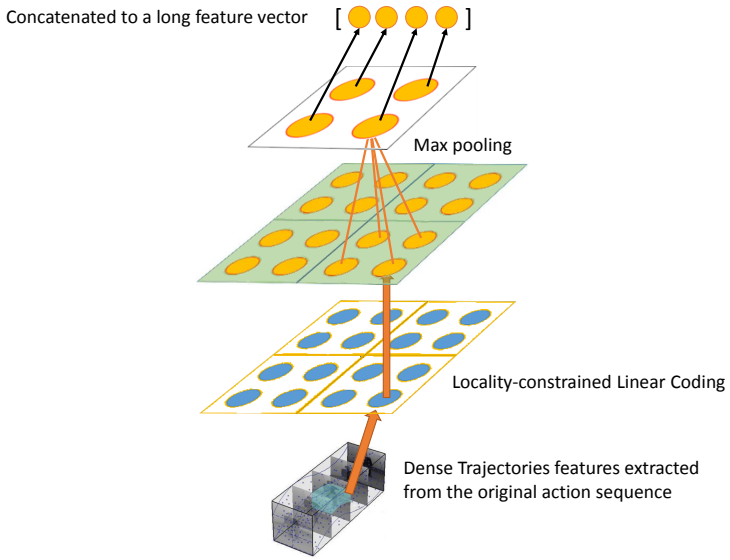


Fig. 3.18 Illustration of how the global feature vector for a action sequence is generated.

Dictionary Learning

In order to eliminate or minimize the data discrepancy between cross-view action sequences that describe the same action category, we simultaneously learn a dictionary pair, based on which actions from different views can be projected into a unified feature space and regular learning methods can be performed on the projected data. Let X_t and X_s be the target domain and source domain projection codes respective, we formulate the objective function for learning such a dictionary pair as:

$$\begin{aligned} F(D_s, D_t) \triangleq & \min_{X_s \in \mathbb{R}, X_t \in \mathbb{R}} \frac{1}{2} \|Y_s \mathbb{A}^T - D_s X_s \mathbb{A}^T\|_2^2 \\ & + \frac{1}{2} \|Y_t - D_t X_t\|_2^2 + \|X_t - X_s \mathbb{A}^T\| \\ & s.t., [\|x_S^i\|, \|x_T^j\|]_0 \leq T, \end{aligned} \tag{3.27}$$

where $\mathbb{A}$ is a transformation matrix that formulates the source domain data in correspondence with the target domain data. Specifically, for each column j, $\mathbb{A}(i, j)$ can be computed as:

$$\mathbb{A}(i^*, j) = \begin{cases} 1, & if \quad i^* = \arg\max_{i=1:n} (G(i, j)) \\ \\ 0, & otherwise, \end{cases} \tag{3.28}$$

where $G_c(i,j)$ is the Euclidean distance between the ith source domain instance and the jth target domain instance. Note that the computation of $\mathbb{A}$ is similar to Equation. 3.3, while the only difference is that since the target domain label information are not available in this work, the computation of $\mathbb{A}$ is not conducted within each category. By applying the constraint that each pair of correspondence cross-view actions share identical representation after the projection to Equation. 3.27, i.e., $X_t = X_s\mathbb{A}^T$, the new learning function can be rewritten as:

$$\begin{aligned} F(D_s,D_t) \triangleq & \min_{X_s\in\mathbb{R},X_t\in\mathbb{R}} \frac{1}{2}\|Y_s\mathbb{A}^T - D_sX_s\mathbb{A}^T\|_2^2 + \frac{1}{2}\|Y_T - D_tX_t\|_2^2 \\ & s.t., [\|x_S^i\|, \|x_T^j\|]_0 \leq T, \end{aligned} \tag{3.29}$$

The optimization and convergence analysis of Equation. 3.29 is the same as Section. 3.1.6 and Section. 3.1.6 respectively. Once the cross-view dictionary pair D_s and D_t are obtained, view-invariant cross-view action representations can be computed by projecting original action LLC features to the dictionary pair. Finally, classification can be achieved by applying a linear SVM classifier to the projected data.

3.4.3 Experiments and Results

Table 3.14 Performance comparison of action recognition with and without knowledge transfer.

%	Camera 0		Camera 1		Camera 2		Camera 3		Camera 4	
	woTran	wTran	woTran	wTran	woTran	wTran	woTran	wTran	woTran	wTran
Camera 0	-	-	23.03	92.42	23.94	89.09	26.67	91.52	30.61	90.00
Camera 1	25.76	92.42	-	-	35.15	90.61	33.33	92.42	30.30	90.30
Camera 2	16.06	92.73	7.27	92.42	-	-	29.39	92.12	34.55	90.91
Camera 3	12.42	94.24	9.39	93.33	26.36	90.91	-	-	30.91	90.30
Camera 4	12.42	93.94	10.91	93.03	10.3	92.12	17.27	95.15	-	-

We evaluate our approach on the IXMAS multi-view action dataset [158] (exemplar actions are shown in Fig. 3.19, which contains eleven action categories, e.g., walk, kick, wave, etc. Each action is performed three times by ten actors taken from five different views. We follow the leave-one-action-class-out scheme [82] to separate the data, where we consider one action class (called an 'orphan action') in the target view, and exclude all the videos of that class when learning the dictionary pair. Samples used for dictionary learning are randomly selected from the non-orphan actions. In accordance with previous work [82],

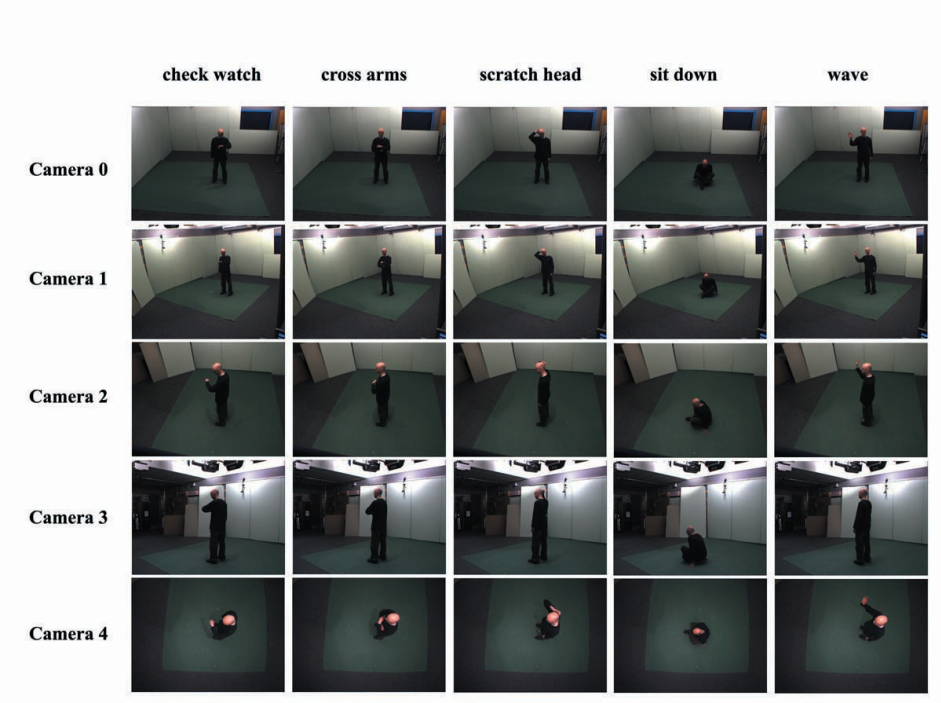


Fig. 3.19 Exemplar frames from the IXMAS multi-view action recognition dataset. The columns show 5 action categories, including *check watch*, *cross arms*, *scratch head*, *sit down*, *wave*, and the rows show all the 5 camera views for each action category.

