ABSTRACT

Title of dissertation: Domain Transfer Learning for Object and Action Recognition

Jingjing Zheng, Doctor of Philosophy, 2015

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Visual recognition has always been a fundamental problem in computer vision. Its task is to learn visual categories using labeled training data and then identify unlabeled new instances of those categories. However, due to the large variations in visual data, visual recognition is still a challenging problem. Handling the variations in captured images is important for real-world applications where unconstrained data acquisition scenarios are widely prevalent.

In this dissertation, we first address the variations between training and testing data. Particularly, for cross-domain object recognition, we propose a Grassmann manifold-based domain adaptation approach to model the domain shift using the geodesic connecting the source and target domains. We further measure the distance between two data points from different domains by integrating the distance of their projections through all the intermediate subspaces along the geodesic. Our proposed approach that exploits all the intermediate subspaces along the geodesic produces a more accurate metric. For cross-view action recognition, we present two effective approaches to learn transferable dictionaries and view-invariant sparse representations. In the first approach, we learn a set of transferable dictionaries where each dictionary corresponds to one camera view. The set of dictionaries is learned simultaneously from sets of correspondence videos taken at different views with the

aim of encouraging each video in the set to have the same sparse representation. In the second approach, we relaxes this constraint by encouraging correspondence videos to have similar sparse representations. In addition, we learn a common dictionary that is incoherent to view-specific dictionaries for cross-view action recognition. The set of view-specific dictionaries is learned for specific views while the common dictionary is shared across different views. In this way, we can align view-specific features in the sparse feature spaces spanned by the view-specific dictionary set and transfer the view-shared features in the sparse feature space spanned by the common dictionary.

In order to handle the more general variations in captured images, we also exploit the semantic information to learn discriminative feature representations for visual recognition. Class labels are often organized in a hierarchical taxonomy based on their semantic meanings. We propose a novel multi-layer hierarchical dictionary learning framework for region tagging. Specifically, we learn a node-specific dictionary for each semantic label in the taxonomy and preserve the hierarchial semantic structure in the relationship among these node-dictionaries. Our approach can also transfer knowledge from semantic label at higher levels to help learn the classifiers for semantic labels at lower levels. Moreover, we exploit the semantic attributes for boosting the performance of visual recognition. We encode objects or actions based on attributes that describe them as high-level concepts. We consider two types of attributes. One type of attributes is generated by humans, while the second type is data-driven attributes extracted from data using dictionary learning methods. Attribute-based representation may exhibit variations due to noisy and redundant attributes. We propose a discriminative and compact attribute-based representation by selecting a subset of discriminative attributes from a large attribute set. Three attribute selection criteria are proposed and formulated as a submodular optimization problem. A greedy optimization algorithm is presented and its solution is guaranteed to be at least (1-1/e)-approximation to the optimum.

Domain Transfer Learning for Object and Action Recognition

by

Jingjing Zheng

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Piya Pal
Professor Min Wu
Professor Larry Davis
Professor Amitabh Varshney

Dedication

To my parents, Guangqing Zheng and Cuilan Li, my grandparents, Hongyun Zheng, and Guinan Dong, and my brother Jiajia Zheng.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Rama Chellappa for his guidance and mentoring over the years. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Min Wu, Prof. Piya Pal, Prof. Larry Davis, and Prof. Amitabh Varshney, for their valuable feedback on this dissertation, but also for the hard question which incented me to widen my research from various perspectives.

I am grateful to Dr. Jonathan Phillips for his constructive comments and practical advice. I am also thankful to him for reading my papers, commenting on my views and helping me understand and enrich my ideas.

I also would like to thank Dr. Zhuolin Jiang for his encouragement and support at different stages of my research. He enlightened me the first glance of research and his work ethic inspired me to work hard on the dissertation. I am also thankful to him for a lot of fruitful discussions and collaboration on research projects.

I also indebted to the members of the Center for Automatic Research with whom I have interacted. I would like to acknowledge Dr. Vishal Patel, Dr. Sima Taheri, Dr. Qiang Qiu, Dr. Jie Ni, Dr. Ming Du, Dr. Ruonan Li, Dr. Sumit Shekhar, Dr. Jaishanker Pillai, Dr. Raghuraman Gopalan and Jun-Cheng Chen for

teaching me numerous things about computer vision. I also thank Janice Perrone, Melanie Prange and Arlene Schenk for all the administrative help.

Many friends have helped me stay same through these difficult years. Their support and care helped me overcome setbacks and stay focused on my graduate study. Particularly, I would like to acknowledge one of my friends Xin Yang who was always there to listen and give advice.

Last but not the least, I would like to thank my family: my parents and brother for supporting me spiritually throughout writing this thesis and my my life in general.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Visual recognition has always been a fundamental problem in computer vision. Its task is to learn visual categories using labeled training data and then identify unlabeled new instances of those categories. Many vision task relies on the ability to recognize objects, scenes and categories. Visual recognition itself has many real-world applications, such as face identification, object recognition, action recognition, image search and retrieval, video surveillance and so on. Meanwhile, there exist a variety of literature on feature extraction and learning methods for recognition. However, visual recognition is challenging due to the large variations in captured images. For example, in face and object recognition applications, images may be acquired from different viewpoints and illumination conditions. Other factors resulting in the variation may include partial occlusions and unrelated background clutter. Different instances of the same category can exhibit significant variations in appearance. Handling the variations in captured images is important for real-world applications where unconstrained data acquisition scenarios are widely prevalent.

Recently, a very fruitful line of work is proposed to address the variances between training (source domain) and testing data (test domain) [95, 51, 33, 32].

The varia between the training and testing data may be caused by many factors including viewpoints, illuminations, and background clutters. These varia will result in distribution shift between the training and testing domain. In this scenario, most traditional visual recognition approaches that directly apply the classifiers trained from the training domain to the test domain often yields poor recognition performance. What's worse, the instances of the same category between the training and test domains may be much larger than the varia among instances of the same category within each domain. This is referred the domain adaptation problem.

Another line of work for dealing with varia in captured images focus on learning discriminative feature representations [1, 121, 43, 23, 53, 66, 60]. Feature representation are critical for the visual recognition performance. A good discriminative feature representation is often the one that is robust to varia of instances of the same category. Dictionary learning methods have been proposed to learn discriminative sparse representations for visual recognition [1, 121, 43]. These dictionary learning methods can learn both representative and discriminative dictionaries, and the corresponding sparse coefficients are discriminative for classification. Semantic information has also been exploited in [53, 66, 60] to learn robust feature representations. The semantic information includes the structured taxonomy of class labels, and the high-level concepts called attributes.

In this dissertation, a Grassmann manifold-based algorithm for cross-domain object and face recognition was first presented. This approach models the domain shift using the geodesic connecting the source and target domains on a Grassmann manifold. For the action recognition problem, domain shift may be caused by

changes in camera viewpoints and background clutter. In order to solve the domain shift caused by camera views, two dictionaries corresponding to two camera views are learned for cross-view action recognition. The problem of learning representative and discriminative features for images and videos is also studied in this dissertation. Semantic class labels are often organized in a hierarchical taxonomy based on their semantic meanings. In order to exploit the semantic information in the taxonomy, a novel multi-layer hierarchical dictionary learning framework is proposed for image tagging. The proposed method transfers knowledge from semantic label at higher levels to help learn the classifiers for semantic labels at lower levels. Finally, the concept of attributes and its application for representation and recognition of action videos is introduced. In order to derive effective attribute-based representation, a novel method on attributes learning and selection for action recognition is further presented.

1.2 A Grassmann Manifold-based Domain Adaptation Approach

In the first part of the dissertation, we consider the problem of domain adaptation in object and face recognition [142]. Recently a Grassamnn manifold-based domain adaptation algorithm that models the domain shift using intermediate subspaces along the geodesic connecting the source and target domains was presented in [33]. We build upon this work and propose replacing the step of concatenating feature projections on a very few sampled intermediate subspaces by directly integrating the distances between feature projections along the geodesic. The proposed

approach considers all the intermediate subspaces along the geodesic. Thus, it is a more principled way of quantifying the cross-domain distance. Our approach has two major advantages. Experimental results on two standard datasets show that the proposed algorithm yields favorable performance over previous approaches. Note that while this work was under review for ICPR 2012, we became aware of a paper presented at CVPR 2012 by [32] discussing a similar approach.

1.3 Transferable Dictionary Learning for Action Recognition

In the second part of this dissertation, we study the problem of cross-view action recognition where the domain shift is caused by camera viewpoints in [141, 138]. Discriminative appearance features are effective for recognizing actions in a fixed view, but may be poor at generalizing to a new view. We present two effective approaches to learn transferable dictionaries for robust action recognition across views. In the first approach, we learn a set of transferable dictionaries where each dictionary corresponds to one camera view. The set of dictionaries is learned simultaneously from sets of correspondence videos taken at different views with the aim of encouraging each video in the set to have the same sparse representation. In the second approach, we also learn a common dictionary that is incoherent to view-specific dictionaries for cross-view action recognition. The set of view-specific dictionaries is learned for specific views while the common dictionary is shared across different views. Our approach represents videos in each view using both the corresponding view-specific dictionary and the common dictionary. More importantly, it

encourages the set of videos taken from different views of the same action to have similar sparse representations. In this way, we can align view-specific features in the sparse feature spaces spanned by the view-specific dictionary set and transfer the view-shared features in the sparse feature space spanned by the common dictionary. The learned common dictionary not only has the capability to represent actions from unseen views, but also makes our approach effective in a semi-supervised setting where no correspondence videos exist and only a few labels exist in the target view. Extensive experiments using three public datasets demonstrate that the proposed approach outperforms recently developed approaches for cross-view action recognition.

1.4 Semantic Taxonomy Aware Dictionary Learning for Image Tagging

In the third part of this dissertation, we exploit the semantic taxonomy to learn discriminative sparse representations for image tagging [139]. Tags of image regions are often arranged in a hierarchical taxonomy based on their semantic meanings. Using the given tag taxonomy, we propose to jointly learn multi-layer hierarchical dictionaries and corresponding linear classifiers for region tagging. Specifically, we generate a node-specific dictionary for each tag node in the taxonomy, and then concatenate the node-specific dictionaries from each level to construct a level-specific dictionary. The hierarchical semantic structure among tags is preserved in the relationship among node-dictionaries. Simultaneously, the sparse codes obtained using

the level-specific dictionaries are summed up as the final feature representation to design a linear classifier. Our approach not only makes use of sparse codes obtained from higher levels to help learn the classifiers for lower levels, but also encourages the tag nodes from lower levels that have the same parent tag node to implicitly share sparse codes obtained from higher levels. Experimental results using three benchmark datasets show that the proposed approach yields the best performance over recently proposed methods.

1.5 Attribute Learning and Selection for Visual Recognition

In the final part of this dissertation, we exploit the semantic attributes for boosting the performance of visual recognition [140]. In real-world visual recognition problems, low-level features cannot adequately characterize the semantic content in images, or the spatio-temporal structure in videos. In this work, we encode objects or actions based on attributes that describe them as high-level concepts. We consider two types of attributes. One type of attributes is generated by humans, while the second type is data-driven attributes extracted from data using dictionary learning methods. Attribute-based representation may exhibit variations due to noisy and redundant attributes. We propose a discriminative and compact attribute-based representation by selecting a subset of discriminative attributes from a large attribute set. Three attribute selection criteria are proposed and formulated as a submodular optimization problem. A greedy optimization algorithm is presented and its solution is guaranteed to be at least (1-1/e)-approximation to the optimum. Experimental

results on four public datasets demonstrate that the proposed attribute-based representation significantly boosts the performance of visual recognition algorithms and outperforms most recently proposed recognition approaches.

1.6 Contributions of the Dissertation

In this dissertation, we make the following contributions.

- We have extensively studied the problem of domain adaptation for cross-domain face and object recognition. Our proposed manifold-based domain adaptation approach has two advantages. First, it avoids ad-hoc sampling of intermediate subspaces in [33]. Second, it is more expressive because it implicitly projects data onto all the subspaces along the geodesic and smoothly accumulate the distance between data projections along the geodesic. Lastly, it does not suffer from information lass that occurs in [33] due to discrete sampling.

- We present two dictionary learning approaches for cross-view action recognition by transferring sparse representations across views. The first approach directly exploits the video-level correspondence and bridges the gap of sparse representations of pairs of videos taken from different views of the same action. The second approach simultaneously learns a set of view-specific dictionaries to exploit the video-level correspondence across views and a common dictionary to model the common patterns shared by different views. Both frameworks are very general and can be applied to cross-view and multi-view action

recognition under both unsupervised and supervised settings.

- We present a multi-layer supervised dictionary learning framework that simultaneously learns multi-layer dictionaries and classifiers. We are the first to use the supervised dictionary learning to explore the semantic structure among tags, which not only takes advantages of the compactness and efficiency of dictionary learning, but also explores different group structures among image regions. Our approach proposes to sum up sparse codes from different levels as the feature representation to learn a linear classifier, which enables us to make use of discriminative information encoded in sparse codes from different levels. Our approach is robust to datasets with unbalanced tag classes.

- We exploit human-labeled attributes and data-driven attributes for improving the performance of visual recognition. We propose three attribute selection criteria for the selection of discriminative and compact attributes. We formulate the selection procedure as one of optimizing a submodular function based on the entropy rate of a random walk and weighted maximum coverage function. The selected attributes not only have strong and similar discrimination capability for all pairwise classes, but also maximize the sum of largest discrimination capability that each pairwise classes can obtain from the selected attributes.

1.7 Organization of the Dissertation

The rest of the dissertation proposal is organized as follows. We first introduce a Grassman manifold-based domain adaptation approach for cross-domain object recognition in Chapter 2. In order to solve the domain shift caused by camera views, we present two novel methods for cross-view action recognition in Chapter 3. In chapter 4, we describe a hierarchical dictionary learning method for region tagging. In Chapter 5, we present an attribute-based representation to overcome the large variations in low-level features. We conclude the dissertation and discuss future directions in chapter 6.

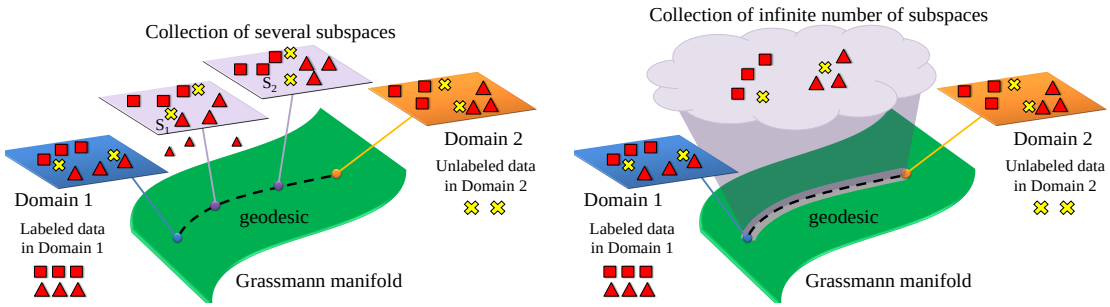
Chapter 2: A Grassmann Manifold-based Domain Adaptation Approach

2.1 Related Work

Traditional visual object recognition methods assume that testing and training data are sampled from the same distribution. However, in practice, the training and testing data are captured under different conditions and exhibit different distributions. Failing to model this shift often leads to inferior results. Methods that can handle domain shift are essential for improving the recognition performance. This is referred as the domain adaptation problem.

Several methods have been proposed to handle domain shift for support vector machines [127, 41, 19]. In the field of visual object recognition, [95, 51] computed domain-invariant metrics to quantify the similarity between objects of different domains. Recently, Gopalan et al. modeled the domain shift using the geodesic connecting the source and target domains on a Grassmann manifold [33]. The key idea was to synthesize intermediate domains using intermediate subspaces along the geodesic and represent an object by concatenating its projections on these subspaces.

In this chapter, we present an alternative Grassmann manifold-based approach



(a) (b)

Figure 2.1: Finite sampling versus continuous integration. (a) Gopalan et al. [33] sample several intermediate subspaces along the geodesic connecting two domains on the Grassmann manifold, represent the data by the concatenation of data projections on the sampled subspaces, and perform cross-domain class analysis using the concatenated representation. (b) We measure the similarity between two data points from different domains by integrating the distance of their projections to the intermediate subspace along the geodesic. We consider all the intermediate subspace which renders a smoother metric.

to address the domain adaptation problem. Specifically, we propose replacing the concatenation of very *few* intermediate subspace projections in [33] by integrating the distance between feature projections on *all* the intermediate subspaces along the geodesic. Fig 2.1 illustrates the difference between our approach and that of [33]. Note that we developed this work independent of [32] which discussed a similar approach.

2.2 Geodesic on the Grassman Manifold

A Grassmann manifold $\mathcal{G}_{n,d}$ is the set of all the d-dimensional subspaces of the vector space $\mathbb{R}^n$. We denote a subspace $\mathcal{S} \in \mathcal{G}_{n,d}$ using a matrix S in $\mathbb{R}^{n\times d}$ whose columns are orthogonal and form a basis for this subspace. Note that if S is right-multiplied by a d-dimensional orthogonal matrix, it still denotes $\mathcal{S}$ because the subspace spanned by the columns of S remains the same.

Let S_0 and S_1 be two matrices in $\mathbb{R}^{n\times d}$ whose columns are orthogonal bases for the d-dimensional subspaces $\mathcal{S}_0$ and $\mathcal{S}_1$ respectively. Let $U_1\Gamma V_1^T$ be a singular value decomposition (SVD) of the $d \times d$ matrix $S_0^T S_1$. The geodesic $\psi(t)$ on the Grassmann manifold $\mathcal{G}_{n,d}$ starting from $\mathcal{S}_0$ to $\mathcal{S}_1$ is given by

$$\psi(t) = Qexp(tB)J \quad \text{s.t.} \begin{cases} \psi(0) = S_0 \\ \\ \psi(1) = S_1 \end{cases} \tag{2.1}$$

where $J = \begin{bmatrix} I_d \\ \\ O_{n-d,d} \end{bmatrix}$, I_d is a $d \times d$ identity matrix, and $O_{n-d,d}$ is a matrix with all zeros [109]. Here, Q is an orthogonal matrix with determinant $+1$ and is given by

$$Q = \begin{bmatrix} S_{01} - I_d \\ \\ S_{02} \end{bmatrix} [I_d - S_{01}^T]^{-1}[S_{01}^T - I_d S_{02}^T]. \tag{2.2}$$

The matrices $S_{01} \in \mathbb{R}^{d\times d}$ and $S_{02} \in \mathbb{R}^{(n-d)\times d}$ are the upper and lower parts of S_0 respectively, i.e., $S_0 = \begin{bmatrix} S_{01} \\ \\ S_{02} \end{bmatrix}$, and the matrix B is asymmetric and block-diagonal

given by $B = \begin{bmatrix} O_{d,d} & A^T \\ -A & O_{d,d} \end{bmatrix}$ where $A \in \mathbb{R}^{(n-d)\times d}$.

Instead of directly calculating $\psi(t)$, we use the approach proposed by Gallivan et al. [30] which calculates the equivalent geodesic $\bar{\psi}(t) = \psi(t)U_1$ connecting $\mathcal{S}_0$ and $\mathcal{S}_1$ such that $\bar{\psi}(0) = S_0U_1$ and $\bar{\psi}(1) = S_1V_1$. The intuition behind this is that the subspaces represented by $\psi(t), S_0,$ and S_1 will be the same when these matrices are right multiplied by an orthogonal matrix.

Now the geodesic $\bar{\psi}(t)$ connecting $\mathcal{S}_0$ and $\mathcal{S}_1$ is given by

$$\bar{\psi}(t) = Q\exp(tB)JU_1 \quad \text{s.t.} \begin{cases} \bar{\psi}(0) = S_0U_1 \\ \bar{\psi}(1) = S_1V_1 \end{cases} \tag{2.3}$$

Using the results pertaining to the geodisc on Grassmann manifold [30], the geodesic $\bar{\psi}(t)$ can be further simplified to

$$\bar{\psi}(t) = Q \begin{bmatrix} U_1\Gamma(t) \\ -\tilde{U}_2\Sigma(t) \end{bmatrix}, \tag{2.4}$$

where $\tilde{U}_2 \in \mathbb{R}^{(n-d)\times d}$ is made up of d orthogonal columns. The derivation of $\tilde{U}_2$ makes use of the boundary condition $\bar{\psi}(1) = S_1V_1$ and will be given. The matrices $\Gamma(t)$,$\Sigma(t) \in \mathbb{R}^{d\times d}$ are diagonal with diagonal elements being $\gamma_i = \cos(t\theta_i)$ and $\sigma_i = \sin(t\theta_i)$ respectively where $0 \leq \theta_1 \leq ... \leq \theta_d \leq \pi/2$. Note that the $\{\theta_i\}_{i=1}^d$ form the rotation angles from S_0 to S_1. We use Θ to denote the diagonal matrix with diagonal elements given by $\{\theta_i\}_{i=1}^d$. Further details of the derivation can be found in [30].

Later on in the chapter, we use the derived geodesic form to construct a measure that quantifies the distance between samples of different domains.

2.3 Domain-adaptive Similarity Function on the Grassmann Manifold

Let $X^s \in \mathbb{R}^{n \times m_s}$ and $X^t \in \mathbb{R}^{n \times m_t}$ denote the feature representation of m_s and m_t samples in source and target domains respectively where each column $\mathbf{x}_i \in \mathbb{R}^n$ denotes a sample and n is the feature dimension.

In [33], Gopalan et al. propose an approach which performs cross-domain class analysis using intermediate subspace along the geodesic on the Grassmann manifold. Specifically, they first apply the principle component analysis (PCA) on X^s and X^t respectively, which generates two d-dimensional subspaces denoted by matrices $S_0, S_1 \in \mathbb{R}^{n \times d}$. The geodesic path $\bar{\psi}(t)$ from S_0 to S_1 is then given by Eq. (2.3). Since each point on the geodesic is a subspace, the intermediate subspaces can be obtained by sampling the geodesic $\bar{\psi}(t)$ at different time points t_i. Let $\hat{S} = \{S_t\}_{t=t_1}^{t_k}$ denote the collection of the k sampled intermediate subspaces, where $0 = t_1 \leq ... \leq t_k = 1$. They then project each sample from both domains onto k subspaces in $\hat{S}$ and concatenate all the k projections to form a long vector of size $d \times k$. A discriminative classifier is then trained to classify samples of unknown labels based on the long vector representation using the samples whose labels are known. Note that in the semi-supervised classification task, labels of some samples in the target domain are also known.

The sampling approach of [33] has two main disadvantages. First, it is not clear which sampling method should be used since different sampling methods result in

different intermediate subspace representations and the final classification recognition degrades if an inferior sampling method is used. Second, the number of sampled points is limited because a large number of sampled points along the geodesic results in a very high dimensional feature vector which increases computational complexity. In order to overcome the two disadvantages, we propose an alternative approach. Instead of sampling some points along the geodesic, we integrate the distance of data projections onto the subspaces along the geodesic. This yields a cross-domain distance metric which can be used for cross-domain class analysis. Our approach consists of the following three steps.

Calculate the Θ: Given S_0 and S_1, the matrix Q in 2.4 can be computed according to 2.2 and $Q^T S_1$ is given by $Q^T S_1 = \begin{bmatrix} S_0^T S_1 \\ S_{12} - S_{02} Z^T \end{bmatrix}$ where $Z \in \mathbb{R}^{d \times d}$ satisfies $Z(I_d - S_{01}^T) = (S_1^T S_0 - S_{11}^T)$. Since

$$\psi(1) = Q \begin{bmatrix} U_1 \Gamma(t) \\ -\tilde{U}_2 \Sigma(t) \end{bmatrix} = \bar{S}_1 = S_1 V_1, \tag{2.5}$$

we have

$$Q^T S_1 = \begin{bmatrix} U_1 \Gamma(1) V_1^T \\ -\tilde{U}_2 \Sigma(1) V_1^T \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & \tilde{U}_2 \end{bmatrix} \begin{bmatrix} \Gamma(1) \\ -\Sigma(1) \end{bmatrix} V_1^T \tag{2.6}$$

Note that $\tilde{U}_2$ and Θ can be obtained by computing the thin CS decomposition of $Q^T S_1$ [30].

Calculate geodesic $\bar{\psi}(t)$: With the matrix Θ and $\tilde{U}_2$, one can obtain $\Gamma(t)$ and $\Sigma(t)$ using their definitions. By substituting $\Gamma(t)$ and $\Sigma(t)$ in (2.4), we obtain

the geodesic starting from the source domain S_0 to the target domain S_1:

$$\bar{\psi}(t) = Q \begin{bmatrix} U_1\Gamma(t) \\ -\tilde{U}_2\Sigma(t) \end{bmatrix} \tag{2.7}$$

Calculate domain-invariant distances: For a given pair of examples $(\mathbf{x}_1, \mathbf{x}_2)$ where $\mathbf{x}_1$ and $\mathbf{x}_2$ come from the source and target domain respectively, we project them onto the subspace $\bar{\psi}(t)$ indexed by t on the geodesic to obtain $\tilde{\mathbf{x}}_1 = \bar{\psi}(t)^T\mathbf{x}_1$ and $\tilde{\mathbf{x}}_2 = \bar{\psi}(t)^T\mathbf{x}_2$. The final distance between $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ is calculated by integration given by

$$d(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = \int_0^1 (\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2)^T(\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_2)dt \tag{2.8}$$

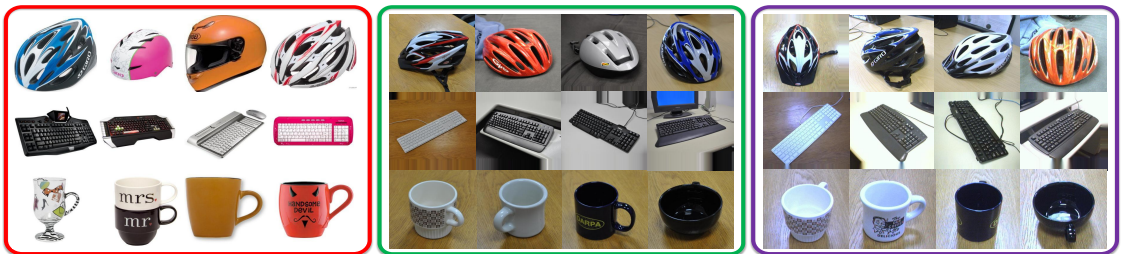
$$= (\mathbf{x}_1 - \mathbf{x}_2)^T \left(\int_0^1 \bar{\psi}(t)\bar{\psi}(t)^T dt \right) (\mathbf{x}_1 - \mathbf{x}_2) \tag{2.9}$$

$$= Q \begin{bmatrix} U_1 & 0 \\ 0 & \tilde{U}_2 \end{bmatrix} P \begin{bmatrix} U_1^T & 0 \\ 0 & \tilde{U}_2^T \end{bmatrix} Q^T. \tag{2.10}$$

where the matrix P can be easily determined using the subspace angles between S_0 and S_1. Note that (2.10) is an analytical form and can be computed in constant time. Finally, we calculate the distance between a test sample and all the labeled samples from both domains and use a nearest neighbor algorithm for classification.

2.4 Experiments

We experimented the proposed algorithm on the tasks of cross-domain object category recognition and face recognition under different imaging conditions.



(a) *amazon* (b) *dslr* (c) *webcam*

Figure 2.2: Sample images from the benchmark dataset [95]. We show several images from the object categories of bike helmet, keyboard, and mug in the three domains of *amazon*, *dslr*, and *webcam*. Domain shift in the dataset is mainly due to changes in image resolution, object pose, and scene lighting.

2.4.1 Cross-domain Object Recognition

We evaluated the proposed algorithm on the cross-domain object category classification task using the benchmark dataset [95], which contains images from 31 object categories. Depending on the acquisition condition, the dataset images are divided into three domains, namely *amazon*, *dslr* and *webcam*. The *amazon* domain includes an average of 90 product images for each category, downloaded from the Amazon's website. Both *dslr* and *webcam* domains have about 30 images per category; they are captured by a DSLR and a webcam respectively. We show some of the images in Fig 2.2. One can see that domain shift in the dataset is mainly due to changes in image resolution, object pose, background clutters, and scene lighting.

We used an image representation based on SURF [5] features similar to [95, 33].

Settings	source domain	target domain	[95] (asymm)	[95] (symm)	[33]	proposed
same-category	*webcam*	*dslr*	25	27	37	**66**
	dslr	*webcam*	30	31	36	**61**
	amazon	*webcam*	48	44	**57**	45
new-category	*webcam*	*dslr*	53	49	59	**66**

Table 2.1: Classification accuracies (in percentage) of our approach and state of the art [95, 33] under different settings. Asymm and symm are two variants proposed in [95].

Specifically, we extracted SURF features for all the images in the *amazon* domain and used a random subset of the features to learn a codebook of 800 codewords. The codebook was used to encode the SURF features and each image in the dataset was denoted by an 800-dimensional histogram. We further normalized the histograms so that it sums up to one. To obtain the final representation, the histograms of images in the same domain were further normalized to assure a zero mean and unit deviation for each dimension. Note that PCA is performed on the final representation.

There were two evaluation settings on the benchmark: same-category and new-category. In the same-category setting, there were labeled images for all the categories and for both domains. In the new-category setting, there were labeled images for all the categories in the source domain, but only half of the categories in the target domain contained labeled images.

The classification accuracies of different approaches are shown in Table 2.1.

The accuracies of each approach are averaged over 20 trials; each trial contained a random set of labeled images in both source and target domains. We observe that the proposed algorithm yields a better performance for two out of three tasks in the same-category setting. In addition, the proposed algorithm significantly improves the performance for the task in the new-category setting—by a margin of more than 10%. This shows the benefit of the integration-based approach which accumulated the distance along the geodesic over the previous approach [33]. However, we note that the proposed algorithm is not effective for the adaptation from *amazon* to *webcam*. We believe the reason is because the proposed algorithm only uses a simple nearest neighbor classification technique, while [95] and [33] were based on powerful machine learning algorithms of information theoretic metric learning [16] and partial least squares method [120].

2.4.2 Face recognition across blur and illuminations

We conducted face recognition experiments using the CMU-PIE dataset [100]. This dataset consists of images from 68 subjects captured under 21 different illumination conditions. We randomly selected 11 illumination conditions. All the images captured under these 11 conditions constituted the source domain data, while the remaining ones formed the target domain data. The images in the source domain were labeled, but not those in the target domain.

We synthesized domain shifts by applying two different types of blur kernels to the target domain data: 1) the Gaussian blur kernel, and 2) the motion blur kernel.

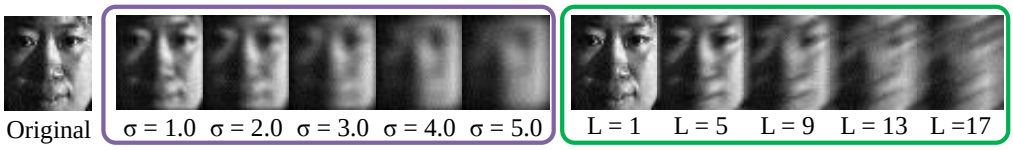


Figure 2.3: Target image samples. We illustrate several synthesized target images under different Gaussian blur and motion blur. The variable σ denotes the standard deviation, while L denotes the motion speed.

Gaussian blur	σ=1.0	σ=1.5	σ=2.0	σ=2.5	σ=3.0	σ=3.5	σ=4.0	σ=4.5	σ=5.0
[33]	93.8	86.8	86.3	70.9	57.5	43.7	28.4	21.5	17.7
Proposed	**94.9**	**88.7**	**88.2**	**74.6**	**62.7**	**47.4**	**31.9**	**24.3**	**19.4**

Table 2.2: Comparison of recognition accuracy under different Gaussian blur. We vary the standard deviation of the Gaussian blur from 1 to 5 and compare our recognition performance with [33].

Moreover, we gradually increased the kernel sizes to synthesize different degrees of domain shifts. For the Gaussian blur, we varied the standard deviation from 1 to 5. For the motion blur, we varied the motion speed, from 1 to 17 pixels. (The motion angle was set to 30 degrees.) Some of the target images were visualized in Fig. 2.3.

In Table 2.2 and 2.3, we compare the proposed algorithm to [33] (without applying the partial least square analysis) for the Gaussian and motion blurs respectively. It can be seen that the recognition accuracy of both methods decreases as the domain shift increases. However, the proposed algorithm consistently yielded a better performance than [33].

Motion blur	L=1	L=3	L=5	L=7	L=9	L=11	L=13	L=15	L=17
[33]	95.0	93.4	90.4	86.0	77.5	65.2	53.2	43.4	36.8
Proposed	**95.2**	**94.0**	**92.1**	**88.2**	**82.8**	**70.3**	**58.8**	**53.8**	**42.7**

Table 2.3: Comparison of recognition accuracy under different motion blur. We vary the motion speed from 1 to 17 pixels per sensor integration time and compare our recognition performance with [33].

2.5 Summary

We presented a cross-domain classification approach based on integrating the distance between data projections on the subspaces along the geodesic on a Grassmann manifold. We showed that the integration-based approach yields a better performance as compared to the previous approach that only samples few intermediate subspaces along the geodesic. In future, we plan to extend proposed approach by incorporating powerful machine learning methods.

Chapter 3: Transferable Dictionary Learning for Action Recognition

3.1 Related Work

Human action recognition has been receiving significant attention in computer vision over the past decades. The interest in action recognition is motivated by many real-world applications, such as large video archives, video search and editing, human computer interaction, autonomous vehicles and video surveillance. The task of human action recognition is to automatically analyze and recognize the action category from an unknown video. However, action recognition is challenging due to the large variations in action videos as shown in [68, 46]. For example, different subjects who perform the same action may have different expression, posture, clothing and motion rate; different environments in which the action takes place may result in different viewpoints, background, camera motions, lighting conditions and occlusions. Therefore, developing methods for action recognition that can generalize over all variations within one class and distinguish between actions of different classes becomes a major challenge.

In order to accurately recognize human actions, various approaches focus on developing robust and discriminative features from image sequences. These feature representations can be divided into two categories: global and local representa-

tions. Global representations encode the visual observations as a whole and are obtained from silhouettes, edges, trajectories and optical flow. For example, [9, 116] introduced a binary motion energy image by aggregating differences of silhouettes between subsequence frames of an actions. [118, 13] aligned and combined silhouettes from multiple cameras to obtain a new feature representation by using motion history volumes. Instead of using silhouette shape, [20, 4] extracted spatio-temporal motion patterns from the optical flow for human actio recognition. The methods presented in [132, 7] formed a 3D spatio-temporal volume by stacking silhouettes over a given sequence to extract local descriptors. Local representations describe the observation as a collection of local descriptors extracted from densely sampled patches or around space-time interest points. For example, [54, 17] used the Harris corner detector to detect space-time interest points and derive local descriptors. [119, 47, 97] extended 2D SURF features [5], HOG features, SIFT features[73] to 3D respectively.[56, 55] bin histograms of oriented gradients and flow extracted at interest points into a spatio-temporal grid. [97, 64, 67] exploited correlations between local descriptors for selection to construct higher-level descriptors.

These approaches are effective for recognizing actions taken from similar viewpoints, but they perform poorly when viewpoints vary significantly. Extensive experiments in [68, 141, 138] have shown that failing to handle feature variations caused by viewpoints may yield inferior results. This is because the same action looks quite different from different viewpoints. Thus action models learned from one view become less discriminative for recognizing actions in a much different view.

Many view-invariant approaches that use 2D image data acquired by multiple

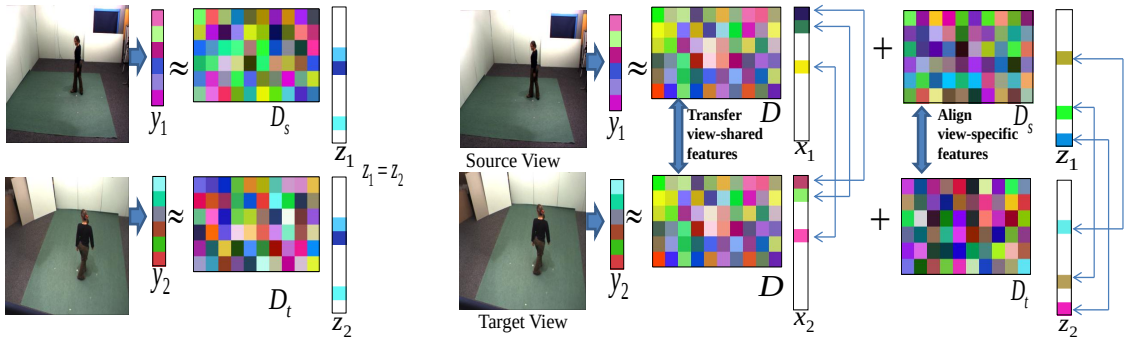
cameras have also been proposed. [89, 82, 81] proposed view-invariant representations based on view-invariant canonical body poses and trajectories in 2D invariance space. [46, 45] captured the structure of temporal similarities and dissimilarities within an action sequence using a Self-Similarity Matrix. [110] proposed a view-invariant matching method based on epipolar geometry between actor silhouettes without tracking and explicit point correspondences. [59] learned two view-specific transformations for source and target views, and then generated a sequence of linear transformations of action descriptors as the virtual views to connect two views. [58] proposed the Hankel matrix of a short tracklet which is a view-invariant feature to recognize actions across different viewpoints.

Another fruitful line of work for cross-view action recognition concentrates on using the 3D image data. The method introduced in [116] employed three dimensional occupancy grids built from multi-view points to model actions. [124] developed a 4D view-invariant action feature extraction to encode the shape and motion information of actors observed from multiple views. These approaches lead to computationally intense algorithms because they need to find the best match between a 3D model and a 2D observation over a large model parameter space. [117] developed a robust and view-invariant hierarchical classification method based on 3D HOG to represent a test sequence.

Recently, several transfer learning techniques have been proposed for cross-view action recognition [24, 68, 36, 113]. Specifically, [24] proposed to generate the same split-based features for correspondence video frames from both source and target views. It is computationally expensive because it requires the construction

of feature-to-feature correspondence at the frame-level and learning an additional mapping from original features to the split-based features. [68] used a bipartite graph to model the relationship between two view-dependent codebooks. [113] proposed a statistical translation framework (STF) to estimate the transfer probabilities of visual words in different views. Even though approaches in [68, 113] exploit the codebook-to-codebook correspondence between the two views, they can not guarantee that videos taken at different views of shared actions will have similar features. [36] used canonical correlation analysis to derive a correlation subspace as a joint representation from different bag-of-words models at different views and incorporate a corresponding correlation regularizer into the formulation of a support vector machine.

We propose to transfer sparse feature representations of videos across views for cross-view action recognition. Specifically, we make use of dictionary learning methods to exploit the video-to-video correspondence by encouraging a set of *correspondence* videos taken from different views of the same action to have the same or similar sparse representations. Here we present two different dictionary learning methods corresponding to different alignment of sparse features of correspondence videos in different views. In the first method as shown in Figure 3.1(a), we encourage the sparse representations of correspondence videos of the same action to be the same. In order to achieve this goal, we learn a set of view-specific dictionaries to represent videos from the corresponding view. Moreover, we encourage videos across views of the same action to have the same sparse representation when encoding using the corresponding view-specific dictionary. This procedure enables the



(a) Restricted transferable DL (b) Relaxed transferable DL

Figure 3.1: Restricted transferable dictionary learning(DL) versus relaxed transferable dictionary learning(DL). (a) Restricted transferable dictionary learning: We learn two view-specific dictionaries D^s and D^t corresponding to source and target views respectively. A pair of videos taken at the same time of the same class is denoted as y_1 and y_2. The sparse representations of y_1 and y_2 when encoded using the corresponding view-specific dictionaries are equally the same. (b) Relaxed transferable dictionary learning: We jointly learn two view-specific dictionaries D^s and D^t and a common dictionary D. Each video in each view is represented by both the common dictionary and corresponding view-specific dictionary. The sparse representations of y_1 and y_2 share the same sparsity patterns (selecting the same items) instead of being equally the same.

transfer of the sparse representations across views. However, the assumption in our first method that sparse representation of videos from different views of the same action should be equal may be too strong to flexibly model the relationship between different views.

In order to overcome this drawback, our second approach relaxes this assump-

tion by encouraging correspondence videos to have similar sparse representations as shown in Figure 3.1(b). Meanwhile, we also learn a common dictionary shared by different views to model view-shared features. Both common dictionary and the corresponding view-specific dictionary are used to represent videos in each view. Instead of transferring the split-features as in [6], we transfer the indices of the non-zero elements (i.e., the indices of selected dictionary items) in sparse codes of videos from the source view to sparse codes of the corresponding videos from the target view. In other words, we not only use the same subset of dictionary items from the common dictionary to represent view-shared features in correspondence videos from different views, but also use the same subset of dictionary items from different view-specific dictionaries to represent view-specific features. In this way, videos across different views of the same action tend to have similar sparse representations. Note that our approach enforces the common dictionary to be incoherent with view-specific dictionaries, the incoherence between the common dictionary and view-specific dictionaries enables our approach to drive the shared pattern to the common dictionary and focus on exploiting the discriminative correspondence videos taken from different views of the same action using a more flexible method.

Furthermore, actions are categorized into two types: *shared* actions observed in both training and test views and test actions that are only observed in the training view. In addition, we consider two scenarios for the shared actions: (1) shared actions in both views are unlabeled. (2) shared actions in both views are labeled. These two scenarios are referred to as unsupervised and supervised settings, respectively, in subsequent discussions. Note that under both settings, only the set

of videos taken from different views of the shared actions are used for dictionary learning. This means that the dictionaries will not be affected by videos of orphan actions.