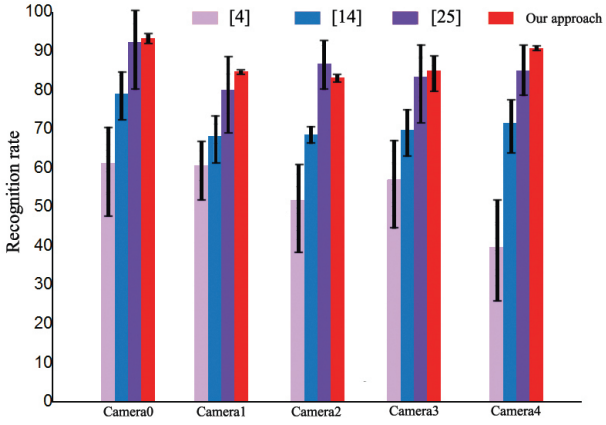


Fig. 3.20 Performance comparison with state-of-the-art methods.

30% of the non-orphan actions are chosen from each view separately. The experiments are conducted on any possible combinations of view pairs, i.e., twenty combinations in total are considered.

We use the dense trajectories [153] as the primary feature to represent raw action video sequences. Dense trajectories are extracted with 8 spatial scales spaced by a factor of $1/\sqrt{2}$, and feature points are sampled on a grid spaced by 5 pixels and tracked in each scale, separately. Each point at frame t is tracked to the next frame $t+1$ by median filtering in a dense optical flow field. To avoid the drifting problem, the length of trajectory is limited to 15 frames. Additionally, HOGHOF [60] and Motion Boundary Histogram (MBH) [105] are computed within a $32 \times 32 \times 15$ volume along the dense trajectories, where each volume is sub-divided into a spatio-temporal grid of size $2 \times 2 \times 3$ to impose more structural information in the representation.

The experimental results are shown in Table 3.14, where rows correspond to the source views and columns correspond to the target views. For each view pair, we show both results for 'woTran' and 'wTran' settings, which denote action recognition with and without knowledge transfer respectively. In both settings, low-level dense-trajectories are first coded with LLC in each individual view. The 'woTran' setting is treated as a normal supervised classification task in the same feature domain, so that the LLC codes are directly fed into classification. On the other hand, in the 'wTran' setting, the LLC codes are further decomposed to sparse linear combinations of basis dictionary elements, which are learned utilizing samples from both views. We construct a codebook with $1,000$ visual words for the LLC codes, and the learned dictionary pair for cross-view knowledge transfer is set to be size

90. The average accuracies are 22.52% and 92.00% for 'woTran' and 'wTran' respectively. In all the cases, the proposed method outperforms those that directly use classifiers trained on the source view to predict action labels in the target view, where the most significant improvement is 85.15%. We also compare the average performance for each camera view of the proposed approach with state-of-the-arts methods in Fig. 3.20. Clearly, our approach significantly outperforms others even though with a stricter setting.

3.4.4 Conclusion

In this work, we have presented an unsupervised dictionary learning method to address the cross-view action recognition problem. By setting up virtual connections across the source and target view samples, dictionary learning is performed on these samples. Being coded by the learned dictionary pair, the discriminative power of action representations from different views can be guaranteed in the new feature space, so that the cross-view action recognition problem can be solved as a traditional supervised learning problem. The proposed approach achieves state-of-the-arts results on the IXMAS action dataset using only labeled source view samples, and even outperforms some methods which utilize correspondence annotations of action samples across different views. This work leads to a novel cross-view action recognition setting towards real-world applications with little information provided.

Chapter 4

Multi-View Camera Fusion[1]

4.1 Motivation and Overview

Many previous human action recognition works have considered challenging problems, such as illumination or background variations, occlusions and viewpoint changes [144]. Among them, data with viewpoint changes are very common and basically inevitable in real-world applications due to human or camera movements. The apparent deficiency of single-camera systems prompts the advancement of recent approaches using multiple-cameras to deal with such viewpoint change problems. Algorithms based on multi-view cameras have recently received considerable attentions. Many approaches have been proposed and tested on the multi-view IXMAS dataset. Fig 2.1 illustrates examples of actions and their associated silhouettes from this dataset. In [135], Shao et al. adopted body pose silhouettes as feature descriptors to build the Correlogram of Body Poses (CBP) global representation for each video sequence beyond its baseline Histogram of Body Poses (HBP) [133] representation, and achieved satisfying results on the IXMAS dataset. In [71], Junejo et al. explored the self-similarities of action sequences overtime as a measurement to overcome view-changes. However, recognition in all these methods is done on individual cameras and the fusion of different camera views is neglected.

The idea of decision trees and its extension, decision forests, have been previously studied for both action localization and recognition, e.g., in [123], [174], [85], [106]. In [85],

[1]The content of this chapter is published at:
F. Zhu, L. Shao and M. Lin, Multi-View Action Recognition Using Local Similarity Random Forests and Sensor Fusion, Pattern Recognition Letters, vol. 34, no. 1, pp. 20-24, Jan. 2013.

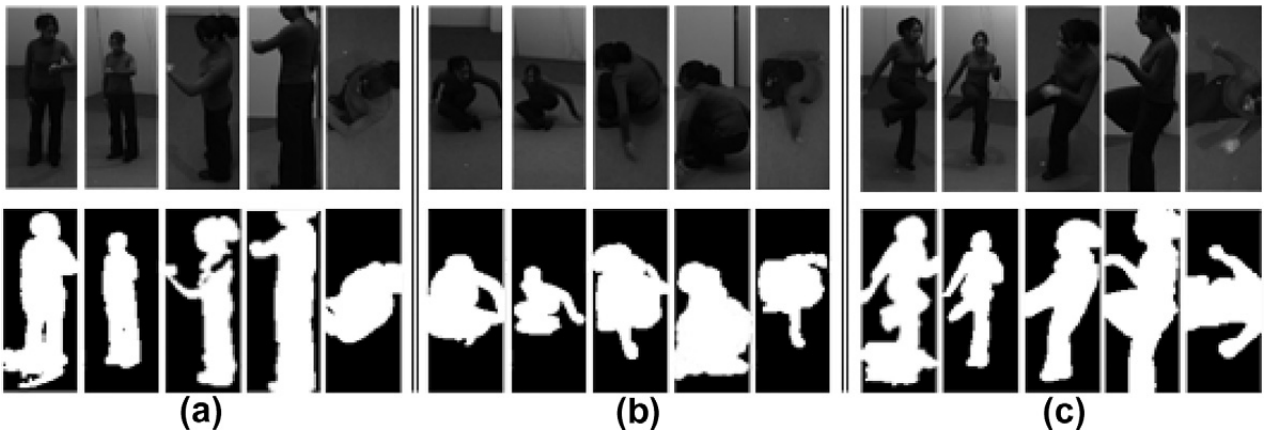


Fig. 4.1 Body poses from (a) check watch, (b) sit down, and (c) kick. Each action is performed by the same person Amel and captured from cameras 0–4.

an action prototype tree is learned in both shape and motion spaces using the hierarchical k-means clustering. Then they take a prototype-prototype distance from the codebook as a measurement, upon which they calculate the joint likelihood with both action location and prototype. In [103], in contrast to most previous recognition works that utilize small and flat codebooks, they apply a large number of features represented in many vocabulary trees instead. In addition to action recognition, their approach also accomplishes action localization simultaneously. An image-feature vocabulary using a novel quantization method with randomized trees as an alternative of kmeans clustering was proposed in [116]. The advantage of choosing randomized trees for vocabulary generation has also been demonstrated in [85] that the randomized trees can dramatically outperform k-means in terms of both efficiency and accuracy, especially when dealing with large scale data. In contrast to [135], [133], [85], in which the random forests is employed as a vocabulary generating or indexing tool, [171] applied the randomized trees to learn the 3D local video patches to acquire their corresponding votes in the 4D Hough-transformed space. The same as in [133], both action label and location are obtained from their voting framework.

Some existing techniques explored the fusion results from multi-cameras on the IXMAS dataset. Among them, a very common strategy for camera fusion is to concatenate the feature descriptors from different cameras, e.g., [163], Such a fusion method would benefit the recognition accuracy by describing the actions with more features. However, it will consequently lead to two new problems. Firstly, concatenating the feature descriptors will result in a much longer feature descriptor, which can be five times long for five cameras. Therefore, this will naturally increase the computational complexity for clustering and dimension

reduction. The second shortcoming, which is more critical, is the likelihood that the fused recognition accuracy would deteriorate due to the unequal performance of each camera with respect to different actions. Correspondingly, the first problem can be solved with the randomized forests classifier and the second can be tackled by our new voting strategy.

In this report, we propose a simple approach that employs local segments of binary silhouettes on the random forests classifier, and then apply a novel voting strategy to label the testing actions. The random forests was introduced in [11], and it has the advantages over other learning algorithms in efficiency and effectiveness, and it can avoid the overfitting problem by setting more decision trees. Although the action representation is simple and not robust against viewpoint changes, we can still get impressive results due to the effectiveness of the random forests as a classifier and the voting strategy. We also extend our approach from single camera view to multi-camera fusion and evaluate the performance of different camera fusion scenarios on the IXMAS dataset.

4.2 Local Segment Representation and Randomized Tree Training

4.2.1 Segment of 2D Silhouettes

Silhouette extraction is a popular technique for action recognition. With the silhouette data, intra-class variations, such as background changes and clothing, that may affect the recognition performance are easily overcome. Obviously, the quality of the silhouettes is closely related to the recognition performance. Since silhouette extraction is not the focus of this report and the 2D silhouette per frame for each action sequence is provided by the IXMAS dataset, we simply use those silhouettes to represent body poses without discussing how the silhouettes are extracted. A bounding box is placed around each silhouette and normalized to the size of 20×30, which is then converted into a 600 dimensional descriptor containing only binary values. As shown in Fig. 4.2, the two sequences at the right side of the graph illustrate two sets of 2D binary bounding boxes of camera 0 and camera 4.

In order to consider the temporal order of poses, we use temporally densely sampled segments, which have overlaps with neighboring segments. Each segment is set to be the size of $20 \times 30 \times T$, where T refers to the segment's length in frame number. With this setting, each segment has the full spatial size of the input silhouette sequence and only

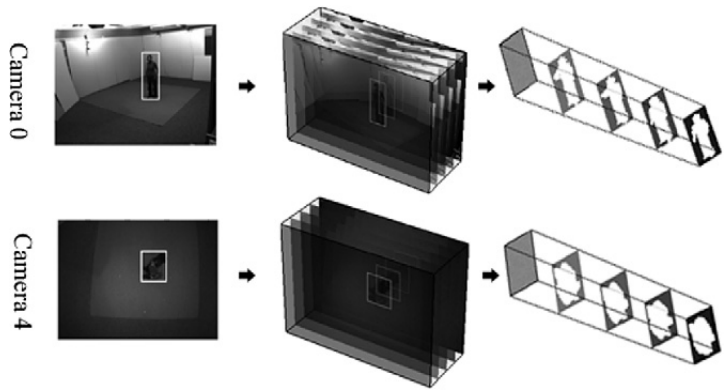


Fig. 4.2 Silhouettes extracted from different camera views.

varies temporally. A segment is then represented by further concatenating each row of its 2D silhouettes, which results in a $600 \times T$ dimensional segment descriptor. It is claimed in [14] that the very densely sampled segments can help boost the recognition performance, thus we place the segments as densely as possible in the temporal axis. Specifically, the overlap between consecutive segments is $T \times 1$ frame, i.e., the step for the sliding segment is one frame. Assuming the total frame number of a silhouettes sequence is N, $N - T + 1$ segments can be generated from a video sequence.

4.2.2 Randomized tree training

The training process is constructed according to the standard random forests structure in [9]. Local segments from the training sets are trained with the random forests classifier, which is assembled by a set of randomized decision trees. In each decision tree, M segment features are randomly selected from the training sets and placed at a root node, which is mapped to a set of termination leaf nodes through the interior binary splitting joints.

Let $V^m = \left\{v_1^m, v_2^m, \cdots, v_{600\times T}^m\right\}$ be the $600 \times T$-dimensional feature vector of segment $m \in M$, K be the total number of leaf nodes generated within a decision tree, t_k be the threshold at leaf node $k \in K$ and $f_k(V)$ be the splitting function that takes all the segments at note k as inputs, then the splitting decision at note k can be defined as:

$$\text{splitting decision} = \begin{cases} \text{right}, & \text{if} \quad f_k(v_i^m) \geq t_k, \\ \text{left}, & \textit{otherwise}. \end{cases} \tag{4.1}$$

Fig. 4.3 illustrates of the structure a decision tree. The posterior probability p_c^m that segment m belongs to class c can be computed by the proportion of segments of each action class at

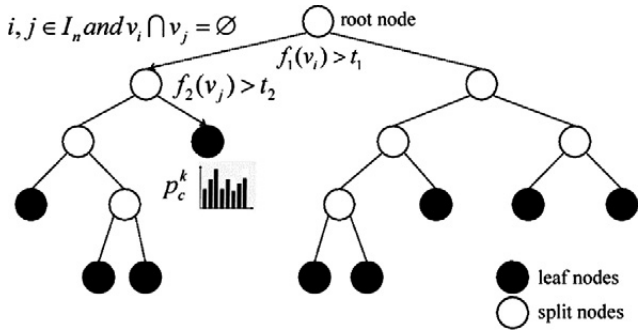


Fig. 4.3 The structure of a decision tree.

the leaf node k. The quality of split at each leaf node can be measured by the information gain:

$$\begin{aligned} \Delta E &= E(I_1^k) + E(I_r^k), \\ E(I_1^k) &= -\frac{N_l^k}{N_l^k + N_r^k} \log(\frac{N_l^k}{N_l^k + N_r^k}), \\ E(I_r^k) &= -\frac{N_r^k}{N_l^k + N_r^k} \log(\frac{N_r^k}{N_l^k + N_r^k}), \end{aligned} \tag{4.2}$$

where I_l^k and I_r^k denote the left splitting features and the right splitting features at leaf node k, and correspondingly, N_l^k and N_n^k denote the number of the left splitting features and the right splitting features respectively. In the training phase, the training set is equally partitioned into a number of subsets, which are then fed to different decision trees. In order to boost the general performance, the subsets are set to have overlaps with each other.

4.2.3 Random forest classification

We assume there are N_Q decision trees in the random forest, then in the testing phase, each segment within the testing video is fed to each of the N_Q decision trees, and finally terminates at a leaf node of a decision tree.