This chapter is organized as follows: Section 2 briefly reviews sparse coding and dictionary learning. Sections 3 and 4 present the restricted and relaxed dictionary learning frameworks for cross-view action recognition respectively. Section 5 describes the optimization procedure of the proposed approaches. Section 6 provides experimental results and analysis on three public multi-view action datasets. Section 6 concludes the chapter.

3.2 Sparse Coding and Dictionary Learning

In this section, we give a brief review of sparse coding and the K-SVD algorithm [1] for learning an over-complete dictionary.

Let $Y = [y_1, ..., y_N] \in \mathbb{R}^{n \times N}$ be a set of N input signals in a n-dimensional feature space. Assuming a dictionary D of size K is given, the sparse representations $X = [x_1, ..., x_N] \in \mathbb{R}^{K \times N}$ for Y are obtained by solving:

$$X = arg \min_{X} ||Y - DX||_F^2 \quad \text{s.t.} \quad \forall i, ||x_i||_0 \leq s, \tag{3.1}$$

where $||Y - DX||_F^2$ denotes the reconstruction error and $||x_i||_0 \leq s$ is the sparsity constraint. The sparsity constraint requires that each signal has s or fewer items in its decomposition. The orthogonal matching pursuit (OMP) algorithm [108] can then be used to solve (3.1).

The performance of sparse representation depends critically on D. The K-

SVD [1] is well known for efficiently learning an over-complete dictionary from a set of training signals. It solves the following optimization problem:

$$(D, X) = arg \min_{D,X} ||Y - DX||_2^2 \quad \text{s.t.} \quad \forall i, ||x_i||_0 \leq s \tag{3.2}$$

where $D = [d_1, ..., d_K] \in \mathbb{R}^{n \times K}$ is the learned dictionary, and $X = [x_1, ..., x_N]$ are the sparse representations of Y. K-SVD is an iterative method that alternates between sparse coding of the signals based on the current dictionary and a process of updating the dictionary atoms to better fit the data. Later, we will formulate the problem of learning transferable dictionaries as an optimization problem which can be efficiently solved using the K-SVD algorithm.

3.3 Restricted Transferable Dictionary Learning

In this section, we present the restricted transferable dictionary learning (RSTDL) for cross-view action recognition. In this method, we learn a set of view-specific dictionaries such that the sparse representations of correspondence videos of the shared actions across views are the same. We further consider two settings: *unsupervised* setting and *supervised* setting for learning view-specific dictionaries.

3.3.1 Unsupervised Setting

In the unsupervised setting where labels of shared actions are not available, our goal is to transfer orphan action models from the source views to the target view. In other words, we want to learn an action model for orphan actions in the source views and test it in the target view. We achieve this goal by making use

of correspondence between a set of *correspondence* videos of the shared unlabeled actions taken from different views. Let $Y^v = [y_1^v, ..., y_N^v] \in \mathbf{R}^{d \times N}$ denote the d-dimensional feature representations of N videos of the shared unlabeled actions taken in the v-th view. $Y_i = [y_i^1, ..., y_i^V]$ are V action videos of the shared action y_i taken from V views, which are referred to as *correspondence* videos. For each view, we learn a view-specific dictionary $D^v \in \mathbf{R}^{d \times J^v}$ to model and align the view-specific features. The objective function for learning dictionaries under the unsupervised setting is as follows:

$$arg \min_{D^v, X} \sum_{v=1}^{V} ||Y^v - D^v X||_2^2 \quad \text{s.t.} \quad \forall i, ||x_i||_0 \leq s. \tag{3.3}$$

where $X = [x_1, ..., x_N]$ are the joint sparse representations for y_i across V views where $i = 1...N$, and s is the sparsity threshold.

3.3.2 Supervised Learning

In the supervised setting where the action categories of shared action videos are available in both views, we will leverage this category information to learn discriminative transferrable dictionaries. The key idea is to partition the total dictionary items into disjoint subsets such that each subset is responsible for representing videos of one action. Specifically, we represent videos of the same action by the same subset of dictionary items. For videos of different action classes, we represent them using disjoint subsets of dictionary items. This results in an explicit correspondence between dictionary items and the labels. The intuition behind this idea is that action videos from the same class tend to have same features and each action video

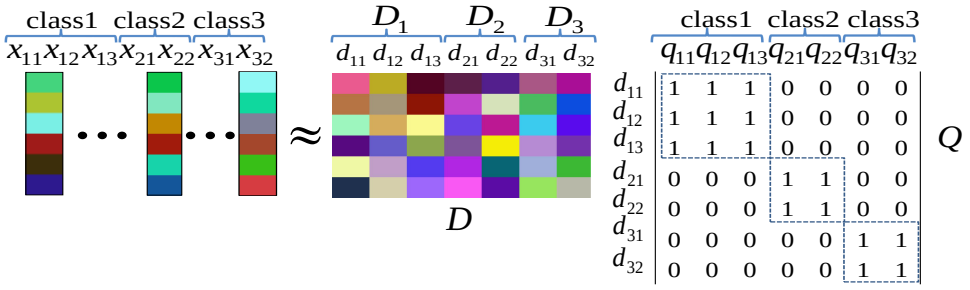


Figure 3.2: An example of the *ideal* sparse codes matrices Q for classification task. Given seven videos (on the leftmost) coming from three classes in the v-th views, we learn a view-specific dictionary for this view. The dictionary $D^v = [d_1^v, d_2^v, d_3^v, d_4^v, d_5^v, d_6^v, d_7^v]$ and $Y^v = [y_1^v, y_2^v, y_3^v, y_4^v, y_5^v, y_6^b, y_7^v]$, where $y_1^v, y_2^v, y_3^v, d_1^v, d_2^v, d_3^v$ are from class 1, $y_4^v, y_5^v, d_4^v, d_5^v$ are from class 2 and $y_6^b, y_7^v, d_6^v, d_7^v$ are from class 3. The defined Q are a block diagonal matrix, where each column corresponds to a discriminative sparse code for an input video.

could be well represented by other videos from the same class. On the contrary, videos from different classes tend to have different features and thus should be well represented by disjoint subsets of other videos.

In order to achieve the above goal, we incorporate a label consistent regularization term introduced in [43] to the objective function in (3.3). Now the objective function for learning dictionaries under the supervised settting is given by:

$$\begin{aligned} arg \min_{D^v, X, A} \sum_{v=1}^{V} ||Y^v - D^v X||_2^2 + \alpha ||Q - AX|| \\ \text{s.t.} \quad \forall i, ||x_i||_0 \leq s. \end{aligned} \tag{3.4}$$

where α controls the tradeoff between the reconstruction error and label consistent regularization. The matrix $Q = [q_1, ..., q_N] \in \mathbb{R}^{K \times N}$ are called the ideal "discriminative" sparse codes of shared action videos in both views. Thus the column vector

$q_i = [q_i^1, ..., q_i^K] = [0...1, 1, ..0] \in \mathbb{R}^K$ in Q is the discriminative sparse code of correspondence videos from the shared action $y_i^v, v = 1, ..., N$. Moreover, the non-zeros values of q_i occur at those indices where the correspondence videos of the shared action y_i^v and the dictionary items d_k share the same label. Figure 3.2 gives an example of the ideal sparse codes matrices Q.

Matrix A represents a linear transformation which transforms the original sparse code X to be most discriminative in sparse feature space $\mathbb{R}^K$. The term $||Q - AX||_F^2$ denotes the discriminative sparse code error, which enforces the sparse codes X to be more like the discriminative sparse codes Q after a linear transformation. This term not only forces videos from the same class to have very similar sparse representations, but also regularizes videos from different classes to have very different sparse representations. Therefore, the learned view-specific dictionaries are discriminative which may result in good classification even using a k-NN classifier.

3.4 Relaxed Transferrable Dictionary Learning

In this section, we present the relaxed transferable dictionary learning (RLTDL) for cross-view action recognition. Note that RSTDL assumed that videos taken at the same time of the same action across views should be strictly equal. The second approach RLTDL, relaxes this assumption, leading to a more flexible model to represent the relationship between views. Moreover, we not only learn a set of view-specific dictionaries, bust also learn a common dictionary shared by different views. The common dictionary models the view-shared features while the view-

specific dictionaries model and align view-specific features across views. We learn these dictionaries by using the same subset of dictionary atoms to represent the correspondence videos of the same action. Therefore, videos across different views of the same action tend to have similar sparse representations. Similarly, we further consider two settings: *unsupervised* and *supervised* settings for learning both common and view-specific dictionaries.

3.4.1 Unsupervised Setting

In the unsupervised setting the goal is to find view-invariant feature representations by making use of correspondence between videos of the shared actions taken from different views. On the one hand, we would like to learn a common dictionary $D \in \mathbf{R}^{d \times J}$ with a size of J shared by different views to represent videos from all views. On the other hand, for each view, we learn a view-specific dictionary $D^v \in \mathbf{R}^{d \times J^v}$ to model and algin the features in the v-th view. The objective function for learning both common and view-specific dictionaries in the unsupervised setting is formulated as follows:

$$\begin{aligned} arg \min_{D, D^v} \sum_{i=1}^{N} \{ \sum_{v=1}^{V} \{ ||y_i^v - Dx_i^v||_2^2 + ||y_i^v - Dx_i^v - D^v z_i^v||_2^2 \} \\ + \lambda ||X_i||_{2,1} + \lambda ||Z_i||_{2,1} \} + \eta \sum_{v=1}^{V} ||D^T D^v||_F^2 \end{aligned} \tag{3.5}$$

where $X_i = [x_i^1, ..., x_i^V], Z_i = [z_i^1, ..., z_i^V]$ are the joint sparse representations for $y_i, i = 1, ..., N$ across V views. This objective function consists of the following five terms:

1. The first two terms are the reconstruction errors of videos from different views

using D only or using both D and D^v. The minimization of the first reconstruction error enables D to encode view-shared features as much as possible while the minimization of the second reconstruction error enables D^v to encode and align view-specific features that can not be modeled by D.

2. The third and fourth terms denote sparse representations via $L_{2,1}$-norm regularization using D and D^v respectively. The $L_{2,1}$-norm minimization for X_i and Z_i can make the entries in each row of the two matrices to be all zeros or non-zeros at the same time. This means that we not only encourage the use of the same subset of dictionary items in D to represent the correspondence videos from different views, but also encourage the use of dictionary items from D^v with the same index of selected dictionary items to further reduce the reconstruction error of videos in each view. Therefore, the testing videos taken from different views of the same action will be encouraged to have similar sparse representations when encoded using the learned D and D^v.

3. The last term regularizes the common dictionary to be incoherent to the view-specific dictionaries. The incoherence between D and D^v enables the proposed approach to separately exploit the discriminative information encoded in the view-specific features and view-shared features.

In addition, the parameters λ and η control the relative contribution of $L_{2,1}$ norm regularization and the incoherence regularization respectively.

3.4.2 Supervised Learning

In the supervised setting where the action categories of correspondence videos are available, we can learn discriminative common dictionary and discriminative views-specific dictionaries by leveraging the category information. Similarly, we partition the dictionary items in each dictionary into disjoint subsets and associate each subset with one specific class label. For videos from action class k, we aim to represent them using the same subset of dictionary items associated with class k. For videos from different classes, we represent them using disjoint subsets of dictionary items.

Assume there are K shared action classes, and $D = [D_1, ..., D_K]$ is the common dictionary where $D_k \in \mathbf{R}^{d \times J_k}, \sum_{k=1}^{K} J_k = J$. Let $D^v = [D_1^v, ..., D_K^v]$ be the view-specific dictionary where $D_k^v \in \mathbf{R}^{d \times J_k^v}$, $\sum_{k=1}^{K} J_k^v = J^v$. The objective function for learning both the common dictionary and view-specific dictionaries under the supervised setting is given as follows:

$$\begin{aligned} arg \min_{D,D^v,A,B} \sum_{i=1}^{N} \{ \sum_{v=1}^{V} \{ ||y_i^v - Dx_i^v||_2^2 + ||y_i^v - Dx_i^v - D^v z_i^v||_2^2 \\ + \alpha ||q_i - Ax_i^v||_2^2 + \alpha ||q_i^v - Bz_i^v||_2^2 \} + \lambda ||X_i||_{2,1} \\ + \lambda ||Z_i||_{2,1} \} + \eta \sum_{v=1}^{V} ||D^T D^v||_F^2 \end{aligned} \tag{3.6}$$

where $q_i = [q_{i_1}, ..., q_{i_K}]^T \in \mathbf{R}^{J \times 1}$ and $q_i^v = [q_{i_1}^v, ..., q_{i_K}^v]^T \in \mathbf{R}^{J^v \times 1}$ are called 'discriminative' sparse coefficients associated with D and D^v respectively. When a video y_i^v is from class k in the v-th view, then q_{i_k} and $q_{i_k}^v$ are ones and other entries in q_i and q_i^v are zeros. $A \in \mathbf{R}^{J \times J}$ and $B \in \mathbf{R}^{J^v \times J^v}$ are called transformation matrices

which transform x_i^v and z_i^v to approximate q_i and q_i^v respectively. The discriminative sparse-code error terms $||q_i - Ax_i^v||_2^2$ and $||q_i^v - Bz_i^v||_2^2$ encourage the dictionary items with class k to be selected to reconstruct those videos from class k. Note that the $L_{2,1}$-norm regularization only regularize the relationship between the sparse codes of correspondence videos, but can not regularize the relationship between the sparse codes of videos from the same action class in each view. The integration of discriminative sparse code error term in the objective function can address this issue. In other words, the proposed approach not only encourages the videos taken from different views of the same action to have similar sparse representations, but also encourages videos from the same class in each view to have similar sparse representations.

3.5 Optimization

In the section, we describe optimization procedure for RSTDL and RLTDL approaches under both unsupervised and supervised settings.

3.5.1 Optimization of Restricted Transferable Dictionary Learning

The objective functions in (3.4) and (3.6) under the supervised setting reduce to the objective function in (3.3) and (3.5) under the unsupervised setting when $\alpha = 0$. Therefore, the optimization procedures of these two objective functions employ a very similar procedure. Here we only discuss the optimization of procedure for the objective function in 3.4.

We use the efficient K-SVD algorithm to find the optimal solution for all parameters simultaneously. Since we have the same number of shared action videos across V views, we rewrite the objective function in 3.4 as follows:

$$< D^v, X >= arg \min_{D^v, X} || \begin{bmatrix} Y^1 \\ Y^2 \\ ... \\ Y^V \\ \sqrt{\alpha}Q \end{bmatrix} - \begin{bmatrix} D^1 \\ D^2 \\ ... \\ D^V \\ \sqrt{\alpha}A \end{bmatrix} X||_2^2 \quad \text{s.t.} \quad \forall i, ||x_i||_0 \leq s. \tag{3.7}$$

Let $Y = [Y^{1T}, Y^{2T}, ..., Y^{VT}, \sqrt{\alpha}Q^T]^T$, and $D = [D^{1T}, D^{2T}, ..., D^{VT}, \sqrt{\alpha}A^T]^T$, each column of D is further normalized to have a L_2 norm of 1. The optimization of (3.7) is equivalent to solving the following problem:

$$< D, X > = arg \min_{D,X} ||Y - DX||_F^2 \quad \text{s.t.} \quad \forall i, ||x_i||_0 \leq s. \tag{3.8}$$

Since this is exactly the problem as shown in [1], we follow K-SVD to find the optimal solution for all parameters simultaneously. For the initialization of A, we employ the multivariate ridge regression model with the quadratic loss and L_2 norm regularization as follows:

$$A = arg \min_{A} ||Q - AX^0|| + \lambda_2 ||A||_2^2. \tag{3.9}$$

This yields the following solution:

$$A = QX^{0T}(XX^T + \lambda_2 I)^{-1}. \tag{3.10}$$

Algorithm 1 summarizes the RSTDL approach.

Algorithm 1: Restricted Transferable Dictionary Learning

1: **Input:** $Y^v = [Y_1^v, ..., Y_K^v], Q, \alpha, s, v = 1, ..., V$

2: Set $\alpha = 0$ in (3.7) and solve it to obtain the initialization of $D^v, v = 1, ..., V$

3: Initialize A using (3.10)

4: Reset α to the original given value and compute D by solving (3.7) using K-SVD algorithm

5: Decompose D into $D^v, v = 1, ..., V$ and A

6: Normalize each column in D^v

7: **Output:** $D^v, v = 1, ..., V$

3.5.2 Optimization of Relaxed Transferable Dictionary Learning

In the RLTDL approach, we only describe the optimization of the objective function in (3.6) while the optimization of (3.5) utilizes the similar procedure except that A and B components are excluded. This optimization problem is divided into three subproblems: (1) computing sparse codes with fixed D^v, D and A, B; (2) updating D^v, D with fixed sparse codes and A, B; (3) updating A, B with fixed D^v, D and sparse codes.

3.5.2.1 Computing Sparse Codes

Given fixed D^v, D and A, B, we solve the sparse coding problem of the correspondence videos set by set and (3.6) is reduced to:

$$\sum_{v=1}^{V} \{||y_i^v - Dx_i^v||_2^2 + ||y_i^v - Dx_i^v - D^v z_i^v||_2^2 + \alpha||q_i - Ax_i^v||_2^2$$

$$+ \alpha||q_i^v - Bz_i^v||_2^2\} + \lambda||X_i||_{2,1} + \lambda||Z_i||_{2,1}\}. \tag{3.11}$$

We rewrite (3.11) as follows:

$$\sum_{v=1}^{V} ||\tilde{y}_i^v - \tilde{D}^v \tilde{z}_i^v||_2^2 + \lambda||\tilde{Z}_i||_{2,1} \tag{3.12}$$

where $\tilde{y}_i^v = \begin{bmatrix} y_i^v \\ y_i^v \\ \sqrt{\alpha}q_i \\ \sqrt{\alpha}q_i^v \end{bmatrix}, \tilde{D}^v = \begin{bmatrix} D & O_1 \\ D & D^v \\ \sqrt{\alpha}A & O_2 \\ O_3 & \sqrt{\alpha}B \end{bmatrix}, \tilde{z}_i^v = \begin{bmatrix} x_i^v \\ z_i^v \end{bmatrix}, \tilde{Z}_i = [\tilde{z}_i^1, ..., \tilde{z}_i^V]$ and $O_1 \in \mathbf{R}^{d \times J^v}, O_2 \in \mathbf{R}^{J \times J^v}, O_3 \in \mathbf{R}^{J^v \times J}$ are matrices of all zeros. The minimization of (3.12) is known as a multi-task group lasso problem [65] where each view is treated as a task. We use the software SLEP in [65] for computing sparse codes.

3.5.2.2 Updating Dictionaries

Given fixed sparse codes and A, B, (3.6) is reduced to,

$$\sum_{i=1}^{N}\sum_{v=1}^{V}\{||y_i^v - Dx_i^v||_2^2 + ||y_i^v - Dx_i^v - D^v z_i^v||_2^2\} + \eta\sum_{v=1}^{V}||D^T D^v||_F^2 \tag{3.13}$$

We rewrite (3.13) as:

$$\sum_{v=1}^{V}\{||Y^v - DX^v||_F^2 + ||Y^v - DX^v - D^v Z^v||_F^2\} + \eta\sum_{v=1}^{V}||D^T D^v||_F^2 \tag{3.14}$$

where $Y^v = [y_1^v, ..., y_N^v], X^v = [x_1^v, ..., x_N^v], Z^v = [z_1^v, ..., z_N^v]$. Motivated by [48], we first fix D^v and then update $D = [d_1, ..., d_J]$ atom by atom, i.e. updating d_j while fixing other column atoms in D. Specifically, let $\hat{Y}^v = Y^v - \sum_{m \neq j} d_m x_{(m)}^v$ where

$x^v_{(m)}$ corresponds to the m-th row of X^v, we solve the following problem for updating d_j in D:

$$arg\min_{d_j} f(d_j) = \sum_{v=1}^{V}\{||\hat{Y}^v - d_j x^v_{(j)}||_F^2 + ||\hat{Y}^v - D^v Z^v - d_j x^v_{(j)}||_F^2 + \eta||d_j^T D^v||_F^2. \tag{3.15}$$

Let the first-order derivative of $f(d_j)$ with respect to d_j equal to zero, *i.e.* $\frac{\partial f(d_j)}{\partial d_j} = 0$, then we can update d_j as:

$$d_j = \frac{1}{2}\sum_{v=1}^{V}(||x^v_{(j)}||_2^2 I + \frac{\eta}{2} D^v D^{vT})^{-1}(2\hat{Y}^v - D^v Z^v)x^{vT}_{(j)}. \tag{3.16}$$

Now we fix D and update D^v atom by atom. Each item d^v_j in D^v is updated as :

$$d^v_j = \frac{1}{2}(||z^v_{(j)}||_2^2 I + \frac{\eta}{2} D D^T)^{-1}\bar{Y}^v z^{vT}_{(j)}. \tag{3.17}$$

where $\bar{Y}^v = Y^v - DX^v - \sum_{m\neq j} d^v_m z^v_{(m)}$.