The overall prediction P^m that describes the probabilities of the segment m belonging to each of the C classes can be computed by summing over all the leaf node histograms $P_n^m = \left[p_1^m, p_2^m, \cdots, p_C^m\right]$:

$$P^m = \sum_{n=1}^{N_Q} P_n^m. \tag{4.3}$$

The discriminative capacity varies over different segments. For example, segments that

contain key frames of an action can be most discriminative, while segments that mainly contain transition frames (i.e., frames between two actions) are meaningless. In order to take the respective contribution of each segment into account, we assign a weight to the prediction of each segment when deciding the action class of the input video sequence. Thus, the weight assigned to segment m at a leaf is defined as:

$$w_m = \frac{N_m^{c^*}}{N_m}, \tag{4.4}$$

where c^* denotes the action class that possesses the maximum number of segments, N_m denotes the total number of segments and $N_m^{c^*}$ denotes the corresponding segment number of class $c = c^*$ in the training. To be consistent with the definition in section 4.2.1, we assume the number of segments within a query video sequence is $N - T + 1$. Consequently, the video-level prediction histogram can be computed as:

$$\begin{aligned} P &= \sum_{m=1}^{N-T+1} w_m P_m \\ &= \sum_{m=1}^{N-T+1} \frac{N_m^{c^*}}{N_m} \sum_{n=1}^{N_Q} P_n^m. \end{aligned} \tag{4.5}$$

Each bin P^c within P denotes the posterior probability that the input action belongs to class $c = \{1, 2, \cdots, C\}$, and the predicted action class can be found at the bin with the largest value:

$$l^* = \arg\max_c P^c. \tag{4.6}$$

4.2.4 Multi-Camera Voting Strategy

The multi-camera fusion strategy is designed by further assigning a weight onto the prediction histogram of each camera view. Similar to the local voting strategy, the weight is computed by the proportion of the segments that have the same maximum voting class label in their prediction histograms to the total number of segments within the video sequence, where the segments with this class label are more than those with other class labels. Such a weighting strategy is based on the fact that cameras from different observation views would have fluctuating performance against different specific actions, as shown in Fig. 4.4. The

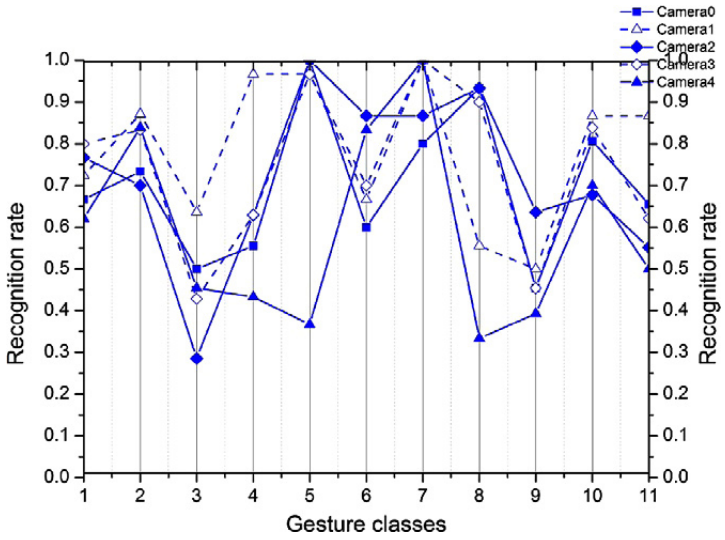


Fig. 4.4 Individual camera performance of all the five cameras.

weight assigned to the prediction histogram of each view can be described as:

$$w_v = \frac{N_v^{c^*}}{N - T + 1}, \quad v = 0, 1, 2, 3, 4, \tag{4.7}$$

where v denotes each camera view and $N_v^{c^*}$ denotes the maximum number of segments that are predicted as the action class $c = c^*$. Since each camera has the same number of segments for the same testing video sequence and the same number of decision trees to classify each local segment, normalization is not required. Thus, the multi-camera fusion prediction histogram can be obtained by accumulating each camera's weighted prediction histogram:

$$P_{\text{multi-view}} = \sum_{v=\{0,1,2,3,4\}} w_v P_v. \tag{4.8}$$

Finally, the multi-view camera-based prediction can be made by finding the largest bin of $P_{\text{multi-view}}$:

$$l^*_{\text{multi-view}} = \arg\max_c P^c_{\text{multi-view}} \tag{4.9}$$

4.3 Evaluations

For evaluation, we choose the leave-one-out cross-validation method, i.e., in each iteration action sequences performed by one out of ten subjects are selected as the testing set and the remaining sequences as training set, and the final recognition rate is the average of the ten iterations. To optimize the performance, we vary the segment length t from 8 to 18, and the best result is achieved when $T = 18$. In the case of $T = 18$, a segment feature has the dimension $d = 20 \times 30 \times 18 = 10800$. Then we reduce the dimensionality of such a binary representation using PCA and set the reduced dimension to be $k = 30$. For the random forests classifier, we set the number of decision trees to be 600 and the number of predictors sampled at each splitting node to be equal to the square of the feature dimension.

To demonstrate the effectiveness of our method, we first compare the results of our method with those of the baseline BoW [110] and the NBNN [9] methods, which both employ the same segment representation as in the proposed algorithm for fair comparison. Fig. 4.5 depicts recognition results of these three techniques when individual camera views are used. For most views, the results of the BoW method and the proposed method are analogous, while both significantly outperform the NBNN method. Table 4.1 shows results on different combinations of camera views with different methods. The highest classification accuracy we achieve is 88%, which outperforms most of the methods in comparison. The analogous performance between the proposed method and the BoW method of each single view (shown in Fig. 4.5) also proves the effectiveness of the proposed camera fusion strategy as it outperforms the BoW concatenation fusion method by almost 10%. As shown in Table 4.1, the AFMKL method achieves the best results. Note that the learning process of the AFMKL method is much more complicated than our method (where we only take the concatenations of binary silhouettes in different frame segments as inputs of the random forests classifier) that the performance of the AFMKL method for each single camera view is consequently much better, i.e., 5% better in average for Cameras0-2. However, for the fusion performance comparison, the result of our method is only 0.2% lower than the AFMKL method, which, therefore, proves the effectiveness of our fusion strategy.

Table 4.2 shows the comparison between the early concatenation fusion method and the fusion strategy we propose. Our approach outperforms the early concatenation fusion method in most scenarios. For Cameras 0&2, the reason that the early concatenation fusion method is slightly better than our method might be that the individual performance of Camera 0 and Camera 1 over different action classes has similar distribution, which consequently leads to mediocre performance of our fusion strategy. The confusion matrix for 11

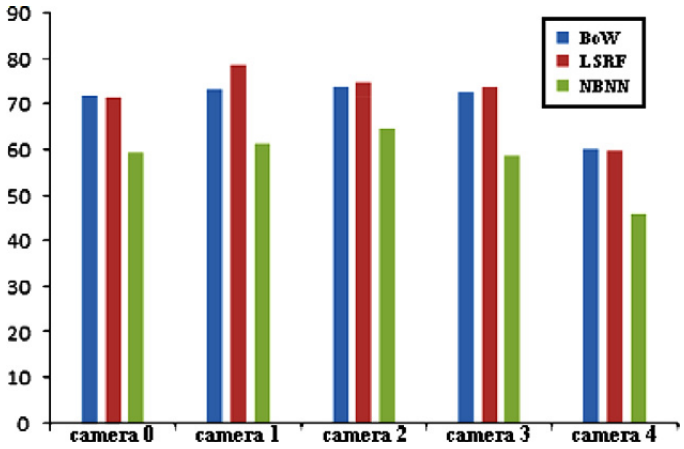


Fig. 4.5 Comparison between our method and the BoW and the NBNN methods on each camera view.

Table 4.1 Classification accuracies of different methods for both single and multiple camera views on the IXMAS dataset.

Method	Camera 0	Camera 1	Camera 2	Cameras 0 and 2	Cameras 0–2	Cameras 0–4
Ours	71.5%	78.7%	73.9%	85.7%	86.6%	**88.0**%
BoW [110]	71.6%	72.3%	72.7%	74.2%	79.1%	78.7%
NBNN [9]	59.5%	61.3%	64.8%	62.4%	63.1%	61.1%
AFMKL [163]	81.9%	80.1%	77.1%	86.6%	87.7%	**88.2**%
GMKL [152]	76.4%	74.5%	73.6%	76.2%	81.3%	81.3%
Liu and Shah [87]	73.3%	72.1%	-	-	-	82.8%

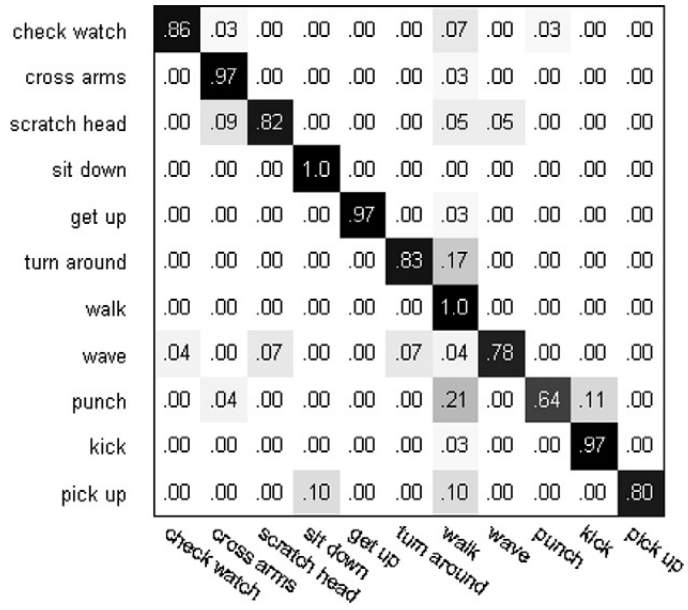


Fig. 4.6 The confusion matrix of Cameras 0–4.

Table 4.2 Classification accuracy comparison between early concatenation fusion strategy and our fusion strategy under different scenarios.

Segment temporal scale	Early concatenation fusion		Our method	
	Camera 0&1	Camera 0-4	Camera 0&1	Cameras 0-4
8 Frames	72.9%	77.0%	74.0%	80.7%
10 Frames [110]	75.3%	78.5%	78.3%	83.9%
18 Frames [9]	80.5%	85.4%	80.1%	88.0%

actions, when fusing five camera views, is shown in Fig. 4.6. The reason why the action "punch" has the lowest recognition rate may be that most "punch" actions are performed by the actors' arms which move in front of the actors' upper bodies so that the variations cannot be reflected on the 2D silhouettes. And the reason why the "turn around" actions are always miss-classified with the "walk" actions may be due to the high similarity between these two actions.

4.4 Conclusion

In this part of work, we propose a novel method for action recognition based on the random forests and a multi-sensor fusion strategy. Since the focus is on classification and multi-sensor fusion, we directly use the silhouettes available in the IXMAS dataset to represent local segments. Our multi-sensor fusion strategy is built to overcome the unequal classification capabilities that would happen due to the high disparity in different observation views. To achieve this, we weight on each camera prediction histogram, inside which each voting segment is first weighted with respect to classification results of all decision trees in the entire random forests. We demonstrate both the multi-sensor fusion results and the single-sensor results using different segment scales, and compare them with the baseline BoW and the NBNN methods. Extensive experimental results show that the proposed algorithm outperforms the above two methods and our 5 camera fused result is comparable with state-of-the-art solutions even though we only use a primitive feature representation.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis mainly focuses on the study of visual feature learning for visual categorization or retrieval tasks. In this section, I conclude my work with some discussions of the main finds through above demonstrated experiments. The following issues have been raised in this thesis.

Segmentation-based image feature learning

Effective segmentation-based image representations are proposed for object recognition tasks. Based on our experiments, the performance of object recognition can be significantly improved if the foreground object regions can be extracted from the original images. Based on recent advanced CPMC image segmentation techniques, it is possible to generate image parts which are close enough to the ground-truth object segments. Two criteria, a facility location term and an entropy term, are proposed to select such image parts among all candidates, where the former can effectively select salient regions and the later can select discriminative regions.

Transfer feature learning

Four feature learning methods are proposed to address the transfer learning scenario.We consider all cross-domain, cross-modality and cross-view transfer learning scenarios. In order to deal with challenges in different transfer learning scenarios, the learning objectives are

formulated based on a cross-domain mapping function or the correspondence information across both domains respectively. The common objective for transfer feature learning is to build appropriate representations that can map cross-view, cross-domain or cross-modality data to a unified feature space. The demonstrated experimental results state that abruptly using data in a different domain can break the smoothness of the original target data, and consequently degrade the categorization performance. By transferring the original data representations in each separate view/domain/modality through the proposed feature learning techniques, the reconstruction error and the overall smoothness over both domains can be preserved.

Multi-View Camera Fusion

We propose a simple though effective action recognition approach based on random forests and a multi-sensor fusion strategy. We aim to effectively fuse actions captured from different viewpoints, and thus build a more powerful recognition system than can be obtained using each single camera sensor. Extensive experimental results suggest the best performance is achieved when fusing actions from all five viewpoints.

5.2 Future Work

5.2.1 Pedestrian Detection

Motivation

The study of pedestrian detection can be traced back 50 years. Recent pedestrian detection techniques are either based on the Hough transform, or sliding windows. The Hough transform [59] is one classical detection technique, and it was first introduced to deal with line or circle detection problems. Hough transform proceeds by converting the input image into a new space, which is known as the *Hough space*. Each point in the Hough space corresponds to a hypothesis which indicates the presence of the target object in the input image at a particular location and other configurations (e.g., scale). If we take the simplest line detection case as an example, the straight line can be described as $y = kx + b$ in the input image, where k and b denote the slope and the intercept of the line respectively. The two parameters (k,b) can be located in a new 2-dimensional space, i.e., the Hough space. Thus, as long as we can confirm the location of (k,b) in the Hough space, we can precisely

locate the straight line the in original image. We understand that each point in the Hough space can represent a line in the original image, vice versa, each point in the original image corresponds to two parameters of a line in the Hough space. Utilizing this property, we can choose any two points on a line in the original image, so that the intersection point of corresponding two lines in the Hough space corresponds to the line in the original image. Recent Hough transform-based object detection techniques are variants of the above case. In the case mentioned above, each pixel in the input image is regarded as a voting element, which can cast votes to hypotheses in the Hough space. While in the extended formulations, voting elements can be pixels, image patches, segments, etc.

Fig. 5.1 Examples of some preliminary pedestrian detection results. The color around the pedestrian area denotes different confidence values (red: high, blue: low). The majority of body areas are covered by colored masks in the first five sub-figures, while the last figure shows some inaccurate detections. One false positive and one false negative detections can be found in the left and the right part of this sub-figure respectively.

On the other hand, sliding windows-based approaches proceed by predicting the presence/absence of the target object in a particular window, which can either be densely sampled or detected in the input image. In the training phase, image patches of pyramid scales are fed to a classifier to train the detector for a particular object. Then, in the testing phase, each sliding window sequentially passes through the detector to obtain a prediction. Finally, decisions are made according to the confidence returned for each window.

Either the Hough transform-based approaches or the sliding windows-based approaches

have their flaws. For the former, it lacks consistent probabilistic model, which lead to both theoretical and practical problems. As analysed in [2], the Hough-transform-based approaches are based on a crude independence assumption that distributions over hypotheses generating voting elements are independent. However, in many applications, there are obvious connections between the hypotheses' origins, e.g., if two voting elements are close, there exists strong correlations between the hypotheses that they associate to. On the other hand, the sliding window-based approaches lead too high computational cost, which prevents their generalization to many applications.

We plan to propose a fundamentally different pedestrian detection approach which replies on figure-ground image segmentations. In principle, such a strategy is superior to either the Hough transform-based approaches or sliding window-based approaches. Thinking in the Hough transform manner, the hypotheses are voting elements themselves in our approach. Due to the reduced number of segments compared to the number of densely sampled sliding windows, the computational complexity and cost can be significantly reduced. We also provide a greedy-based solution based on submodularity, so that the computation time can be further constrained. Specifically, similar as our submodular object recognition work, the basis of our framework constructed by the set of figure-ground segmentations, which are generated by the Constrained Parametric Min-Cuts (CPMC) [17] algorithm. We iteratively select a optimum segment sets from all figure-ground hypotheses according to the criteria defined in the objective function. By proving the submodularity of the objective function, this step can proceed in a greedy manner. Some preliminary detection results are shown in Fig. 5.1.