3.5.2.3 Updating A, B

Given sparse codes and all the dictionaries, we employ the multivariate ridge regression model [87] to update A, B with the quadratic loss and l_2 norm regularization:

$$\min_A \sum_{i=1}^{N}\sum_{v=1}^{V} ||q_i - Ax^v_i||_2^2 + \lambda_1||A||_2^2$$
$$\min_B \sum_{i=1}^{N}\sum_{v=1}^{V} ||q^v_i - Bz^v_i||_2^2 + \lambda_2||B||_2^2$$

which yields the following solutions:

$$
\begin{aligned}
&A^* = Q\sum_{v=1}^{V} X^{vT}(\sum_{v=1}^{V} X^v X^{vT} + \lambda_1 I)^{-1},\\
&Q = [q_1, ..., q_N], X = [x_1, ..., x_N],\\
&B^* = \sum_{v=1}^{V} Q^v Z^{vT}(\sum_{v=1}^{V} Z^v Z^{vT} + \lambda_2 I)^{-1},\\
&Q^v = [q_1^v, ..., q_N^v], Z^v = [z_1^v, ..., z_N^v].
\end{aligned}
\tag{3.18}
$$

Algorithm 2 summarizes the RLTDL approach. The algorithm converged after a few iterations in our experiments.

3.6 Implementation Details

In this section, we provide the implementation details of our approaches. We used both spatio temporal interest point-based features [17] (STIP) and shape-flow features [107] in the experiments. In order to detect interest points for the STIP feature, we applied a 2D Gaussian smoothing filter to video along the spatial dimension, followed by a pair of 1D Gabor filters temporally. Then we detect up to 200 interest points at the local maximum response from each action video. We extract the ST volumes around the interest points and obtain a 100-dimensional gradient-based descriptor via PCA. Following [68], these interest points-based descriptors are further quantized into 1000 visual words by k-mean clustering and each action video is represented by a 1000-dimensional histogram.

The shape-flow features are based on histograms of the silhouette and of the optical flow inside the normalized bounding box. Specifically, each frame descriptor has three channels: horizontal optical flow, vertical optical flow and silhouette. In

Algorithm 2: Relaxed Transferable Dictionary Learning

Input: $Y^v = [Y_1^v, ..., Y_K^v], Q, Q^v, v = 1, ..., V, \lambda, \eta$

Initialize D **and** D^v

for $k = 1 \rightarrow K$ **do**

Initialize class-specific dictionary D_k in D by solving

$$D_k = arg\min_{D_k, \alpha_k} ||[Y_k^1 ... Y_k^V] - D_k \alpha_k||_F^2 + \lambda ||\alpha_k||_1$$

Initialize class-specific dictionary D_k^v in D^v by solving

$$D_k^v = arg\min_{D_k^v, \beta_k^v} ||Y_k^v - D_k^v \beta_k^v||_F^2 + \lambda ||\beta_k||_1$$

end for

repeat

Compute sparse codes x_i^v, z_i^v of a set of correspondence videos y_i^v by solving the multi-task group LASSO problem in (3.12) using the SLEP [65]

Update each atom d_j in D and d_j^v in D^v using (3.16) and (3.17)

Update transformation matrices A, B using (3.18)

until convergence or certain rounds

Output: $D = [D_1, ..., D_K], D^v = [D_1^v, ..., D_K^v]$

order to capture the motion context, the current frame descriptors are combined with a context descriptor extracted from neighboring frames. We learn a codebook of size 500 by k-means clustering on these shape-flow descriptors. Similarly, this codebook is used to encode shape-flow descriptors and each action video is represented by a 500-dimensional histogram. The interest point-based features capture rich local motion information while shape-flow features capture the global shape.

For the IXMAS dataset, we set the spatial and temporal scale parameters $\sigma = 2$ and $\tau = 1.5$ for interest points detection. The concatenation of both STIP and shape-flow feature descriptors forms a 1500-dimensional descriptor to represent an action video. For the WVU dataset, we set $\sigma = 2$ and $\tau = 2.5$ to detect interest points and each video is represented by only a 1000-dimensional STIP feature descriptor. For the MuHAVi dataset, we set $\sigma = 2, \tau = 1.5$ for interest points detection and each video is represented by only a 1000-dimensional STIP feature descriptor.

For a fair comparison [24, 68, 59], we use three evaluation modes for experiments: (1) *unsupervised correspondence* mode; (2) *supervised correspondence* mode ; (3) *partially labeled* mode. For the first two correspondence modes, we use the *leave-one-action-class-out* strategy for choosing the test action which means that each time we only consider one action class for testing in the target view. And all videos of the test action are excluded when learning the quantized visual words and constructing dictionaries. The only difference between the first and the second mode is whether the category labels of the correspondence videos are available or not. For the third mode, we follow [59] to consider a semi-supervised setting where a small portion of videos from the target view is labeled and no matched correspondence

videos exist. From this we want to show that the proposed approach can be applied to the domain adaptation problem.

Note that the test actions from the source and target views are not seen during dictionary learning whereas the test action can be seen in the source view for classifier training in the first two evaluation modes. On the contrary, the test action from different views can be seen during both dictionary learning and classifier training in the third mode. For all modes, we report the classification accuracy by averaging the results over different combinations of selecting test actions.

For the first mode, we generate sparse features using the dictionaries learned from RSTDL and RLTDL as follows:(1)RSTDL: Given the learned two view-specific dictionaries $\{D^1, D^2\}$ for the training and test views, we reconstruct the training and test videos over D^1 and D^2 respectively using the OMP algorithm to obtain the sparse features. (2)RLTDL: Given the learned common dictionary D and two view-specific dictionaries $\{D^1, D^2\}$, we use both D and D^1 to represent the training videos. Similarly, we encode test video over both D and D^2. Based on the sparse features, a k-NN classifier is used to classify test videos. The value of k ranges from 1 to 15 for three test data sets.

In addition, we set the sparsity factor $T = 20$, $T = 25$ and $T = 5$ for the IXMAS, WVU and MuHAVi datasets respectively. Throughout the experiments, the parameters $\alpha = 0.3, \eta = 1$ and λ varies from 0.1 to 5.

For the third mode, we use SRC method [121] to predict the label of y, *i.e.* $k^* = arg\min_k ||y - \hat{D}_k\beta_k||_2^2$ where $\hat{D}_k = [D_k \;\; D_k^t]$ and β_k is the associated sparse codes.

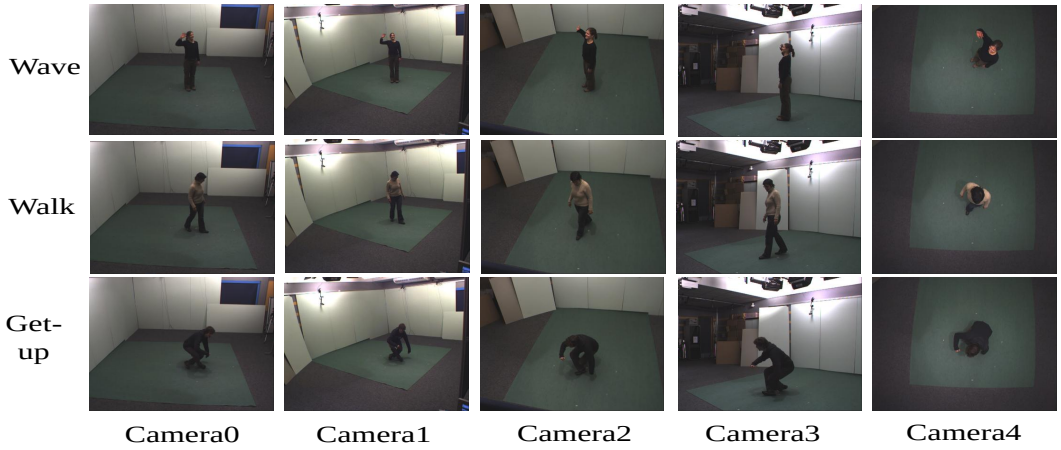


Figure 3.3: Exemplar frames from the IXMAS multi-view dataset. Each row shows one action viewed across different angles.

3.7 Experiments

We evaluated the proposed approaches for both cross-view and multi-view action recognition on three public multiview action data sets: IXMAS action dataset [116], WVU action dataset [88] and MuHAVi action dataset [101].

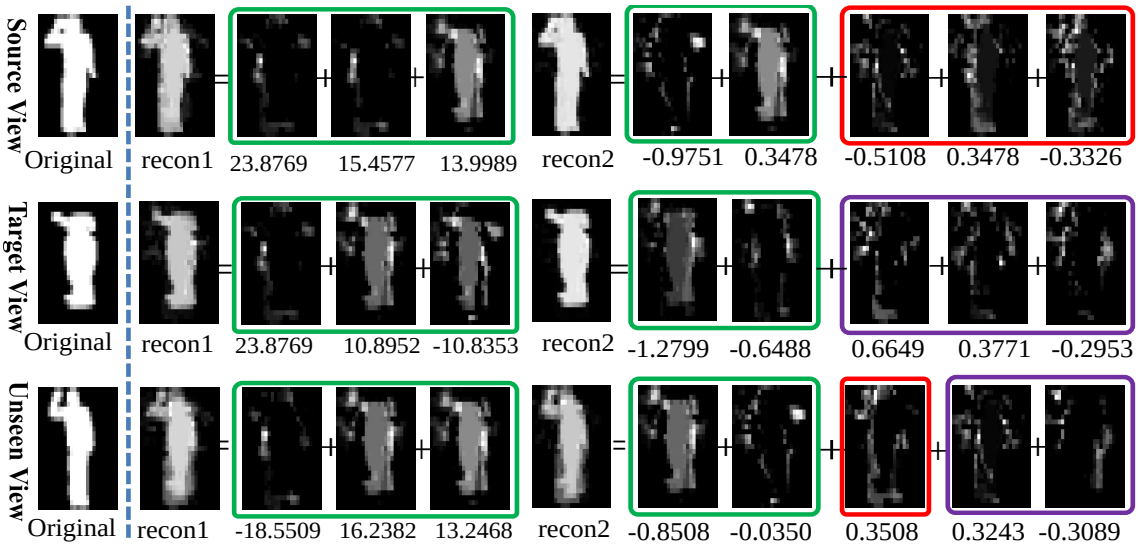
3.7.1 Evaluation on IXMAS action dataset

The IXMAS action dataset contains 11 daily life actions performed three times by ten actors taken from four side views and one top view. These actions are check-watch, cross-arms, scratch-head, sit-down, get-up, turn-around, walk, wave, punch, kick and pick-up. Figure 3.3 shows some example frames.

3.7.1.1 Benefits of the Separation of the Common and View-specific Dictionaries

In this section, we demonstrate the benefits of the separation of the common and view-specific dictionaries. For visualization purpose, two action classes "check-watch" and "waving" taken by Camera0 and Camera2 from the IXMAS dataset were selected to construct a simple cross-view dataset. We extract the shape descriptor [62] for each video frame and learn a common dictionary and two-view specific dictionaries using our approach. We then reconstruct a pair of frames taken from Camera0 and Camer2 views of the action "waving" using two methods. The first one is to use the common dictionary only to reconstruct the frame pair. The other one is use both the common dictionary and the view-specific dictionary for reconstruction. Figure 3.4(b) shows the original shape feature and the reconstructed shape features of two frames of action "waving" from two seen views and one unseen view using the aforementioned two methods. First, comparing the dictionary items in D and $\{D^s, D^t\}$, we see that some items in D mainly encode the body and body outline which are just shared by frames from the same action from two view while items in $\{D^s, D^t\}$ mainly encode different arm poses that reflects the class information in the two views. It demonstrates that the common dictionary has the ability to exploit view-shared features from different views. Second, it can be observed that better reconstruction is achieved by using both the common dictionary D and view-specific dictionaries. This is because the common dictionary may not reconstruct the more detailed view-specific features well such as arm poses. The

(a) Visualization of all dictionary items.



(b) Reconstruction of shape features of action "waving".

Figure 3.4: Illustration of the benefits of the common dictionary. (a) Visualization of all dictionary atoms in D (green color), D^s (red color) and D^t (purple color). (b) Columns $2 \sim 5$ show the reconstruction result using D only. Columns $6 \sim 11$ show the reconstruction result using $\{D, D^s\}$, $\{D, D^t\}$ and $\{D, D^s, D^t\}$ respectively. Only at most top-3 dictionary items are shown.

separation of the common dictionary enables the view-specific dictionaries to focus on exploiting and aligning view-specific features from different views. Third, from

the last row in Figure 3.4(b), we find that a good reconstruction of an action frame taken from the unseen view can be achieved by using the common dictionary only. It demonstrates that the common dictionary learned from two seen views has the capability to represent videos of the same action from an unseen view. Moreover, the two methods have nearly the same reconstruction performance for frames of the same action from the unseen view. This is because $\{D^s, D^t\}$ are learned by exploiting features that are specific for the two seen views. In addition, the separation of the common dictionary and view-specific dictionaries enables us to learn more compact view-specific dictionaries.

3.7.1.2 Cross-view Action Recognition

We first evaluate RSTDL and RLTDL approaches for cross-view action recognition under the first two different modes. We denote our proposed RSTDL and RLTDL under the unsupervised and supervised modes as un-RSTDL, un-RLTDL and su-RSTDL, su-RLTDL respectively.

Tables 3.1 displays recognition accuracies of cross-view action recognition for different combinations of training and test cameras under the unsupervised mode. We averaged the recognition accuracies over all classes. It can be seen that both un-RSTDL and un-RLTDL yield a much better performance for all 20 combinations of pairwise views. Moreover, the proposed un-RLTDL achieves more than 90% recognition accuracy for most combinations. Tables 3.2 shows recognition accuracies of cross-view action recognition under the supervised mode. The supervised method

%	C0	C1	C2	C3	C4
C0		(77.6, 79.9, 81.8, 96.7, **99.1**)	(69.4, 76.8, 88.1, **97.9**, 90.9)	(70.3, 76.8, 87.5, **97.6**, 88.7)	(44.8, 74.8, 81.4, 84.9, **95.5**)
C1	(77.3, 81.2, 87.5, 97.3, **97.8**)		(73.9, 75.8, 82.0, **96.4**, 91.2)	(67.3, 78.0, **92.3**, 89.7, 78.4,)	(43.9, 70.4, 74.2, 81.2, **88.4**)
C2	(66.1, 79.6, 85.3, 92.1, **99.4**)	(70.6, 76.6, 82.6, 89.7, **97.6**)		(63.6, 79.8, 82.6, **94.9**, 91.2)	(53.6, 72.8, 76.5, 89.1, **100.0**)
C3	(69.4, 73.0, 82.1, **97.0**, 87.6)	(70.0, 74.4, 81.5, 94.2, **98.2**)	(63.0, 66.9, 80.2, 96.7, **99.4**)		(44.2, 66.9, 70.0, 83.9, **95.4**)
C4	(39.1, 82.0, 78.8, 83.0, **87.3**)	(38.8, 68.3, 73.8, 70.6, **87.8**)	(51.8, 74.0, 77.7, 89.7, **92.1**)	(34.2, 71.1, 78.7, 83.7, **90.0**)	
Ave.	(63.0, 79.0, 83.4, 92.4, **93.0**)	(64.3, 74.7, 79.9, 87.8, **95.6**)	(64.5, 75.2, 82.0, **95.1**, 93.4)	(58.9, 76.4, 85.3, **91.2**, 87.1)	(46.6, 71.2, 75.5, 84.8, **95.1**)

Table 3.1: Cross-view action recognition accuracies of different approaches on the IXMAS dataset under unsupervised correspondence mode. Each row corresponds to a source (training) view and each column a target (test) view. The four accuracy numbers in the bracket are the average recognition accuracies of [46], [68], [59], un-RSTDL and un-RLTDL respectively.

not only outperforms other algorithms, but also improves the accuracies based on the unsupervised approach. This demonstrates that the dictionaries learned using labeled information across views are more discriminative.

For the partially labeled mode, we compare the proposed RLTDL with [58] and two types of SVMs used in [3]. [58] treated linear transformations of action descriptors as virtual views to connect the descriptors extracted from source view to those extracted from target view. The first type of SVM in [3] is AUGSVM, which creates a feature-augmented version of each individual feature as the new feature. The second one is MIXSVM which trains two SVM's on the source and target views and learns an optimal linear combination of them. Table 3.3 shows that the proposed approach outperforms other comparing approaches for most of source-target combinations. It is interesting to note that for the case where Camera4 is the

%	C0	C1	C2	C3	C4
C0		(79, **98.8**, 98.5)	(79, 99.1, **99.7**)	(68, 99.4, **99.7**)	(76, 92.7, **99.7**)
C1	(72, 98.8, **100.0**)		(74, **99.7**, 97.0)	(70, **92.7**, 89.7)	(66, 90.6, **100.0**)
C2	(71, **99.4**, 99.1)	(82, 96.4, **99.3**)		(76, 97.3, **100.0**)	(72, 95.5, **99.7**)
C3	(75, **98.2**, 90.0)	(75, 97.6, **99.7**)	(73, **99.7**, 98.2)		(76, 90.0, **96.4**)
C4	(80, 85.8, **99.7**)	(73, 81.5, **95.7**)	(73, 93.3, **100.0**)	(79, 83.9, **98.5**)	
Ave.	(74, 95.5, **97.2**)	(77, 93.6, **98.3**)	(76, 98.0, **98.7**)	(73, 93.3, **97.0**)	(72, 92.4, **98.9**)

Table 3.2: Cross-view action recognition accuracies of different approaches on the IXMAS dataset under supervised correspondence mode. Each row corresponds to a source (training) view and each column a target (test) view. The accuracy numbers in the bracket are the average recognition accuracies of [25] and our proposed su-RSTDL and su-RLTDL respectively.

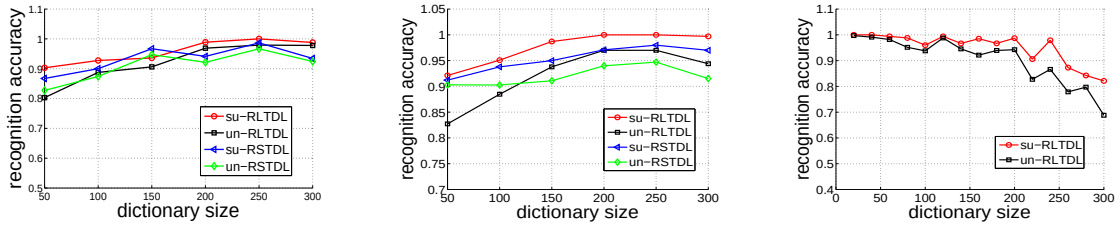
source or target view, the recognition accuracies of other approaches are a little lower than other combinations of pairwise views. This is because the Camera4 was set above the actors and different actions look very similar from the top view. However, our approach still achieves a very high recognition accuracy for these combinations, which further demonstrates the effectiveness of our approach.

We also evaluate the effect of dictionary size of the common dictionary D and view-specific dictionaries D^v on the proposed approaches. Figure 3.5 shows the performance of the proposed approaches on three pairs of source and target combinations with varying dictionary size. For Figure 3.5(a)(b), we fix the dictionary size of D to be 50, and vary the dictionary size of D^v from the range of

%	C0	C1	C2	C3	C4
C0		(42.8, 36.8, 63.6, **64.9**)	(45.2, 46.8, 60.0, **64.1**)	(47.2, 42.7, 61.2, **67.1**)	(30.5, 36.7, 52.6, **65.5**)
C1	(44.1, 39.4, 61.0, **63.6**)		(43.5, 51.8, **62.1**, 60.2)	(47.1, 45.8, 65.1, **66.7**)	(43.6, 40.2, 54.2, **66.8**)
C2	(53.7, 49.1, 63.2, **65.4**)	(50.5, 49.4, 62.4, **63.2**)		(53.5, 45.0, **71.7**, 67.1)	(39.1, 46.9, 58.2, **65.9**)
C3	(46.3, 39.3, 64.2, **65.4**)	(42.5, 42.5, **71.0**, 61.9)	(48.8, 51.2, 64.3, **65.4**)		(37.5, 38.9, 56.6, **61.6**)
C4	(37.0, 40.3, 50.0, **65.8**)	(35.0, 42.5, 59.7, **62.7**)	(44.4, 40.4, 60.7, **64.5**)	(37.2, 40.7, 61.1, **61.9**)	
Ave.	(45.3, 42.6, 59.6, **65.0**)	(42.7, 42.8, **64.2**, 63.2)	(45.4, 47.5, 61.9, **63.5**)	(46.2, 43.5, 64.8, **65.7**)	(37.6, 40.7, 55.4, **65.0**)

Table 3.3: Cross-view action recognition accuracies of different approaches on the IXMAS dataset under *partially* labeling mode. Each row corresponds to a source (training) view and each column a target (test) view. The accuracy numbers in the bracket are the average recognition accuracies of AUGSVM, MIXSVM from [3], [59], and RLTDL respectively.

$\{50, 100, 150, 200, 250, 300\}$. We observe that the performance of our approaches increases as the dictionary size of D^v increases. For Figure 3.5(c), we fix the dictionary size of D^v to be 300, and change the dictionary size of D from the range of $\{50, 100, 150, 250, 300\}$. It can be seen that our approaches achieve high recognition accuracies even using a very small size dictionary. However, when the dictionary size



(a) Camera0 v.s. Camera4 (b) Camera2 v.s. Camera3 (c) Camera0 v.s. Camera1

Figure 3.5: Performance on the IXMAS action dataset with varying dictionary size.

of D is too large, the redundancy in dictionaries will affect the sparse representation of test videos and the performance may decrease.

3.7.1.3 Multi-view Action Recognition

In this section, we evaluate our approaches for multi-view action recognition. We select one camera as a target view and use all other four cameras as source views to explore the benefits of combining multiple source views. Here we use the same classification scheme used for cross-view action recognition. Both D and the set of correspondence dictionaries D^v are learned by aligning the sparse representations of shared action videos across all views. Since videos from all views are aligned into a common view-invariant sparse feature space, we do not need to differentiate the training videos from each source view in this common view-invariant sparse feature space.

Table 3.4 shows the average accuracy of the proposed approach for the first two evaluation modes. Note that algorithms compared to are evaluated using the unsupervised correspondence mode. Both unsupervised and supervised approaches outperform other comparing approaches and achieve nearly perfect performance for all target views. Furthermore, [68, 141] and our unsupervised approach only use training videos from four source views to train a classifier while other approaches used all the training videos from all five views to train the classifier. Table 3.5 shows the average accuracy of different approaches using the *partially labeled* evaluation mode. The proposed approach outperforms [59] on four out of five target views.

%	C0	C1	C2	C3	C4	Avg
un-RLTDL	**97.0**	**99.7**	**97.2**	**98.0**	**97.3**	**97.8**
su-RLTDL	**99.7**	**99.7**	**98.8**	**99.4**	**99.1**	**99.3**
un-RSTDL	98.5	99.1	99.1	100	90.3	97.4
su-RSTDL	99.4	98.8	99.4	99.7	93.6	98.2
[68]	86.6	81.1	80.1	83.6	82.8	82.8
[46]	74.8	74.5	74.8	70.6	61.2	71.2
[67]	76.7	73.3	72.0	73.0	N/A	73.8
[117]	86.7	89.9	86.4	87.6	66.4	83.4

Table 3.4: Multi-view action recognition results using the unsupervised and supervised correspondence modes. Each column corresponds to one target view.

Overall, we accomplish comparable performance with [59] under the *partially labeled* mode.

3.7.2 Evaluation on the WVU action dataset

The WVU action dataset [88] is collected from a network of eight embedded color cameras. This multi-camera network provides completely overlapping coverage of a rectangular region from different view directions. This dataset has eleven action classes which includes nodding head, clapping, waving one hand, waving two hand, punching, jogging, jumping jack, kicking, picking, throwing, and bowling. Each action class has 47 action videos. Figure 3.6 shows exemplar frames of two action classes taken by eight cameras.

%	C0	C1	C2	C3	C4
RLTDL	**66.6**	**68.4**	**65.4**	67.2	**67.8**
[59]	62.0	65.5	64.5	**69.5**	57.9
AUGSVM	54.2	50.8	58.1	49.5	46.9
MIXSVM	46.4	44.2	52.3	47.7	44.7

Table 3.5: Multi-view action recognition results using the partially labeled mode. Each column corresponds to one target view.

Figure 3.6: Exemplar frames from the WVU action dataset. Each row shows one action viewed across different angles.

We evaluate our proposed approaches for cross-view action recognition on this dataset. We use the same strategy as that used in the IXMAS dataset to learn the dictionaries. We compare our method with **STF** [113], which exploits the relationship between visual words across views by estimating the word transfer probabilities. The recognition accuracies for cross-view action recognition under the unsupervised mode are summarized in Table 3.6. Compared with [113], the un-RSTDL approach achieves a highly comparable performance while the un-RLTDL approach yields a much better performance for a majority of combinations of pairwise views. This is because [113] can not guarantee that videos taken at different views of the same

%	C0	C1	C2	C3	C4	C5	C6	C7
C0		(92.5, 99.8, **100**)	(89.3, 99.8, **100**)	(90.2, 99.6, **100**)	(90.9, 99.8, **100**)	(88.2, 99.2, **99.6**)	(90.7, 99.4, 99.8)	(91.6, 99.8, **100**)
C1	(87.3, 99.6, **100**)		(86.8, 99.8, **100**)	(89.3, 99.8, **100**)	(84.3, 99.6, **99.8**)	(92.5, 97.3, **98.6**)	(86.6, 99.4, **99.6**)	(89.5, 99.8, **100**)
C2	(88.9, 88.9, **91.9**)	(90.7, **97.4**, 81.8)		(89.8, 90.7, 89.9)	(85.0, 89.4, **90.5**)	(90.5, 89.5, **91.1**)	(89.8, 90.1, **90.1**)	(**92.3**, 90.3, 89.7)
C3	(86.1, 72.7, **100**)	(92.3, 72.9, **99.6**)	(85.7, 72.7, **99.6**)		(86.1, 72.5, **98.8**)	(**90.5**, 71.6, 88.4)	(86.8, 72.7, **97.1**)	(91.6, 72.5, **99.8**)
C4	(91.1, 90.9, 98.2)	(87.7, **91.9**, 90.1)	(86.4, 93.6, **94.0**)	(92.7, 99.8, **98.6**)		(91.4, 92.1, **99.2**)	(86.6, 90.0, **90.0**)	(91.8,93.2, **94.8**)
C5	(90.0, 78.9, **93.0**)	(**92.0**, 88.9, 89.0)	(**90.0**, 80.3, 81.4)	(**90.0**,77.9, 89.7)	(**90.5**, 76.0, 90.1)		(89.3, 81.8, **82.6**)	(**90.2**, 83.5, 76.4)
C6	(**88.0**, 79.5, 81.4)	(**89.8**, 81.8, 74.2)	(89.8,84.1, **83.8**)	(**90.0**, 81.6, 81.2)	(**83.6**, 78.5,81.6)	(89.8, **90.3**, 81.6)		(**91.6**, 81.8, 82.2)
C7	(90.0, 90.7,**98.8**)	(**91.6**, 90.9, 91.3)	(88.4, 91.1, **97.3**)	(**92.0**, 90.9, 90.9)	(86.4, 91.7, **91.9**)	(90.2, **90.7**, 89.4)	(88.6, **98.8**, 96.5)	
Ave.	(88.8, 86.0, **94.7**)	(**90.9**, 89.1, 89.6)	(88.1, 88.8, **93.3**)	(90.6, 91.4, **93.0**)	(86.7, 86.8, **98.9**)	(90.4, 90.1, **92.6**)	(88.3, 90.4, **93.6**)	(91.2, 88.8, **91.8**)

Table 3.6: Cross-view action recognition accuracies of different approaches on the WVU dataset using unsupervised correspondence mode. Each row corresponds to a source (training) view and each column a target (test) view. The accuracy numbers in the bracket are the average recognition accuracies of [113], our proposed un-RSTDL and un-RLTDL approaches respectively.

action will have similar features, even though it estimated the transfer probabilities of visual words across views. However, the proposed approaches directly aligned the features by learning dictionaries for each view. Moreover, the better performance obtained by un-RLTDL over un-RSTDL demonstrates that the relaxation of the regularization of sparse codes enables us to learn better dictionaries for reconstruction.

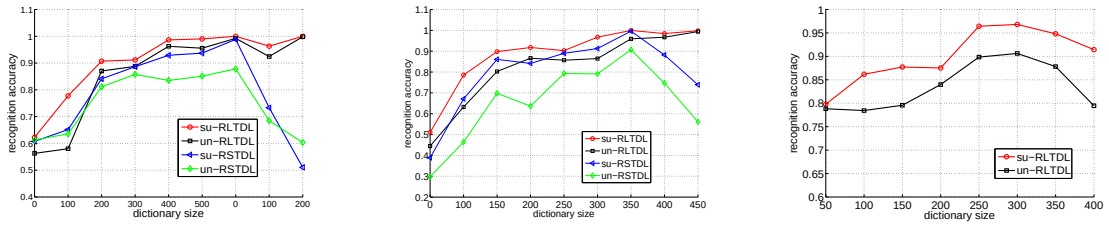
Table 3.7 shows recognition accuracies of cross-view action recognition under the supervised mode. It can be observed that su-RLTDL outperforms su-RSTDL,

%	C0	C1	C2	C3	C4	C5	C6	C7
C0		(100, 100)	(100, 100)	(100, 100)	(100, 100)	(99.6, 99.6)	(99.2, 99.8)	(100, 100)
C1	(100, 100)		(100, 100)	(100, 100)	(99.8, 99.8)	(97.2, 99.4)	(99.0, 100)	(100, 100)
C2	(84.3, 89.9)	(98.4, 98.4)		(93.0, 97.8)	(92.1, 97.5)	(91.9, 99.2)	(89.4, 97.7)	(90.5, 96.7)
C3	(81,2, 99.6)	(76.8, 96.1)	(81.0, 79.4)		(80.9, 96.5)	(81.4, 76.0)	(73.8, 81.0)	(79.6, 79.6)
C4	(90.9, 99.0)	(92.3, 97.7)	(93.0, 93.6)	(99.0, 95.9)		(92.3, 99.8)	(90.9, 90.7)	(93.2, 97.7)
C5	(82.2, 90.3)	(90.9, 90.9)	(88.6, 87.2)	(80.6, 84.7)	(85.9, 88.8)		(89.4, 91.7)	(88.0, 90.9)
C6	(81.8, 82.4)	(81.8, 90.1)	(89.9, 91.8)	(86.7, 93.2)	(82.2, 93.6)	(90.5, 95.6)		(82.7, 90.0)
C7	(90.9, 94.6)	(90.9, 100)	(95.9, 99.8)	(91.5, 98.1)	(93.6, 100)	(94.8, 99.0)	(99.2, 99.6)	
Ave.	(87.3, 93.7)	(90.1, 96.2)	(92.6, 93.1)	(93.0, 96.0)	(90.6, 96.6)	(92.5, 95.5)	(91.6, 94.4)	(90.6, 93.5)

Table 3.7: Cross-view action recognition accuracies of different approaches on the WVU dataset using supervised correspondence mode. Each row corresponds to a source (training) view and each column a target (test) view. The accuracy numbers in the bracket are the average recognition accuracies of our proposed su-RSTDL and su-RLTDL approaches respectively.

but also improves the accuracies based on our unsupervised approaches. This again demonstrates that the dictionaries learned using labeled information across views are more discriminative. In addition, un-RLTDL surprisedly outperforms su-RSTDL which demonstrates that the separation of the common dictionary from view-specific dictionaries enable us to align view-specific features better.

We also evaluate the effect of dictionary size of the common dictionary D and view-specific dictionaries D^v on our approaches. Figure 3.7 shows the performance of our approaches on three pairs of source and target combinations with varying

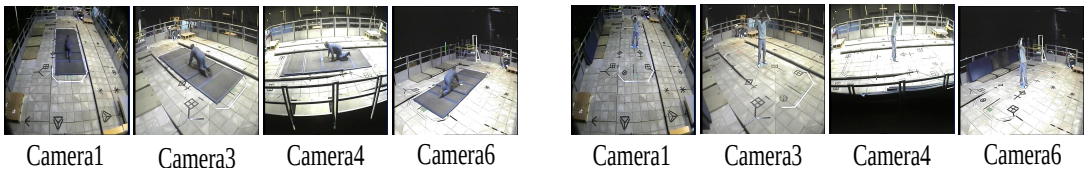


(a) Camera0 v.s. Camera3 (b) Camera0 v.s. Camera6 (c) Camera2 v.s. Camera6

Figure 3.7: Performance on the WVU action dataset with varying dictionary size.

dictionary size. For Figure 3.7 (a)(b), we fix the dictionary size of D to be 50, and vary the dictionary size of D^v from the range of $\{50, 100, 150, 200, 250, 300\}$. We observe that the performance of our approaches increases as the dictionary size of D^v increases. For Figure 3.7 (c), we fix the dictionary size of D^v to be 300, and change the dictionary size of D from the range of $\{50, 100, 150, 250, 300\}$. It can be seen that our approaches achieve high recognition accuracies even using a very small size dictionary. However, when the dictionary size of D is too large, the redundancy in dictionaries will affect the sparse representation of test videos and the performance may decrease.

Figure 3.7 shows the performance of our approaches on three pairs of source and target combinations with varying dictionary size. For Figure 3.7(a)(b), we fix the dictionary size of D to be 50, and vary the dictionary size of D^v from the range of $\{50, 100, 150, 200, 250, 300, 350, 400, 450\}$. Figure 3.7(c), we fix the dictionary size of D^v to be 300, and change the dictionary size of D from the range of $\{50, 100, 150, 250, 300, 350, 400\}$. We observe that the recognition accuracies of un-RSTDL and su-RSTDL first increase as the dictionary size of view-specific dic-



(a) Action class: CrawlOnKnees (b) Action class: SmashObject

Figure 3.8: Exemplar frames from the HuAVi action dataset.

tionaries increases, and drop drastically when the dictionary size is larger than 350. However, the performances of both un-RLTDL and su-RLTDL consistently increase as the dictionary size of view-specific dictionaries increases. One possible reason is that in the RSTDL approach, videos of orphan actions in each view tend to select different sets of dictionary atoms to represent when the dictionary size is too large.

3.7.3 Evaluation on the MuHAVi dataset

MuHAVi dataset [101] contains a large body of human action video data from 17 human action classes. These action classes are WalkTurnBack, RunStop, Punch, Kick, ShotGunCollapse, PullHeavyObject, PickupThrowObject, WalkFall, LookInCar, CrawlOnKnees, WaveArms, DrawGraffiti, JumpOverFence, DrunkWalk, ClimbLadder, SmashObject, JumpOverGap. Each action video is performed by 7 actions and recorded using 9 CCTV Schwan cameras located at 4 sides and 4 corners of a rectangular platform. Due to the computational complexity, we followed [123] to choose the action videos captured by four cameras (i.e. two side cameras and two corner cameras) in our experiments. Figure 8 shows exemplar frames of two action classes taken by four cameras.

%	C0	C1	C2	C3
C0		(77.3, 99.8, 99.2, 99.8)	(73.9, 96.6, 91.6, 99.8)	(82.4, 99.8, 97.5, 99.8)
C1	(68.9, 98.3, 92.4, 99.8)		(68.9, 98.3, 94.1, 99.8)	(72.3, 99.8, 85.7, 99.8)
C2	(83.2, 98.3. 87.4, 99.8)	(68.9, 97.5, 97.5, 99.8)		(84.9, 99.2, 89.9, 99.8)
C3	(71.4, 95.0, 94.1, 99.8)	(77.3, 89.1, 93.3, 99.8)	(58.0, 92.4, 88.2, 99.8)	
Ave.	(74.5, 98.6, 84.0, 99.8)	(74.5, 98.9, 86.8, 99.8)	(66.9, 97.8, 85.4, 99.8)	(79.8, 99.8, 81.5, 99.8)

Table 3.8: Cross-view action recognition accuracies on the MuHAVi dataset. Each row corresponds to a source (training) view and each column a target (test) view. The accuracy numbers in the bracket are the average recognition accuracies of un-RSTDL, un-RLTDL,su-RSTDL, and su-RLTDL respectively.

%	View1	View3	View4	View6	Avg
SVM	93.3	92.4	93.3	95.8	93.7
LSSVM	91.6	94.1	95.8	95.8	94.3
LKSSVM	96.6	93.3	94.1	94.1	94.5
un-RSTDL	78.2	79.0	75.6	83.2	79.0
su-RSTDL	81.5	89.0	91.6	86.6	87.2
un-RLTDL	**96.6**	**97.5**	**99.8**	**99.8**	**98.5**
su-RLTDL	**99.8**	**99.8**	**99.8**	**99.8**	**99.8**

Table 3.9: Multi-view action recognition results on the MuHAVi dataset. Each column corresponds to one target view.

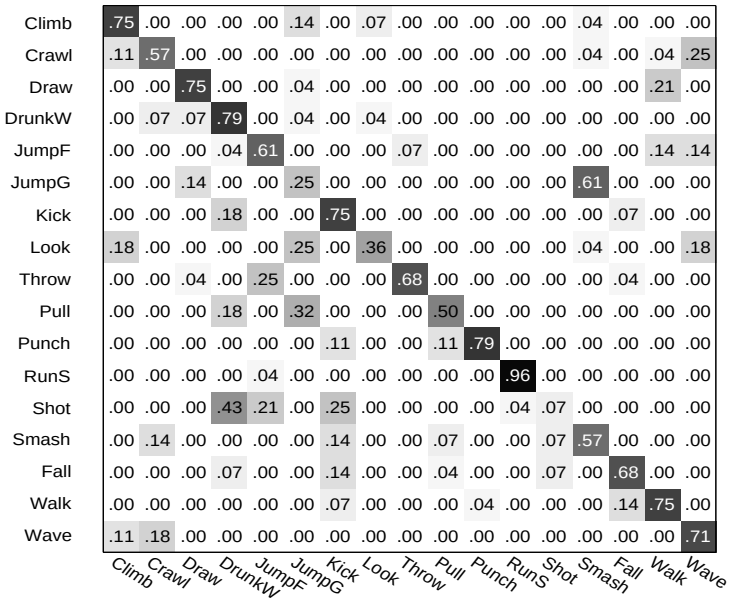
Table 3.8 shows the recognition accuracies of our approaches for cross-view action recognition. Both su-RSTDL and su-RLTDL yield a better performance than un-RSTDL and un-RLTDL respectively. Note that su-RLTDL even outperforms un-

RSTDL by a margin of 20%. This demonstrates the benefits of the separation of the common dictionary from view-specific dictionaries.

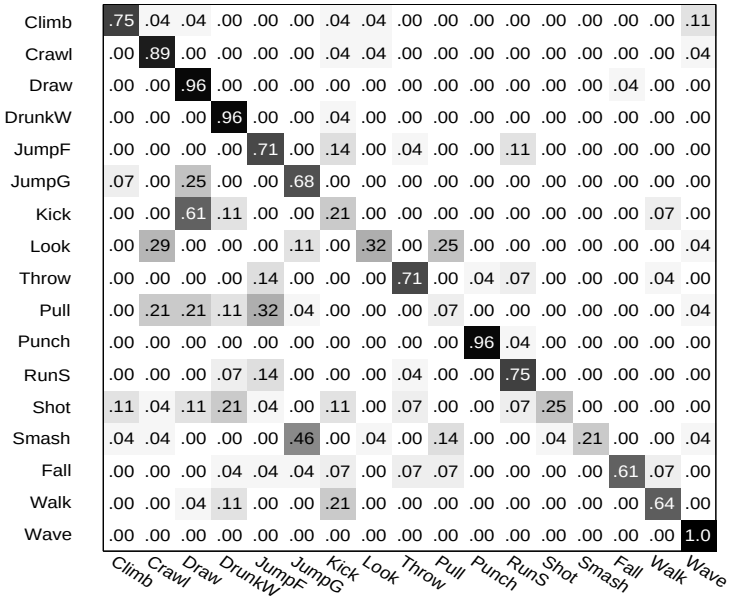
The proposed approaches are also compared with three state-of-the-art algorithms: (1) nonlinear SVM [11], which adopted a χ^2 kernel and one-against-all setting for multi-class classification task; (2) latent structural SVM [133], which modeled the camera views as a latent variable. (3) latent kernelized structural SVM [123] which extended the kernelized structural SVM framework to include the camera views as latent variables. Table 3.9 shows the recognition accuracies of different approaches for multi-view action recognition. It can be seen that the performance of both un-RSTDL and su-RSTDL is worse than comparing algorithms, whereas un-RLTDL and su-RLTDL consistently outperform all the competing approaches. This again illustrates that the separation of the common dictionary enable us to learn more compact view-specific dictionaries. In addition, the confusion matrices for un-RSTDL, su-RSTDL, un-RLTDL and su-RLTDL are shown in Figure 3.9 and 3.10.