5.2.2 Cross-Modality Hashing

Introduction and Motivation

Data plays an important role in modern computer and internet industry. On one side, with the emergence of ever growing data, researchers and engineers can turn past impossibility into feasibility to allow people to benefit from new features of technology. The benefits cover many aspects of our daily life. For example, users can get better experience of using search engines with more accurate search results that match their desires; a patient can receive more helpful treatment by referencing more relevant patients with similar symptoms. On the other side, the growing data brings a potentially larger database, from which we retrieve relevant information, thus consequently slows down the query time for each sample. Hashing-based

retrieval models construct a bridge to balance the effectiveness and efficiency. In general, hashing-based methods speed up the querying process by mapping high dimensional data into compact binary hash codes, and consequently conduct approximate nearest neighbor search between query instances and the training database. While sacrificing minimal accuracy, efficiency can be significantly improved (from a few hundreds of images per second to millions of images per second on a standard computer). The main efficiency improvement of applying such hash codes comes from the bit XOR operation when conducting searches in the Hamming space [175]. In order to restrict the accuracy loss within a minimal range, the smoothness property of the resulting hash codes should be guaranteed (i.e., the mapping should generate identical or similar hash codes for data that come from the same category). Four types of methods of generating hash codes are listed as follows.

Random Projection: the Locality Search Hashing (LSH) [51] is one of the most popular hashing models. The basic idea is to use several hash functions to ensure that the probability of collision for the given data applies to a smooth distribution [51]. By utilizing random projections to construct such hash functions, LSH is proven to be effective and embarrassingly simple to implement.

Manifold Learning: Spectral Hashing (SH) [160] is the most popular hashing methods that capture the manifold structure of data. SH computes the hash codes by thresholding a subset of eigenvectors of the Laplacian of the similarity graph [160]. Improved performance are demonstrated by SH over LSH.

Deep Learning: Multiple hidden layer deep architectures have also been considered for generating hash codes. Salakhutdinov and Hinton [127] used multiple stacked Restricted Boltzmann Machines (RBMs) to learn a non-linear mapping between input data and hash codes.

Boosting: Shakhnarovich et al. [132] extended AdaBoost [43] to hashing. Such a proposed Boosting SSC method integrates multiple weak learners, where each weak learner gives a binary prediction (similar or non-similar) of the pair of an input query and a candidate instance in the database, and finally outputs predictions of all weak learners as the resulting hash code.

Comparing to traditional single-modality retrieval tasks, cross-modality retrieval tasks are taking up a growing proportions of users' demands in information retrieval. For example, in order to retrieve a set of desired images, users are likely to query with text descriptions (either a few sentences or a combination of several keywords). In comparison to retrieving with a single word, the increased information that accompanies with the text can refine the returned results so that more desired images can be retrieved to the user. Moreover,

using an image as a query to obtain relevant text descriptions can be considered as a helpful tool to help people understand what the image describes. While the demands of effective cross-modality hashing techniques are increasing fast, research efforts paying on such an area are still at a startup point. In this work, we propose a cross-modality hashing framework, which constructs a cross-modality dictionary pair by selecting modality-invariant dictionary bases, and consequently generates compact and discriminative hash codes upon the modality-invariant sparse image/text representations. By exploiting the monotonicity and submodularity properties of the objective function within the matroid constraint, a highly efficient greedy-based optimization algorithm is adopted to obtain dictionary bases with performance guarantee.

References

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

[2] Barinova, O., Lempitsky, V., and Kholi, P. (2012). On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 34(9):1773–1784.

[3] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

[4] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

[5] Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

[6] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. volume 19, page 153. 1998.

[7] Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115.

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[9] Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[10] Boureau, Y., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566.

[11] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[12] Breiman, L., JH, F., R. A, O., and Stone, C. (1984). Classification and regression trees. In *Wadsworth International Group*.

[13] Cao, L., Liu, Z., and Huang, T. (2010). Cross-dataset action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1998–2005.

[14] Cao, X., Ning, B., Yan, P., and Li, X. (2012). Selecting key poses on manifold for pairwise action recognition. *IEEE Transactions on Industrial Informatics*, 8(1):168–177.

[15] Cao, X., Wang, Z., Yan, P., and Li, X. (2013). Transfer learning for pedestrian detection. *Neurocomputing*, 100(1):51–57.

[16] Carreira, J., Li, F., and Sminchisescu, C. (2012). Object recognition by sequential figure-ground ranking. *International Journal of Computer Vision*, 98(3):243–262.

[17] Carreira, J. and Sminchisescu, C. (2010). Constrained parametric min-cuts for automatic object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3241–3248. IEEE.

[18] Chen, Q., Song, Z., Hua, Y., Huang, Z., and Yan, S. (2012). Hierarchical matching with side information for image classification. In *IEEE International Conference on Computer Vision*, pages 3426–3433.

[19] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61.

[20] Costa Pereira, J., Coviello, E., Doyle, G., Rasiwasia, N., Lanckriet, G., Levy, R., and Vasconcelos, N. (2013). On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine INtelligence*, 36(3):521–535.

[21] Dai, W., Yang, Q., Xue, G., and Yu, Y. (2007). Boosting for transfer learning. In *International Conference on Machine Learning*, pages 193–200.

[22] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893.

[23] Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5.

[24] Daumé, H. (2007). Frustratingly easy domain adaptation. In *ACL*, volume 1785, page 1787.

[25] Dikmen, M., Ning, H., Lin, D., Cao, L., Le, V., Tsai, S., Lin, K., Li, Z., Yang, J., Huang, T., et al. (2008). Surveillance event detection. In *TRECVID Video Evaluation Workshop*.

[26] Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 126–135.

[27] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72.

[28] Dong, J., Xia, W., Chen, Q., Feng, J., Huang, Z., and Yan, S. (2013). Subcategory-aware object classification. In *IEEE International Conference on Computer Vision*, pages 827–834.

[29] Duan, L., Tsang, I., and Xu, D. (2012). Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 34(3):465–479.

[30] Duan, L., Tsang, I., Xu, D., and Maybank, S. (2009). Domain transfer svm for video concept detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1375–1381.

[31] Duchenne, O., Laptev, I., Sivic, J., Bach, F., and Ponce, J. (2009). Automatic annotation of human actions in video. In *International Conference on Computer Vision*, pages 1491–1498.

[32] Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *International Conference on Computer Vision*, pages 726–733.

[Everingham et al.] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[34] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

[35] Fathi, A. and Mori, G. (2008). Action recognition by learning mid-level motion features. In *IEEE Conference on Computer Vision and Pattern Recognition.*, pages 1–8.

[36] Fei-Fei, L. (2006). Knowledge transfer in learning to recognize visual objects classes. In *International Conference on Development and Learning*, pages 1–8.

[37] Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 28(4):594–611.

[38] Fei-Fei, L., Fergus, R., and Perona, P. (2007). 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.

[39] Feng, F., Wang, X., and Li, R. (2014). Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the ACM International Conference on Multimedia*, pages 7–16.

[40] Fernando, B. and Tuytelaars, T. (2013). Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation. In *International Conference on Computer Vision*, pages 2544–2551.

[41] Ferrari, V., Jurie, F., and Schmid, C. (2010). From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303.

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

[43] Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

[44] Frieze, A. M. (1974). A cost function property for plant location problems. *Mathematical Programming*, 7(1):245–248.

[45] Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *IEEE International Conference on Computer Vision*, pages 670–677.

[46] G. Qi, C. Aggarwal, Y. R. Q. T. S. C. and Huang, T. (2011). Towards cross-category knowledge propagation for learning visual concepts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 897–904.