3.8 Summary

In this chapter, we introduced two effective transferable dictionary learning-based approaches for robust action recognition across views. In the first method, we learn a view-specific dictionary for each view. By forcing the shared action videos across different views to have the same sparse representations, the set of dictionary is made to have the transferability property. This is because action

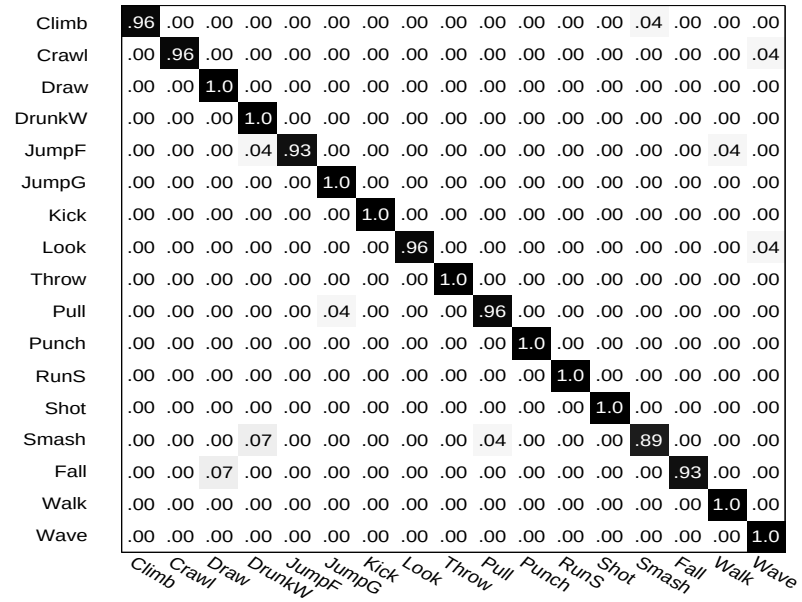


(a) un-RSTDL

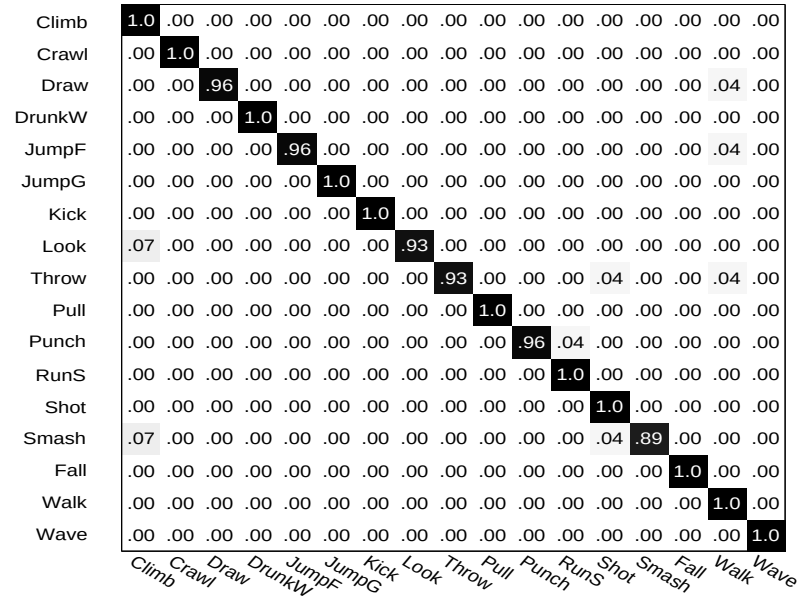


(b) su-RSTDL

Figure 3.9: Confusion matrices for our proposed RSTDL approach on the MuHAVi dataset.



(a) un-RLTDL



(b) su-RLTDL

Figure 3.10: Confusion matrices for our proposed RLTDL approach on the MuHAVi dataset.

videos of the same class in different views encoded using the corresponding view-dependent dictionary tend to have the same sparse representations. Using the set of transferable dictionaries, we can directly transfer action models across views. In the second method, we additionally learn a common dictionary shared by different views to model view-shared features. Both the common dictionary and the corresponding view-specific dictionary are used to represent videos of each view. We transfer the indices of non-zeros in sparse codes of videos from the source view to the sparse codes of the corresponding videos from the target view. In this way, the mapping between the source and target view is encoded in the common dictionary and view-specific dictionaries. Meanwhile, the associated sparse representations are view-invariant because the non-zeros positions in the sparse codes of correspondence videos share the same set of indices. In addition, our approach can be applied to cross-view and multi-view action recognition under the unsupervised, supervised and domain adaptation settings.

Our approaches have two limitations that need to be addressed. First, we need sets of videos of the same class taken from different views to learn the transferable dictionaries. However, videos in different views may be not aligned. Future work includes extending our approach to handle this case. It will exploit the relationship between different views more flexibly. Second, the view of test videos are given at first and we did not fuse the knowledge from different training views for multi-view action recognition. A more flexible approach is to automatically estimate the view of test videos and classify the test videos by fusing knowledge from different training views.

Chapter 4: Semantic Taxonomy Aware Dictionary Learning for Image Tagging

The goal of image tagging is to assign image regions with labeled tags, has attracted significant attention in computer vision and multimedia [71, 130, 35, 135, 125, 126]. Region tagging at a more fine-grained region-level has two benefits. First, it establishes the correspondences between image regions and semantic labels and thus can handle the diversity and arbitrariness of Web image content well. Second, experiments in [22, 125] reveal that accurate region-level annotations can effectively boost the performance of image-level annotations. In order to achieve robust content-based image retrieval, we focus on improving the accuracy of region tagging.

Recently several proposed region tagging approaches attempt to explore the contextual constraints among image regions using sparse coding techniques [71, 130, 35]. However, these approaches that simply used all training regions as the dictionary for spare coding have three main disadvantages. First, redundancy in training regions can increase the reconstruction error, which may degrade the effectiveness of region tagging. Second, the computational complexity of sparse coding increases with the size of dictionary and it is impossible to use all the training regions as the dictionary for large-scale datasets. Thus learning a compact and discrimina-

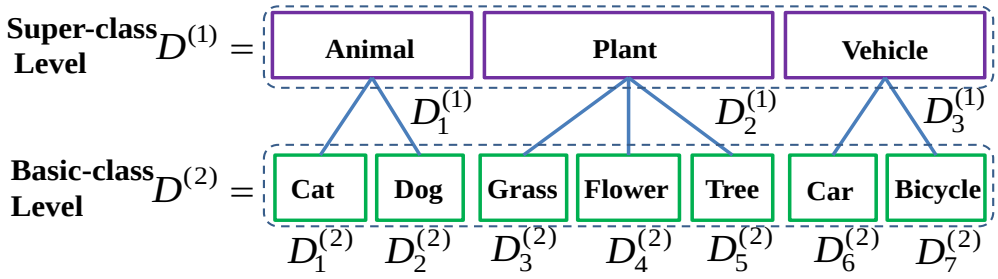


Figure 4.1: A two-layer tag taxonomy and the corresponding dictionary framework. This tag taxonomy has two levels: super-class level and basic-class level. At the super-class level, training samples are divided into three super-classes *Animal*, *Plant* and *Vehicle*, whereas training samples within each super-class are further divided into a few basic classes. We associate each tag node with a node-specific dictionary and concatenate the node-specific dictionaries from each level to create a level-specific dictionary. The level-specific dictionaries for this taxonomy are $D^{(1)}$ and $D^{(2)}$ while the node-specific dictionaries are $\{D_s^{(1)}\}_{s=1...3}$ and $\{D_k^{(2)}\}_{k=1...7}$. We reconstruct each image region using different level-specific dictionaries and sum up the sparse codes obtained from different levels as the final feature representation to learn a linear classifier for region tagging.

tive dictionary for region tagging is desirable. Third, for datasets with unbalanced tag classes, the performance of these approaches may decrease drastically. This is because unbalanced tag classes result in an unbalanced group structure in the dictionary such that the computed sparse codes become less discriminative for classification task. In addition, tags are often arranged into a hierarchical taxonomy based on their semantic meanings, such as the tag taxonomy shown in Figure 4.1. However, the tag taxonomy has not been exploited to improve the accuracy of re-

gion tagging, even though the similar category taxonomy has been shown to benefit the accuracy as well as the scalability of learning algorithms [77, 78, 34] for object recognition.

To overcome the above drawbacks, we present a novel multi-layer hierarchical dictionary learning framework for region tagging when the tag taxonomy is known. For illustration, a two-layer tag taxonomy and the corresponding dictionary learning framework is depicted in Figure 4.1. To our best knowledge, we are the first to use the supervised dictionary learning to explore the semantic relationship among tags. Specifically, we generate a node-specific dictionary for each tag node in the taxonomy and concatenate the node-specific dictionaries in each level to construct a level-specific dictionary. Thus the hierarchical semantic relationship among tags is preserved in the relationship among node-specific dictionaries, which enables us to exploit the discriminative information among regions in a hierarchial way. Moreover, dictionary items from the same node-specific dictionary are considered as a group so it introduces a group structure for each level-specific dictionary. Based on each level-specific dictionary and corresponding group structure, we reconstruct each image region using the group sparse coding algorithm [136] to obtain level-specific sparse codes. Compared with single-level sparse codes in existing sparse coding-based region tagging approaches [71, 130, 35], our multi-layer sparse codes not only encodes the contextual constraints among regions, but also encodes the relationship among tags. Finally, we sum up the sparse codes obtained from different levels as the final feature representation to learn a linear class classifier. For datasets with unbalanced tag classes, we can create balanced group structure for higher levels and make use

of sparse codes obtained from higher levels to help design the classifiers for lower levels. Therefore, our approach is robust to datasets with unbalanced tag classes in contrast to existing sparse coding-based region tagging approaches that tend to perform poorly on datasets with unbalanced tag classes.

4.1 Related Work

Recently, several region tagging approaches have used sparse coding techniques to encode contextual constraints among image regions for region tagging [71, 130, 35]. [71] proposed a bi-layer sparse coding framework to reconstruct image regions from over-segmented image patches that belong to a few images, and then propagate image labels of selected patches to the entire label to obtain region assignment. However, this method ignores the contextual correlations among regions, *e.g.*, co-occurrence and spatial correlations. [130] considered regions within the same image as a group, and used the group sparse coding with spatial kernels to jointly reconstruct image regions in the same image from other training regions. However, the contextual correlations of training regions across images are ignored due to the group structure of regions-in-image relationship. [35] extended group sparse coding with graph-guided fusion penalty to encourage highly correlated regions to be jointly selected for the reconstruction. However, the performance of the group sparse coding depends on a balanced group structure which has the similar number of training regions in each group so it might not be robust to datasets that have very unbalanced training regions.

Other techniques have also been proposed to boost the performance for region tagging or region-based image annotation. [125, 126] used multiple-instance learning techniques to learn the correspondence between image regions and keywords. The idea is that each image is annotated by the tag that has at least one sample region (seen as '*instance*') within this image (seen as '*bag*'). [135] regularized segmented image regions into $2D$ lattice layout, and employed a simple grid-structure graphical model to characterize the spatial context constraints. [22] used both the dominant image region and the relevant tags to annotate the semantics of natural scenes. [63] proposed a unified solution to tag refinement and tag-to-region assignment by using a multi-edge graph, where each vertex of the graph is a unique image encoded by a region bag with multiple image segmentations. [31] proposed a multi-layer group sparse coding framework to encode the mutual dependence between the class labels as well as the tag distribution information.

Supervised dictionary learning which combines dictionary learning with classifier training into a unified learning framework has been extensively studied [129, 87, 76, 137]. [129] performed supervised dictionary learning by minimizing the training error of classifying the image-level features, which are extracted by max pooling over the sparse codes within a spatial pyramid. [76] proposed a novel sparse representation of signals belonging to different classes in terms of a shared dictionary and discriminative models. This approach alternates between the step of sparse coding and the step of dictionary update and discriminative model learning. [137] extended the K-SVD algorithm by incorporating the classification error into an objective function that allows the simultaneous optimization of the dictionary and classifiers. In

addition, [39, 12] proposed to use proximal methods for structured sparse learning where dictionary items are embedded in different structures.

4.2 Tag Taxonomy Aware Dictionary Learning

In this section, we first introduce the group sparse coding algorithm and then describe the formulation of our multi-layer supervised dictionary learning, its optimization and how to tag image regions using sparse codes.

4.2.1 Group Sparse Coding

Given a dictionary $D = [D_1, D_2, ..., D_G] \in \mathbb{R}^{d \times J}$ where $D_g \in \mathbb{R}^{d \times J_g}$ consists of a group of J_g visually correlated dictionary items, an image region $x \in \mathbb{R}^d$ can be reconstructed from the dictionary with the group LASSO penalty [136] as follows:

$$\begin{aligned} \mathbf{z} &= arg\min_{\mathbf{z}} \frac{1}{2}||x - \sum_{g=1}^{G} D_g z_g||_2^2 + \lambda \sum_{g=1}^{G} \beta_g ||z_g||_2 \\ &= arg\min_{\mathbf{z}} \frac{1}{2}||x - D\mathbf{z}||_2^2 + \lambda \sum_{g=1}^{G} \beta_g ||z_g||_2 \end{aligned} \tag{4.1}$$

where $\mathbf{z} = [z_1^T, z_2^T, ..., z_G^T]^T \in \mathbb{R}^{J \times 1}$ is the reconstruction coefficients where z_g is the encoding coefficient corresponding to the g^{th} group. And $\lambda \geq 0$ is a trade-off parameter and $\beta_g = \sqrt{J_g}$ weights the penalty from the g-th group. Since the group LASSO uses a group-sparsity-inducing regularization instead of the l_1 norm as in LASSO [105], we can treat multiple visually similar dictionary items within the same group as a whole and exploit implicit relations among these dictionary items to some extent.

4.2.2 Multi-layer Supervised Dictionary Learning

We consider an image dataset $\mathcal{D}$ with a two-layer tag taxonomy whose levels from the top to the bottom are called: super-class level and basic-class level as shown in Figure 4.1. Note that extensions to learning multiple level-specific dictionaries for a multi-layer tag taxonomy can be accomplished in a similar way. Suppose that each image has been segmented into regions and a d-dimensional feature vector has been extracted for each region. Let $X \in \mathbb{R}^{d \times N}$ denote N training image regions from K tag classes. According to the tag taxonomy, image regions from these K classes in the basic-class level can be merged into S super-classes in the super-class level, *e.g.*, *cat* and *dog* belong to the super-class *animal* , whereas *grass* and *tree* belong to the super-class *plant* (See Figure 4.1). Thus each image region has one class label from the basic-class level and one super-class label from the super-class level. Let $H^{(2)} \in \{0,1\}^{K \times N}$ denote the class label indicator matrix for all the regions, where $H^{(2)}_{(i,j)} = 1$ if the jth image region belongs to the ith tag and $H^{(2)}_{(i,j)} = 0$ otherwise. Similarly, we use $H^{(1)} \in \{0,1\}^{S \times N}$ to denote the super-class label indicator matrix respectively. Note that we use the superscript to index the level in the tag taxonomy and the subscript to index the node-specific dictionary in that level.

Given an underlying tag taxonomy, we associate a separate dictionary with each tag node. These individual dictionaries are called node-specific dictionaries and they serve as local viewpoints for exploring the discriminative information among training regions from the same class or super-class. We concatenate the node-specific dictionaries in each level to construct a new large dictionary which is called a level-

specific dictionary. Suppose that the level-specific dictionaries in the super-class and basic-class levels are learned and represented as $D^{(1)} = [D_1^{(1)}, D_2^{(1)}, ..., D_S^{(1)}] \in \mathbb{R}^{d\times J}$ and $D^{(2)} = [D_1^{(2)}, D_2^{(2)}, ..., D_K^{(2)}] \in \mathbb{R}^{d\times J}$, where $D_s^{(1)}$ and $D_k^{(2)}$ are associated with the s-th super-class and k-th class respectively. Given level-specific dictionaries $D^{(1)}, D^{(2)}$ and a region $\mathbf{x}_n \in \mathbb{R}^{d\times 1}$from the s-th superclass and k-th class, we obtain the group sparse representations $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ of this region as follows:

$$\begin{aligned}\mathbf{z}_n^{(1)} &= arg\min_{\mathbf{z}_n^{(1)}} \frac{1}{2}||\mathbf{x}_n - D^{(1)}\mathbf{z}_n^{(1)}||_2^2 + \lambda_1 \sum_{s=1}^{S} \beta_s^{(1)} ||z_{n_s}^{(1)}||_2 \\ \mathbf{z}_n^{(2)} &= arg\min_{\mathbf{z}_n^{(2)}} \frac{1}{2}||\mathbf{x}_n - D^{(2)}\mathbf{z}_n^{(2)}||_2^2 + \lambda_2 \sum_{k=1}^{K} \beta_k^{(2)} ||z_{n_k}^{(2)}||_2.\end{aligned} \tag{4.2}$$

Here we introduce $\mathbf{q}_n^{(1)}$ and $\mathbf{q}_n^{(2)}$ to denote the 'ideal' group sparse codes of $\mathbf{x}_n$ corresponding to $D^{(1)}$ and $D^{(2)}$ respectively. In particular, the non-zero values of $\mathbf{q}_n^{(1)}$ or $\mathbf{q}_n^{(2)}$ occur at those indices where the dictionary items belong to the node-specific dictionary $D_s^{(1)}$ or $D_k^{(2)}$. We use $Z^{(1)} = [\mathbf{z}_1^{(1)}, ..., \mathbf{z}_N^{(1)}] \in \mathbb{R}^{J\times N}$ to denote the group sparse codes of all regions at the super-class level. The matrices $Z^{(2)}, Q^{(1)}, Q^{(2)}$ are defined in a similar way.

Based on the sparse representations from the super-class and basic-class levels, we aim to learn two linear classifiers denoted as $f^{(1)}(\mathbf{z}, W_s) = W_s\mathbf{z}$ and $f^{(2)}(\mathbf{z}, W) = W\mathbf{z}$ for the two levels respectively, where $W_s \in \mathbb{R}^{S\times J}$ and $W \in \mathbb{R}^{K\times J}$. The objective function for learning all the dictionaries and classifiers are formulated as:

$$\min_{D^{(i)}{}_{i=1}^{2}, W_s, W} ||H^{(1)} - W_s Z^{(1)}||^2 + ||H^{(2)} - W(Z^{(1)} + Z^{(2)})||^2 \tag{4.3}$$

$$+ \nu(||Q^{(1)} - Z^{(1)}||^2 + ||Q^{(2)} - Z^{(2)}||^2) + \mu(||W_s||_2^2 + ||W||_2^2) \tag{4.4}$$

where $Z^{(1)} = [\mathbf{z}_1^{(1)}, ..., \mathbf{z}_N^{(1)}], Z^{(2)} = [\mathbf{z}_1^{(2)}, ..., \mathbf{z}_N^{(2)}], Q^{(1)} = [\mathbf{q}_1^{(1)}, ..., \mathbf{q}_N^{(1)}], Q^{(2)} = [\mathbf{q}_1^{(2)}, ..., \mathbf{q}_N^{(2)}$.

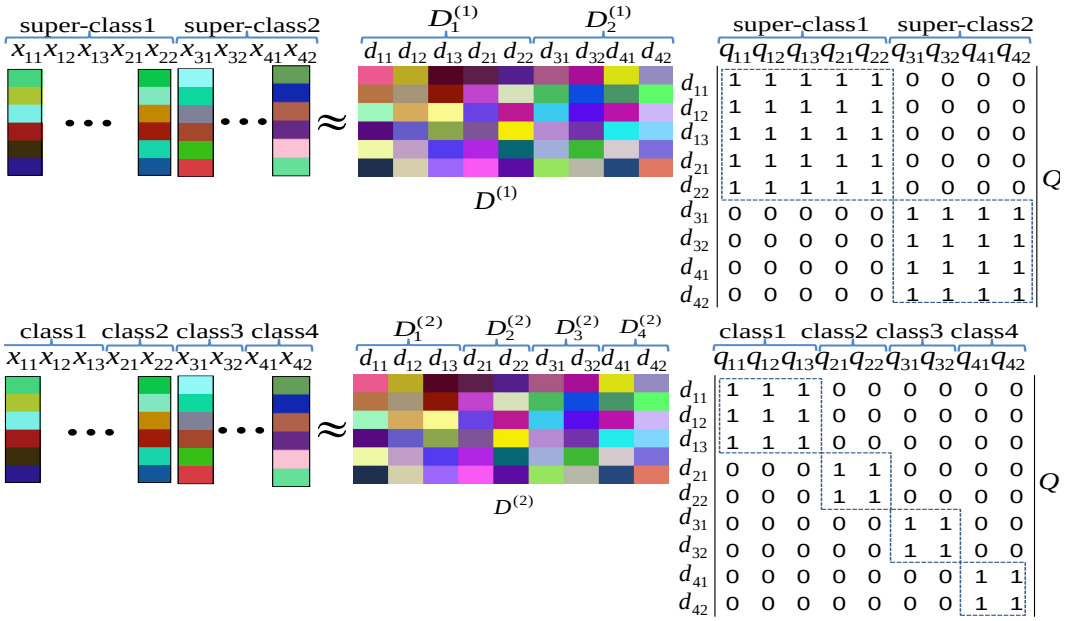


Figure 4.2: An example of the *ideal* sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ for classification task. Given nine image regions (on the leftmost) come from four basic-classes and two super-classes, we learn two level-specific dictionaries for the super-class and basic-class levels respectively. The super-class level dictionary is defined as: $D^{(1)} = [D_1^{(1)}, D_2^{(1)}]$ while the basic-level dictionary is $D^{(2)} = [D_1^{(2)}, D_2^{(2)}, D_3^{(2)}, D_4^{(2)}]$. For each region from one labeled tag, we aim to use only the node-specific dictionary that is associated with the same tag to reconstruct the region. This is because image regions from the same basic-class or super-class are more likely to share visual features and thus can be used to reconstruct each other.