[47] Galvão, R. D. (2004). Uncapacitated facility location problems: contributions. *Pesquisa Operacional*, 24(1):7–38.

[48] Gao, X., Wang, X., Li, X., and Tao, D. (2011). Transfer latent variable model based on divergence analysis. *Pattern Recognition*, 44:2358–2366.

[49] Gavrila, D. and Davis, L. (1996). 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80.

[50] Gilbert, A., Illingworth, J., and Bowden, R. (2011). Action recognition using mined hierarchical compound features. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 33(5):883–897.

[51] Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529.

[52] Golub, G. H., Hansen, P. C., and O'Leary, D. P. (1999). Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194.

[53] Gould, S., Fulton, R., and Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *IEEE International Conference on Computer Vision*, pages 1–8.

[54] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. *Caltech Technical Report*.

[55] H. Kuehne, H. Jhuang, E. G. T. P. and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563.

[56] Harzallah, H., Jurie, F., and Schmid, C. (2009). Combining efficient object localization and image classification. In *IEEE International Conference on Computer Vision*, pages 237–244.

[57] He, X., Zemel, R. S., and Carreira-Perpinán, M. A. (2004). Multiscale conditional random fields for image labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[58] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[59] Hough, P. V. (1959). Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, pages 554–558.

[60] I. Laptev, M. Marszalek, C. S. and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[61] Ikizler-Cinbis, N. and Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *European Conference on Computer Vision*, pages 494–507.

[62] J. Zheng, Z. Jiang, P. P. and Chellappa, R. (2012). Cross-view action recognition via a transferable dictionary pair. In *British Machine Vision Conference*.

[63] Jégou, H., Douze, M., and Schmid, C. (2010). Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336.

[64] Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

[65] Jiang, Z. and Davis, L. S. (2013). Submodular salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2043–2050.

[66] Jiang, Z., Lin, Z., and Davis, L. (2011). Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1697–1704.

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

[68] Jinjun, W., Jianchao, Y., Kai, Y., Fengjun, L., Thomas, H., and Yihong, G. (2010). Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360–3367.

[69] Jolliffe, I. (2002). *Principal Component Analysis*. Springer Science & Business Media.

[70] Junejo, I., Dexter, E., Laptev, I., and Pérez, P. (2011). View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 33(1):172–185.

[71] Junejo, I. N., Dexter, E., Laptev, I., and Pérez, P. (2008). Cross-view action recognition from temporal self-similarities. In *European Conference on Computer Vision*, pages 1–14.

[72] Khan, I., Saffari, A., and Bischof, H. (2009). Tvgraz: Multi-modal learning of object categories by combining textual and visual features. In *AAPR Workshop*, pages 213–224.

[73] Krause, A. and Golovin, D. (2014). Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press.

[74] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[75] L. Duan, D. Xu, I. T. and Luo, J. (2012). Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 34(9):1667–1680.

[76] Ladicky, L., Russell, C., Kohli, P., and Torr, P. H. (2009). Associative hierarchical crfs for object class image segmentation. In *IEEE International Conference on Computer Vision*, pages 739–746.

[77] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178.

[78] Lazic, N., Givoni, I., Frey, B., and Aarabi, P. (2009). Floss: Facility location for subspace segmentation. In *IEEE International Conference on Computer Vision*, pages 825–832.

[79] Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808.

[80] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429.

[81] Li, F., Carreira, J., and Sminchisescu, C. (2010). Object recognition as ranking holistic figure-ground hypotheses. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1719.

[82] Li, R. and Zickler, T. (2012). Discriminative virtual views for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2862.

[83] Li, T. and Ding, C. (2006). The relationships among various nonnegative matrix factorization methods for clustering. In *IEEE International Conference on Data Mining*, pages 362–371.

[84] Lin, Y.-C., Hu, M.-C., Cheng, W.-H., Hsieh, Y.-H., and Chen, H.-M. (2012). Human action recognition and retrieval using sole depth information. In *ACM Multi-media*, pages 1053–1056.

[85] Lin, Z., Jiang, Z., and Davis, L. S. (2009). Recognizing actions by shape-motion prototype trees. In *International Conference on Computer Vision*, pages 444–451.

[86] Liu, J., Luo, J., and Shah, M. (2009a). Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003.

[87] Liu, J. and Shah, M. (2008). Learning human actions via information maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[88] Liu, J., Shah, M., Kuipers, B., and Savarese, S. (2011). Cross-view action recognition via view knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3209–3216.

[89] Liu, J., Yang, Y., and Shah, M. (2009b). Learning semantic visual vocabularies using diffusion distance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 461–468.

[90] Liu, L., Shao, L., and Rockett, P. (2013). Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, 46(7):1810–1818.

[91] Liu, M.-Y., Tuzel, O., Ramalingam, S., and Chellappa, R. (2014). Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(1):99–112.

[92] Liu, W. and Tao, D. (2013). Multiview hessian regularization for image annotation. *IEEE Transactions on Image Processing*, 22(7):2676–2687.

[93] Loui, A., Luo, J., Chang, S.-F., Ellis, D., Jiang, W., Kennedy, L., Lee, K., and Yanagawa, A. (2007). Kodak's consumer video benchmark data set: concept definition and annotation. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 245–254.

[94] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[95] Lv, F. and Nevatia, R. (2007). Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[96] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009a). Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, pages 689–696.

[97] Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. R. (2009b). Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040.

[98] Maji, S., Berg, A. C., and Malik, J. (2013). Efficient classification for additive kernel svms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):66–77.

[99] Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415.

[100] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge University Press.

[101] Marial, J., Leordeanu, M., Bach, F., Hebert, M., and Ponce, J. (2008). Discriminative sparse image models for class-specific edge detection and image interpretation. In *European Conference on Computer Vision*, pages 43–56.

[102] Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936.

[103] Mikolajczyk, K. and Uemura, H. (2008). Action recognition with motion-appearance vocabulary forest. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[104] Minsky, M. and Seymour, P. (1987). *Perceptrons - Expanded Edition: An Introduction to Computational Geometry*. MIT press Boston.

[105] N. Dalal, B. T. and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, pages 428–441.

[106] Nebel, J.-C., Lewandowski, M., Thévenon, J., Martínez, F., and Velastin, S. (2011). Are current monocular computer vision systems for human action recognition suitable for visual surveillance applications? In *Advances in Visual Computing*, pages 290–299.

[107] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set function. *Mathematical Programming*, 14(1):265–294.

[108] Ng, A. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72.

[109] Nilsback, M.-E. and Zisserman, A. (2006). A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454.

[110] Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*, pages 490–503.

[111] Orrite, C., Rodríguez, M., and Montañés, M. (2011). One-sequence learning of human actions. *Human Behavior Unterstanding*, pages 40–51.

[112] Pan, S., Kwok, J., and Yang, Q. (2008). Transfer learning via dimensionality reduction. In *Association for the Advancement of Artifical Intelligence*, pages 677–682.

[113] Pan, S. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[114] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, pages 40–44.

[115] Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, pages 143–156.

[116] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[117] Qiu, Q., Patel, V. M., Turaga, P., and Chellappa, R. (2012). Domain adaptive dictionary learning. In *European Conference on Computer Vision*, pages 631–645.

[118] Quack, T., Ferrari, V., Leibe, B., and Van Gool, L. (2007). Efficient mining of frequent and distinctive feature configurations. In *IEEE International Conference on Computer Vision*, pages 1–8.

[119] Rabinovich, A., Belongie, S., Lange, T., and Buhmann, J. M. (2006). Model order selection and cue combination for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1137.

[120] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. (2007). Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine Learning*, pages 759–766.

[121] Raptis, M. and Soatto, S. (2010). Tracklet descriptors for action modeling and video analysis. In *European Conference on Computer Vision*, pages 577–590.

[122] Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G. R., Levy, R., and Vasconcelos, N. (2010). A new approach to cross-modal multimedia retrieval. In *ACM International Conference on Multimedia*, pages 251–260.

[123] Reddy, K. K., Liu, J., and Shah, M. (2009). Incremental action recognition using feature-tree. In *International Conference on Computer Vision*, pages 1010–1017.

[124] S. Liwicki, S. Zafeiriou, G. T. and Pantic, M. (2012). Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10):1624–1636.

[125] S. Xiang, F. Nie, G. M. C. P. and Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11):1738–1754.

[126] S. Zafeiriou, G. Tzimiropoulos, M. P. and Stathaki, T. (2012). Regularized kernel discriminant analysis with a robust kernel for face recognition and verification. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):526–534.

[127] Salakhutdinov, R. and Hinton, G. E. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. In *International Conference on Artificial Intelligence and Statistics*, pages 412–419.

[128] Salton, G. (1971). The smart retrieval system experiments in automatic document processing. *Experiments in Automatic Document Processing*.

[129] Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *International Conference on Patter Recognition*, pages 32–36.

[130] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *ACM International Conference on Multimedia*, pages 357–360.

[131] Shaban, A., Rabiee, H. R., Farajtabar, M., and Ghazvininejad, M. (2013). From local similarity to global coding: An application to image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2794–2801.

[132] Shakhnarovich, G., Viola, P., and Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *International Conference on Computer Vision*, pages 750–757.

[133] Shao, L. and Chen, X. (2010). Histogram of body poses and spectral regression discriminant analysis for human action categorization. In *British Machine Vision Conference*, pages 1–11.

[134] Shao, L., Liu, L., and Li, X. (2014a). Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 25(7):1359–1371.

[135] Shao, L., Wu, D., and Chen, X. (2011). Action recognition using correlogram of body poses and spectral regression. In *International Conference on Image Processing*, pages 209–212.

[136] Shao, L., Zhu, F., and Li, X. (2014b). Transfer learning for visual categorization: A survey. *IEEE Transactions on Neural Networks and Learning Systems*. doi: 10.1109/TNNLS.2014.2330900.

[137] Sharma, A., Kumar, A., Daume, H., and Jacobs, D. W. (2012). Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167.

[138] Sharma, G., Jurie, F., Schmid, C., et al. (2013). Expanded parts model for human attribute and action recognition in still images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–659.

[139] Shen, L. and Bai, L. (2004). Adaboost gabor feature selection for classification. In *Image and Vision Computing NewZealand*, pages 77–83.

[140] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 22(8):888–905.

[141] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23.

[142] Snoek, C. G. and Worring, M. (2005). Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1):5–35.

[143] Song, Y., Morency, L.-P., and Davis, R. (2013). Action recognition by hierarchical sequence summarization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3562–3569.

[144] Souvenir, R. and Babbs, J. (2008). Learning the viewpoint manifold for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7.

[145] Su, Y. and Jurie, F. (2012). Improving image classification using semantic attributes. *International |Journal of Computer Vision*, 100(1):59–77.

[146] Tang, J., Liu, X., Cheng, H., and Robinette, K. (2011). Gender recognition using 3-d human body shapes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):898–908.

[147] Tao, D., Li, X., Wu, X., and Maybank, S. J. (2007). General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700–1715.

[148] Tao, D., Li, X., Wu, X., and Maybank, S. J. (2009). Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):260–274.

[149] Tao, D., Tang, X., Li, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099.

[150] Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283.

[151] Uemura, H., Ishikawa, S., and Mikolajczyk, K. (2008). Feature tracking and motion compensation for action recognition. In *British Machine Vision Conference*, pages 1–10.

[152] Varma, M. and Babu, B. R. (2009). More generality in efficient multiple kernel learning. In *International Conference on Machine Learning*, pages 1065–1072.

[153] Wang, H., Klaser, A., Schmid, C., and Liu, C. L. (2011a). Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176.

[154] Wang, H., Nie, F., Huang, H., and Ding, C. (2011b). Dyadic transfer learning for cross-domain image classification. In *International Conference on Computer Vision*, pages 551–556.

[155] Wang, K., He, R., Wang, W., Wang, L., and Tan, T. (2013). Learning coupled feature spaces for cross-modal matching. In *IEEE International Conference on Computer Vision*, pages 2088–2095.

[156] Wang, Y. and Mori, G. (2009). Max-margin hidden conditional random fields for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 872–879.

[157] Wang, Y. and Mori, G. (2011). Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 33(7):1310–1323.

[158] Weinland, D., Boyer, E., and Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *International Conference on Computer Vision*, pages 1–7.

[159] Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257.

[160] Weiss, Y., Torralba, A., and Fergus, R. (2009). Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760.

[161] Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 31(2):210–227.

[162] Wu, D., Zhu, F., and Shao, L. (2012). One shot learning gesture recognition from rgbd images. In *IEEE Conference on Computer Vision and Pattern Recognition workshop*, pages 7–12.

[163] Wu, X., Xu, D., Duan, L., and Luo, J. (2011). Action recognition using context and appearance distribution features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 489–496.

[164] Xiang, S., Nie, F., Song, Y., and Zhang, C. (2008). Contour graph based human tracking and action sequence recognition. *Pattern Recognition*, 41(12):3653–3664.

[165] Xu, C., Tao, D., and Xu, C. (2014). Large-margin multi-view information bottleneck. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1559–1572.

[166] Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S. (2007). Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51.

[167] Yang, J., Yan, R., and Hauptmann, A. (2007). Cross-domain video concept detection using adaptive svms. In *ACM International Conference Multimedia*, pages 188–197.

[168] Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801.

[169] Yang, J., Yu, K., and Huang, T. (2010). Supervised translation-invariant sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3524.

[170] Yang, L., Jin, R., Sukthankar, R., and Jurie, F. (2008). Unifying discriminative visual codebook generation with classifier training for object category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[171] Yao, A., Gall, J., and Van Gool, L. (2010). A hough transform-based voting framework for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2061–2068.

[172] Yao, A., Gall, J., and Van Gool, L. (2012). Coupled action recognition and pose estimation from multiple views. *IEEE International Journal of Computer Vision*, 100(1):16–37.

[173] Yu, F. X., Ji, R., Tsai, M.-H., Ye, G., and Chang, S.-F. (2012). Weak attributes for large-scale image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2949–2956.

[174] Yu, G., Yuan, J., and Liu, Z. (2011). Unsupervised random forest indexing for fast action search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 865–872.

[175] Zhang, D., Wang, J., Cai, D., and Lu, J. (2010). Self-taught hashing for fast similarity search. In *ACM International SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25.

[176] Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2126–2136.

[177] Zhang, Q. and Li, B. (2010). Discriminative k-svd for dictionary learning in face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2698.

[178] Zhang, W., Surve, A., Fern, X., and Dietterich, T. (2009). Learning non-redundant codebooks for classifying complex objects. In *International Conference on Machine Learning*, pages 1241–1248.

[179] Zheng, J., Jiang, Z., Phillips, J., and Chellappa, R. (2012a). Cross-view action recognition via a transferable dictionary pair. In *British Machine Vision Conference*, page 7.

[180] Zheng, J., Jiang, Z., Phillips, P. J., and Chellappa, R. (2012b). Cross-view action recognition via a continuous virtual path. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2697.

[181] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004a). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, pages 321–328.

[182] Zhou, D., Weston, J., Gretton, A., Bousquet, O., and Schölkopf, B. (2004b). Ranking on data manifolds. *Advances in Neural Information Processing Systems*, pages 169–176.

[183] Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., and Paisley, J. W. (2009). Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems*, pages 2295–2303.

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

[185] Zhu, F., Shao, L., and Lin, M. (2013). Multi-view action recognition using local similarity random forests and sensor fusion. *Pattern Recognition Letters*, 34(1):20–24.