Note that this is a constrained optimization problem where the constraint is that matrices $Z^{(1)}$ and $Z^{(2)}$ are obtained by minimizing the reconstruction error with group LASSO penalty from the basic-class and super-class levels as shown in (4.2). This objective function consists of two parts:

1. The first part is the classification error from each level as shown in the first line of (4.4). The two classifiers W_s and W are learned by the linear regression. Note that W_s is not used for final region tagging. W_s is learned to guarantee that the sparse codes obtained from the super-class level are discriminative and thus can be used to help learn W for the basic-class level.

2. The second part is the regularization of sparse codes from two levels as shown in the second line of (4.4). The *ideal* sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ are block-diagonal as shown in Figure 4.2. We call sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ *ideal* because they are *ideal* for classification task. We minimize the difference between the true sparse codes and the corresponding *ideal* sparse codes to encourage the true sparse codes to be close to the *ideal* sparse codes. It means that for training regions X_k from the k-th class and X_s from the s-th super-class, we encourage the corresponding node-dictionaries $D_s^{(1)}$ and $D_k^{(2)}$ to be selected for group sparse coding. In addition, the non-zeros in $Q^{(2)}$ are a subset of non-zeros in $Q^{(1)}$. Note that this fixed and structured relationship between $Q^{(1)}$ and $Q^{(2)}$ regularizes the relationship between $Z^{(1)}$ and $Z^{(2)}$ from two levels, which makes it possible to use sparse codes from different levels to improve classification accuracy.

Note that we use the sum of sparse codes from two levels as the features to design the class classifier W for two reasons. First, we make use of the discriminative information encoded in the sparse codes obtained from the super-class level to learn W. Second, it encourage classes within the same super-class to implicitly share

sparse codes obtained from super-class level. This can handle the situation where the training classes are very unbalanced. For example, there are many training regions for the tag *cat* but little training regions for *dog*. Given the feature of an image region from *dog*, it can be reconstructed using the level-specific dictionary from the basic-class level, which may activate multiple node-specific dictionaries in the basic-class level. This is due to the little training regions for the tag *dog* and it will be difficult to classify the class label of this image region. However, when using the level-specific dictionary from the super-class level to reconstruct this image region, it may only activate the node-specific dictionary associated with the super-class *animal*. This is because other tags within the same super-class *animal* may share some features with *dog* and can help to represent this image region better other than *dog* itself. Even if we cannot classify this image region as *dog*, we can at least classify this image regions as other tags that belong to the super-class *animal* instead of totally uncorrelated tags from other super-classes. Thus using the sum of sparse codes from two levels as features for designing the class classifiers can support this implicit feature sharing among classes within the same super-class.

4.2.3 Optimization Algorithm

Motivated by [74], we propose a stochastic gradient descent algorithm for optimizing the objective function. We first rewrite the objective function in (4.4) as follows:

$$\min_{D^{(i)}{}_{i=1}^{2},W_s,W} \sum_{i=1}^{N} \ell^n(D^{(1)}, D^{(2)}, W_s, W) + \mu(||W_s||_2^2 + ||W||_2^2)$$

where

$$\ell^n = \nu(||\mathbf{q}_n^{(1)} - \mathbf{z}_n^{(1)}||^2 + ||\mathbf{q}_n^{(2)} - \mathbf{z}_n^{(2)}||^2) + ||\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)}||^2 + ||\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})||^2.$$

Note that the sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ are functions of $D^{(1)}$ and $D^{(2)}$ respectively. We use the notation $\ell^n(D^{(1)}, D^{(2)}, W_s, W)$ to emphasize that the loss function associated with the n-th region is also a function of $D^{(1)}$ and $D^{(2)}$. We use the following procedure to optimize the objective function: first, we randomly select a training instance $(\mathbf{x}_n, \mathbf{h}_n^{(1)}, \mathbf{h}_n^{(2)})$ for the t-th iteration; next, we compute the sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ using $D^{(1)}$ and $D^{(2)}$ by (4.2); finally, we update $D^{(1)}, D^{(2)}, W_s$ and W by the gradients of the loss function ℓ^n with respect to them.

We next describe the methods for computing the gradients of the loss function ℓ^n with respect to the level-specific classifiers and dictionaries. When the sparse codes $z_n^{(1)}$ and $z_n^{(2)}$ are known, we can compute the gradient of ℓ^n with respect to W_s and W as follows:

$$\begin{aligned} \frac{\partial \ell^n}{\partial W_s} &= -2(\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)})\mathbf{z}_n^{(1)T} \\ \frac{\partial \ell^n}{\partial W} &= -2(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)}))(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})^T. \end{aligned} \tag{4.5}$$

We use the chain rule to compute the gradient of ℓ^n with respect to $D^{(1)}$ and $D^{(2)}$ as follows:

$$\frac{\partial \ell^n}{\partial D^{(1)}} = \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(1)}}\frac{\partial \mathbf{z}_n^{(1)}}{\partial D^{(1)}}, \frac{\partial \ell^n}{\partial D^{(2)}} = \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(2)}}\frac{\partial \mathbf{z}_n^{(2)}}{\partial D^{(2)}} \tag{4.6}$$

where

$$\begin{aligned} \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(1)}} &= -2W_s^T(\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)}) - 2W^T(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})) - 2\nu(\mathbf{q}_n^{(1)} - \mathbf{z}_n^{(1)}) \\ \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(2)}} &= -2W^T(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})) - 2\nu(\mathbf{q}_n^{(2)} - \mathbf{z}_n^{(2)}). \end{aligned}$$

To compute the gradient of $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ with respect to $D^{(1)}$ and $D^{(2)}$, we use implicit differentiation on the fixed point equation similar to [74, 129, 128]. We first establish the fixed point equation of (4.2) by calculating the derivatives of $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ on both sides and have:

$$\begin{aligned} D_\Lambda^{(1)T}(\mathbf{x}_n - D_\Lambda^{(1)}\mathbf{z}_{n\Lambda}^{(1)}) &= \lambda_1\Gamma^{(1)}[\frac{z_{n_1\Lambda}^{(1)T}}{||z_{n_1\Lambda}^{(1)}||_2}, ..., \frac{z_{n_S\Lambda}^{(1)T}}{||z_{n_S\Lambda}^{(1)}||_2}]^T \\ D_\Lambda^{(2)T}(\mathbf{x}_n - D_\Lambda^{(2)}\mathbf{z}_{n\Lambda}^{(2)}) &= \lambda_2\Gamma^{(2)}[\frac{z_{n_1\Lambda}^{(2)T}}{||z_{n_1\Lambda}^{(2}||_2}, ..., \frac{z_{n_K\Lambda}^{(2)T}}{||z_{n_K\Lambda}^{(2)}||_2}]^T \end{aligned} \tag{4.7}$$

where Λ denote the index set of non-zero sparse coefficients in $z_n^{(1)}$ and $z_n^{(2)}$. Both $\Gamma^{(1)}$ and $\Gamma^{(2)}$ are block-diagonal. The s-th block in $\Gamma^{(1)}$ is $\beta_s^{(1)}I_s$ while the k-th block in $\Gamma^{(2)}$ is $\beta_k^{(2)}I_k$, where I_s, I_k are the corresponding identity matrices. We calculate the derivatives of $D^{(1)}$ and $D^{(2)}$ on both sides of (4.7), and have

$$\begin{aligned} \frac{\partial \mathbf{z}_{n\Lambda}^{(1)}}{\partial D_\Lambda^{(1)}} &= (D_\Lambda^{(1)T}D_\Lambda^{(1)} + \lambda_1\Gamma^{(1)}A^{(1)})^{-1}[\frac{\partial D_\Lambda^{(1)T}\mathbf{x}_n}{\partial D_\Lambda^{(1)}} - \frac{\partial D_\Lambda^{(1)T}D_\Lambda^{(1)}}{\partial D_\Lambda^{(1)}}\mathbf{z}_{n\Lambda}^{(1)}] \\ \frac{\partial \mathbf{z}_{n\Lambda}^{(2)}}{\partial D_\Lambda^{(2)}} &= (D_\Lambda^{(2)T}D_\Lambda^{(2)} + \lambda_2\Gamma^{(2)}A^{(2)})^{-1}[\frac{\partial D_\Lambda^{(2)T}\mathbf{x}_n}{\partial D_\Lambda^{(2)}} - \frac{\partial D_\Lambda^{(2)T}D_\Lambda^{(2)}}{\partial D_\Lambda^{(2)}}\mathbf{z}_{n\Lambda}^{(2)}] \end{aligned}$$

where the matrices $A^{(1)}$ and $A^{(2)}$ are block-diagonal and the s-th block in $A^{(1)}$ is $\frac{||z_{n_s\Lambda}^{(1)}||I_s - z_{n_s\Lambda}^{(1)}z_{n_s\Lambda}^{(1)T}}{||z_{n_s\Lambda}^{(1)}||_2^2}$ while the k-th block in $A^{(2)}$ is $\frac{||z_{n_k\Lambda}^{(2)}||I_k - z_{n_k\Lambda}^{(2)}z_{n_k\Lambda}^{(2)T}}{||z_{n_k\Lambda}^{(2)}||_2^2}$. Therefore, (4.6) can be rewritten as

$$\begin{aligned} \frac{\partial \ell^n}{\partial D^{(1)}} &= -D^{(1)}\mathbf{s}_n^{(1)}\mathbf{z}_n^{(1)T} + (\mathbf{x}_n - D^{(1)}\mathbf{z}_n^{(1)})\mathbf{s}_n^{(1)T} \\ \frac{\partial \ell^n}{\partial D^{(2)}} &= -D^{(2)}\mathbf{s}_n^{(2)}\mathbf{z}_n^{(2)T} + (\mathbf{x}_n - D^{(2)}\mathbf{z}_n^{(2)})\mathbf{s}_n^{(2)T} \end{aligned} \tag{4.8}$$

where the auxiliary variables $\mathbf{s}_n^{(1)}$ and $\mathbf{s}_n^{(2)}$ are defined as follows:

$$\begin{aligned} \mathbf{s}_{\Lambda^C}^{(1)} = 0, \mathbf{s}_\Lambda^{(1)} &= (D_\Lambda^{(1)T}D_\Lambda^{(1)} + \lambda_1\Gamma^{(1)}A^{(1)})^{-1}\frac{\partial \ell^n}{\partial \mathbf{z}_{n\Lambda}^{(1)}} \\ \mathbf{s}_{\Lambda^C}^{(2)} = 0, \mathbf{s}_\Lambda^{(2)} &= (D_\Lambda^{(2)T}D_\Lambda^{(2)} + \lambda_2\Gamma^{(2)}A^{(2)})^{-1}\frac{\partial \ell^n}{\partial \mathbf{z}_{n\Lambda}^{(2)}}. \end{aligned}$$

The steps $1-15$ in Algorithm 1 summarize our joint learning algorithm.

Algorithm 3: Multi-layer Supervised Dictionary Learning for Region Tagging (MSDL)

Part 1: Dictionary Learning

Input: X (training regions), $H^{(1)}$ (super-class label indicator matrix), $H^{(2)}$ (class label indicator matrix), $\mathbf{D}$ (initial dictionary), T (number of iterations), N (number of training samples), ρ (initial learning rate), ν, μ, n_0.

Output: classifiers W_s and W; dictionaries $D^{(1)}$ and $D^{(2)}$

for $t = 1...T$ **do**

Permute training samples $(X, H^{(1)}, H^{(2)})$;

for $n = 1...N$ **do**

Evaluate the group sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ of the region $\mathbf{x}_n$;

Choose the learning rate $\rho_t = \min(\rho, \rho * n_0/n)$

Update the classifiers and dictionaries by a projected gradient step

$W_s \leftarrow \prod_{W_s}[W_s - \rho_t(\frac{\partial \ell^n}{\partial W_s} + \mu W_s)]$;

$W \leftarrow \prod_{W}[W - \rho_t(\frac{\partial \ell^n}{\partial W} + \mu W)]$;

$D^{(1)} \leftarrow \prod_{D^{(1)}}[D^{(1)} - \rho_t \frac{\partial \ell^n}{\partial D^{(1)}}]$

$D^{(2)} \leftarrow \prod_{D^{(2)}}[D^{(2)} - \rho_t \frac{\partial \ell^n}{\partial D^{(2)}}]$

end for

end for

Part 2: Region Tagging

Input: $\hat{\mathbf{x}}$ (test region)

Output: $\hat{y}$ (predicted tag class)

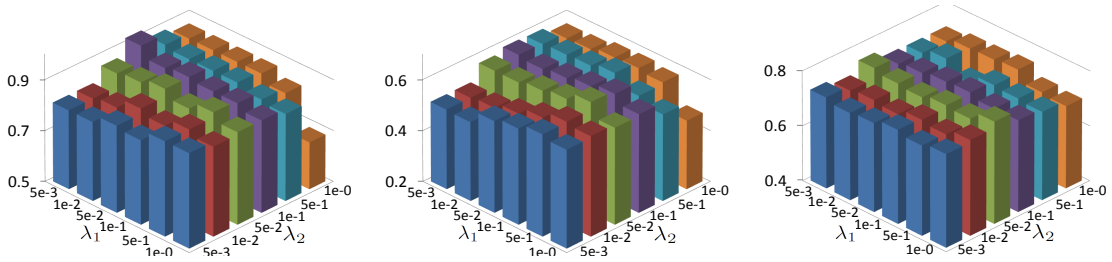
Evaluate the group sparse codes $\hat{\mathbf{z}}^{(1)}$ and $\hat{\mathbf{z}}^{(2)}$ of the test region $\hat{\mathbf{x}}$;

The predicted tag for this test region is $\hat{y} = arg\max_j W(\hat{\mathbf{z}}^{(1)} + \hat{\mathbf{z}}^{(2)})$.

4.3 Experiments

We evaluated our approach for region tagging using several benchmarks, including MSRC-v1, MSRC-v2 [98], and SAIAPR TC-12 datasets [21]. Images in these datasets have been segmented into regions and their ground truth of region masks are also provided. MSRC-v1 contains 240 images that are segmented into 562 regions associated with 13 tags, whereas MSRC-v2 has 591 images and 1482 regions associated with 23 tags. And SAIAPR TC-12 contains 99,535 regions segmented from 20,000 images. The associated 276 tags for this dataset are organized into a hierarchy.

We follow the protocol in [35] to extract RGB color features and sample training and test regions. We use 8 bins for each color channel and count the ratio of pixels whose RGB values fall into each bin to construct a $3D$ histogram. Thus each image region is represented as a 512-dimensional RGB color histogram. For the MSRC-v1 dataset, we randomly sample 200 images and the corresponding regions as the training set, whereas for the MSRC-v2 dataset, 471 images are randomly sampled to form the training set. The remaining regions are used for testing. For SAIAPR TC-12 dataset, we select the same 27 localized tags out of 276 tags as in [35] for evaluation. Then we randomly select 2500 regions whose tags are within the selected subset of 27 tags as the training set and another 500 regions as the test set.



(a) MSRC-v1 (b) MSRC-v2 (c) SAIAPR TC-12

Figure 4.3: The effect of parameters λ_1 and λ_2 on the region tagging performance of our method on three datasets.

4.3.1 Comparing Methods and Parameter Setting

As in [130, 35], we choose LASSO [105], Group LASSO [136] and Sparse Group LASSO [27] as baselines and use the implementation of these methods in SLEP package [65]. We compare our mutli-layer supervised dictionary learning method (MSDL) with two state-of-the-art approaches: SGSC [130], G^{2}SRRT [35]. In order to demonstrate that the super-class level can help improve the accuracy of region tagging, we use single-layer supervised dictionary learning (SSDL) corresponding to the basic-class level as another baseline. The performance of tagging accuracy (number of correctly classified regions over the total test regions) is reported as the average over 5 different trials corresponding to different partitions of training and test sets.

4.3.2 Datasets and Feature Extraction

There are two important parameters in our model: λ_1 and λ_2 that are used to balance the reconstruction error and the sparse penalty for two levels. The ranges of both λ_1 and λ_2 for all datasets are $\{0.005,0.01,0.05,0.1,0.5,1\}$. For other parameters in all experiments, we set the parameters $\nu = 0.1$ and $\mu = 0.001$ for the regularization of sparse codes and classifiers respectively. In addition, the initial learning rate ρ is set to be 0.001 and the level-specific dictionaries are initialized using the software SPAMS [75]. The performance of region tagging by our method with different λ_1 and λ_2 on three datasets are illustrated in Figure 4.3. We see that the highest performance is achieved at different values of the two parameters for the three datasets.

4.3.3 Experimental Results

The accuracies of region tagging using different methods on three datasets are summarized in Table 4.1. We can see that for all the datasets, both SSDL and our method outperform all the other methods. In particular, when compared with other sparse coding-based algorithms, SSDL and our method significantly improve the performance for region tagging on MSRC-v1 dataset—by a margin close to 10% and 20% respectively. This is because the labeled tag distribution in MSRC-v1 is very unbalanced and the tag with most training regions is more likely to be selected for reconstruction of test regions when using the group sparse coding algorithm. On the contrary, both SSDL and our method can reduce the reconstruction error to

Methods	MSRC-v1	MSRC-v2	SAIAPR
Lasso[105]	0.612	0.448	0.652
Group Lasso[136]	0.636	0.458	0.598
Sparse Group Lasso[27]	0.625	0.433	0.561
SGSC[129]	0.726	0.460	-
G^2SRRT(kNN)[35]	0.727	0.473	0.646
G^2SRRT(kNN+Tag)[35]	0.739	0.533	0.667
SSDL	**0.830**	**0.560**	**0.704**
MSDL	**0.926**	**0.634**	**0.772**

Table 4.1: The average accuracies of region tagging by different methods on MSRC-v1, MSRC-v2 and SAIAPR TC-12 datatsets.

some extent by learning a more reconstructive and discriminative dictionary. Furthermore, for the MSRC-v2 and SAIAPR TC-12 datasets, our method improves the tagging accuracy by 10% that is twice than the improvement obtained by SSDL. And this good performance by our method demonstrates that, we effectively explored the semantic relationship among tags and make the super-class level help improve the performance for region tagging. In addition, different from the MSRC datasets, images in the SAIAPR TC-12 dataset are more arbitrary and image regions from the same tag vary drastically; the better performance by our method further demonstrates that our approach can handle the diversity and arbitrariness of image content by exploiting hierarchial relationships among tags. Finally, note that the algorithm SGSC [130] needs to build a spatial kernel for regions within each image,

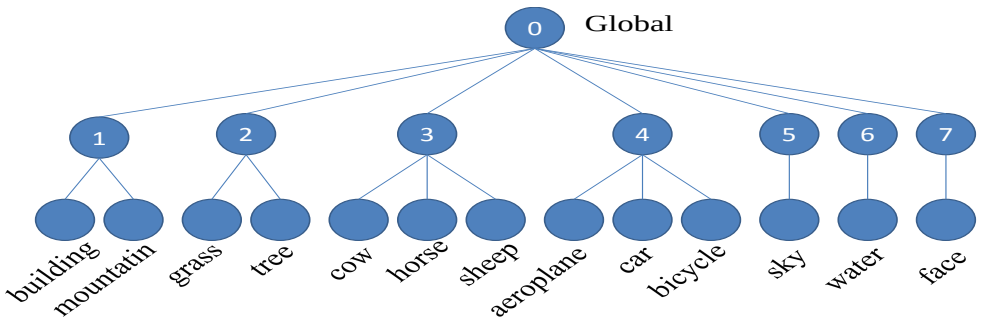


Figure 4.4: The tag taxonomy for MSRC-v1

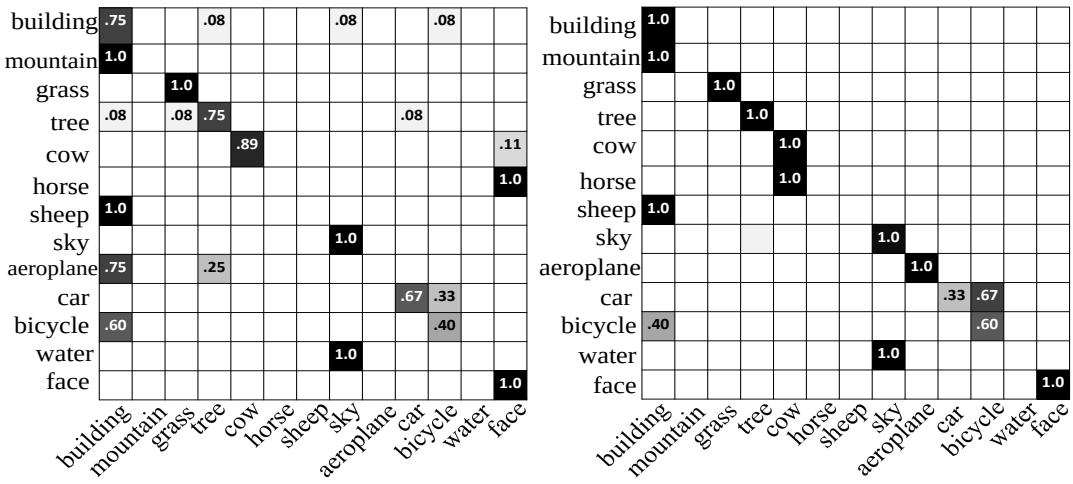


Figure 4.5: Confusion matrices for SSDL (left) and our method MSDL (right) on the MSRC-v1 dataset.

which requires regions within each image to be jointly selected and included in the training and test sets. Since we randomly sampled image regions of the SAIAPR TC-12 dataset and the spatial kernel might not be built, the performance for region tagging by SGSC is not reported in Table 4.1 as in [35].

Figures 4.4 and 4.6 illustrate two tag taxonomies associated with MSRC-v1 and MSRC-v2 respectively while Figures 4.5 and 4.7 display the corresponding confusion matrices obtained by SSDL and our method under the two datasets. Since

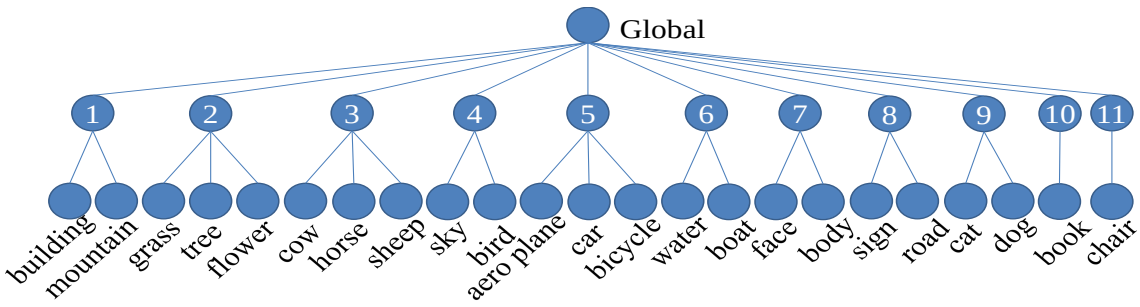


Figure 4.6: The tag taxonomy for MSRC-v2

we obtain similar results in MSRCv1 and MSRCv2 datasets, for simplicity we take MSRC-v1 dataset for analysis. Comparing the confusion matrix obtained by SSDL with our method in Figure 4.5, we can see that tags *building*, *tree*, *cow*, *aeroplane*, *bicycle* have large improvements in tagging accuracy using our proposed method. Moreover, instead of classifying regions from the tag *horse* as *face* by SSDL, our method classifies them as *cow* which is also in the same super-class as *horse*. This demonstrates how our method takes advantages of implicit sharing of sparse codes obtained from the super-class level to help improve the accuracy of tag nodes from the basic-class level. It is also interesting to note that the tag *car* has a slight decrease in tagging accuracy because some regions from *car* are misclassified as *bicycle* which is also in the same-super class. Thus, different tags benefit in different degrees from the implicit sharing of sparse codes and a similar phenomenon has also been observed in [96] which uses a parameter sharing strategy.

Figure 4.9 shows some examples of region tagging results on three datasets. We see that our method correctly classifies those regions that are misclassified by [35] and SSDL. To further investigate the performance of region tagging by SSDL and our method, we select nine tags in each dataset and report the corresponding

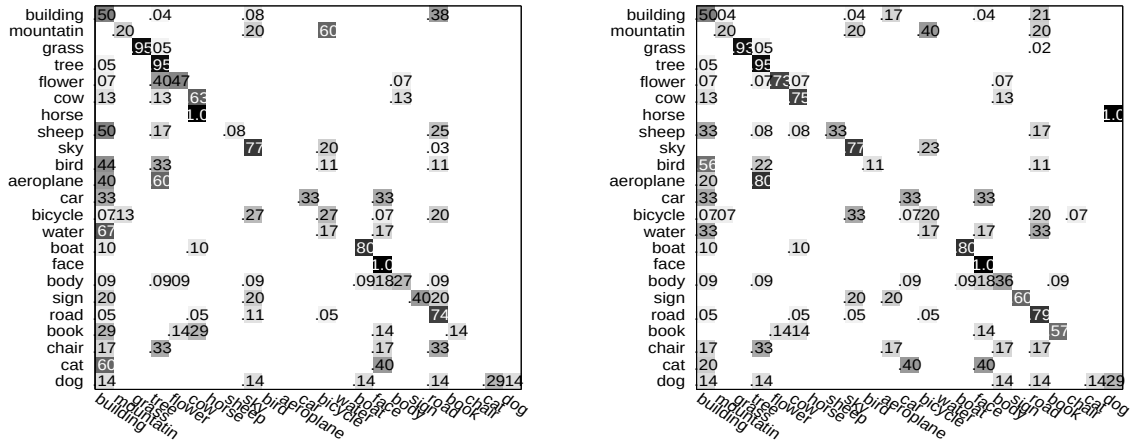


Figure 4.7: Confusion matrices for SSDL (left) and our method MSDL (right) on the MSRC-v2 dataset.

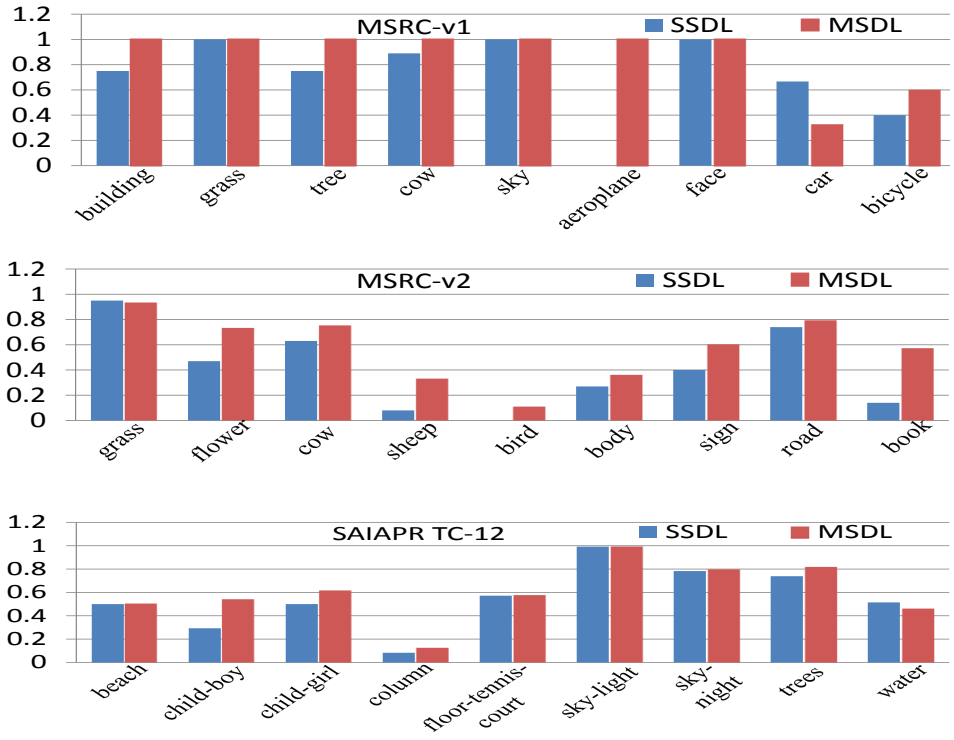


Figure 4.8: The performance comparison using SSDL and MSDL for nine selected tags on each dataset.

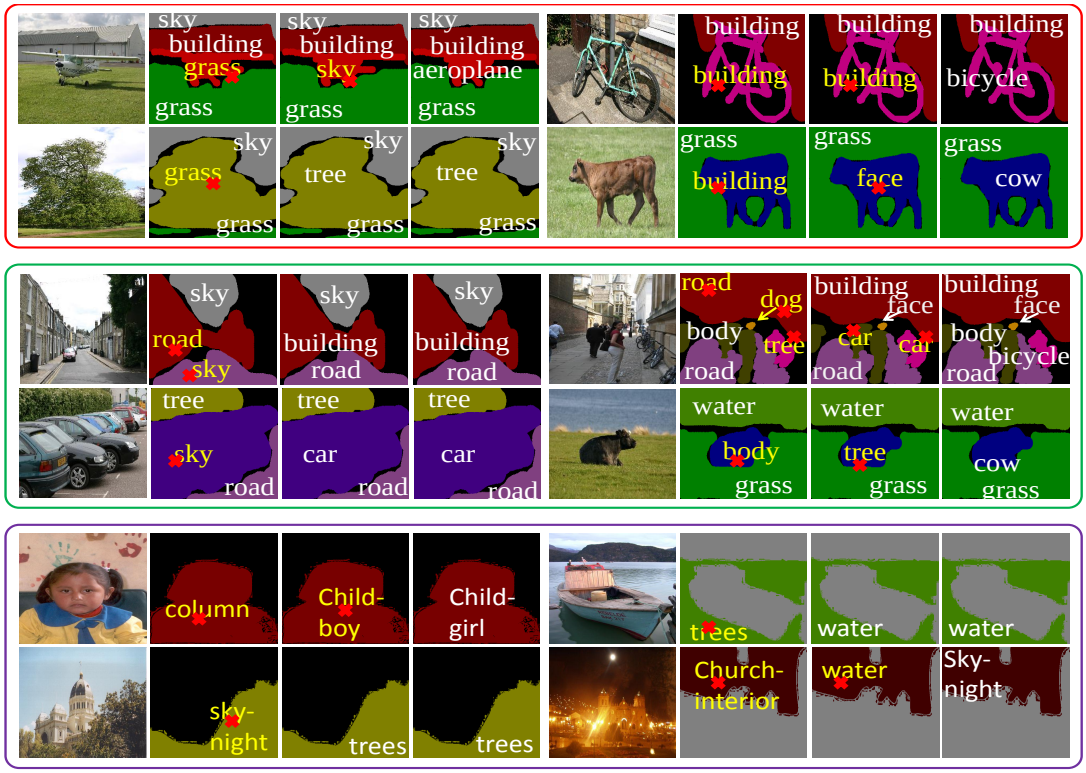


Figure 4.9: Examples of region tagging results on three benchmark image datasets. The subfigures from the top to the bottom corresponds to the MSRC-v1, MSRC-v2 and SAIAPR TC-12 datasets respectively. In each subfigure, the columns from the left to the right correspond to the samples image, region tagging results by [35], our baseline (SSDL) and our method (MSDL). Misclassified tags are in *yellow* while correctly classified tags are in white. The figure is best viewed in color.

tagging accuracy of each tag in Figure 4.8. From the detailed tagging performance, we can see that our method obtains better tagging performance for most of the tags. However, it is also interesting to note that SSDL obtains a slightly better performance for some tags such as *car* in MSRC-v1 dataset and *water* in SAIAPR TC-12 dataset. One possible reason is that the visual appearances of image regions from these tags are very different from other tags within the same super-class which

introduces a negative transfer. Similar facts are also observed in [96].

4.4 Summary

In this chapter, we have proposed a multi-layer hierarchical supervised dictionary learning framework for region tagging by exploring the given tag taxonomy. Specifically, we associate each tag node in the taxonomy with one node-specific dictionary and concatenate the node-specific dictionaries in each level to construct a level-specific dictionary. Using the level-specific dictionary and corresponding level-specific group structure, we obtain level-specific sparse codes that are also close to the *ideal* sparse codes. The sparse codes from different levels are summed up as the final feature representation to learn the level-specific classifier. This enables us to simultaneously take advantages of the robust encoding ability of group sparse coding as well as the semantic relationship in the tag taxonomy. We have extensively tested our approach on three benchmark datasets and results clearly confirm the effectiveness of our approach for region tagging. Although we select region tagging to evaluate our proposed method, we believe that it is a general method and can be developed and applied to object and activity recognition.

Chapter 5: Attribute Learning and Selection for Visual Recognition

5.1 Related Work

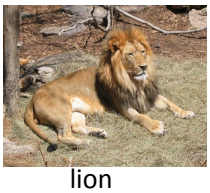
In most traditional approaches for visual recognition, classifiers are trained from patterns of low-level features and corresponding class labels. However, in real-world recognition problems, low-level features can be hardly characterized by a single class label due to large variations within each class. For example, in video-based action recognition, videos of one action class may vary greatly due to large variations in viewpoints, complicated backgrounds, and people performing the actions differently. Conventional low-level features are not able to adequately characterize the rich spatio-temporal structures in action videos. In order to address this problem, multiple high-level semantic concepts called attributes were introduced in [23, 53, 66, 60] to describe the object or action classes. Figure 5.1 shows examples of attributes which describe the object and action classes. For instance, attributes such as "big" and "bush" characterize animal shapes and contextual scenes as shown in Figure 5.1a, while "facing front" and "pushing" describe human poses and spatio-temporal evolution of the action as shown in Figure 5.1b. These attributes are semantically meaningful and interpretable by humans. Since they are relatively robust to changes in viewpoints and scenes, they could bridge the gap

between low-level features and class labels.

Attribute-based representations have proven to be very effective in many computer vision applications. [53, 23] proposed to detect unseen object classes by describing objects using their attributes. [93] learned 20 visual attributes to discover visual relationships among the object categories. [111, 115] jointly modeled the visual attributes and object classes for object naming and localization. [84] modeled relative attributes to generate textural descriptions for new images. [99] presented an approach for ranking and retrieval based on semantic attributes. [52] trained binary classifiers to recognize the presence or absence of describable aspects of visual appearance such as gender and age for face verification. [18] employed a recommender system to select semantic attributes for fine-grained recognition. [10, 85] exploited attributes for classification with human-in-the-loop. Recently, attributes were also employed to improve the performance for action recognition. [131] used attributes and parts for recognizing human actions in still images. [66, 60, 29] introduced different models to learn and exploit attributes for video-based activity recognition.

Even though attribute-based representations appear effective for visual recognition, they require humans to generate a list of attributes that may adequately describe a set of classes. From this list, humans then need to assign the attributes to each class. Previous approaches [66, 60] simply used all the given attributes and ignored the difference in discriminative capabilities among attributes. This caused two major problems. First, a set of human-labeled attributes may not be able to represent and distinguish a set of classes. This is because humans may subjectively

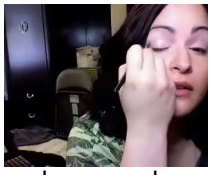
horse



lion

attributes	big	paws	Long-neck	tail	bush
horse	✓	×	✓	✓	×
lion	✓	✓	×	✓	✓

(a) Animals with Attributes dataset



apply eye-makeup

apply lipstick

attributes	indoor	stick like	sharp like	facing front	pushing
apply eye-makeup	✓	✓	✓	✓	×
apply lipstick	✓	✓	✓	✓	×

(b) UCF101 dataset

Figure 5.1: Exemplar images of two classes and their associated attribute sets from the Animals with Attributes dataset and UCF101 dataset.

annotate images or videos with arbitrary attributes. For example, consider the two classes "ApplyEyeMakeup" and "ApplyLipStick" in UCF101 action dataset [102] shown in Figure 5.1b. They have the same set of human-labeled attributes and cannot be distinguished from one another. Second, some manually labeled attributes may be noisy or redundant which leads to degradation in visual recognition performance. In addition, their inclusion also increases the feature extraction time. Thus, it would be beneficial to use a smaller subset of attributes while achieving improved or comparable performance by selecting a set of discriminative and compact attributes.

In order to overcome the first drawback of human-labeled attributes, many methods have been proposed to automatically learn attributes from images or videos. These learned attributes may provide additional discriminative information and are complementary to human-labeled attributes. In particular, [6] proposed to discover and characterize attributes by mining text and image data sampled from the

Internet. [92] used objectness as attributes and also mined part attributes from language resources. [18, 83] made use of human interaction to build both semantic and discriminative attributes. [134] proposed two criteria (the category-separability and learnability) to design discriminative category-level attributes.

Another line of work is to learn different types of mid-level representations to provide additional discrimination capability. These mid-level representations usually identify the occurrence of semantic concepts of interest, such as scene types, actions and objects. [26] constructed mid-level motion features from low-level optical flow features using AdaBoost. [114] learned a global root template and a constellation of several parts to model human actions. [106] used the output of a large number of weakly trained object category classifiers to derive image descriptors. [91] used a max margin framework to learn for discriminative binary codes for representing images. [90] used trajectory clusters as candidates for the parts of an action and assembled these clusters into an action class by graphical modeling. [38] automatically mined discriminative spatio-temporal patches from videos as a new mid-level representation.

In order to overcome the second drawback of human-labeled attributes, many approaches have been proposed to model the relationship between attributes and class labels. [23] exploited semantic and auxiliary discriminative attributes for multi-classification where the discriminative attributes are based on the random splits between one to five classes. [115] jointly modeled class labels and their visual attributes. Specifically, attributes of an object are treated as latent variables and the correlations among attributes are captured in an undirected graphical model

built from training data. [66] modeled attributes as latent variables and searched for the best configuration of attributes for each action using latent SVMs. [60] decomposed a video sequence into short-term segments and characterized segments by the dynamics of their attributes.

We first propose to learn *data-driven* attributes to address the first drawback of human-labeled attributes. We show that data data-driven attributes are complementary to human-labeled attributes. Instead of using clustering-based algorithms to discover data-driven attributes as in [66], we propose a dictionary-based sparse representation method to discover a large data-driven attribute set. Our learned attributes are more suited to represent all the input data points because our method avoids the problem of hard assignment of data points to clusters.

To address the second problem caused by noisy and redundant attributes, we propose to select a compact and discriminative set of attributes from a large set of attributes. Specifically, we first introduce an attribute contribution matrix, where each row represents the discrimination capability of an attribute for differentiating all different pairwise classes. Based on the attribute contribution matrix, we propose three attribute selection criteria for selecting an attribute subset. The first criteria is that the selected attribute subset should provide as much discrimination capability as possible for each pairwise classes. This criteria ensures that the selected attributes are discriminative. The second criteria is that the selected attribute subset should have similar discrimination capability for each pairwise classes. This criteria will balance the discrimination capability obtained by different pairwise classes. In order to achieve the first two criteria, we construct an undirected graph and show that

the selection procedure satisfying the two criteria can be formulated as the entropy rate of a random walk on this graph. The last selection criterion is that the sum of maximum discrimination capability that each pairwise classes can obtain from the selected attributes should be maximized. This criteria will avoid the selection of redundant attributes which can differentiate the same collection of pairwise classes. In other words, one combination of pairwise classes may be repeatedly covered (differentiated) by multiple attributes. It is better to select other attributes which can differentiate uncovered combinations of pairwise classes. We model the last selection criteria as a weighted maximum coverage problem and encourage the selected attribute subset to have a maximum coverage of all pairwise classes. Finally, we integrate the entropy rate term of a random walk and weighted maximum coverage term into the final object function for attribute selection. We demonstrate that the objective function is submodular and present a greedy algorithm which gives a near-optimal solution with a (1-1/e)-approximation bound.

This chapter is organized as follows: Section 2 briefly reviews the concept of submodularity. Section 3 presents the proposed submodular attribute selection approach. Sections 4 introduces the human-labeled attributes and data-driven attributes. Section 5 shows some implementation details and Section 6 provides experimental results and analysis on four public datasets. Section 7 concludes this chapter.

5.2 Submodularity

Submodular functions are a class of set functions that have the the property of *diminishing returns* [79]. Given a set E, a set function $F : 2^E \rightarrow R$ is submodular if $F(A \cup v) - F(A) \geq f(B \cup v) - F(B)$ holds for all $A \subseteq B \subseteq E$ and $v \in E \setminus B$. The diminishing return property means that the marginal gain of the element v decreases if used in a later stage. Recently, submodular functions have been widely exploited in various applications, such as sensor placements [50], superpixel segmentation [70], document summarization [61], object detection and recognition [42, 144] and feature selection [15, 72]. [72] presented a submodular feature selection method for acoustic score spaces based on existing facility location and saturated coverage functions. Krause et al. [49] developed a submodular method for selecting dictionary columns from multiple candidates for sparse representation. Iyer et al. [37] designed a new framework for both unconstrained and constrained submodular function optimization. Streeter et al. [103] proposed an online algorithm for maximizing submodular functions. Different from these approaches, we define a novel submodular objective function for attribute selection. Although we only evaluate our approach for action recognition, it can be applied to other recognition tasks that use attribute descriptions.

Subset	c_1/c_2	c_1/c_3	c_1/c_4	c_2/c_3	c_2/c_4	c_3/c_4
$\mathcal{S}_1$	1	1	1	1	1	1
$\mathcal{S}_2$	2	2	2	2	2	2
$\mathcal{S}_3$	2	1	3	3	1	2

Table 5.1: Vector r corresponding to three different selected subsets. c_i/c_j means class i versus class j.

5.3 Submodular Attribute Selection

In this section, we first introduce the definition of attribute contribution matrix and then propose three attribute selection criteria for selecting a discriminative and compact subset of attributes . In order to satisfy these criteria, we define a submodular function which is a linear combination of the entropy rate of a random walk and a weighted maximum coverage function.

5.3.1 Attribute Selection Criteria

Assume that we have C classes and a large attribute set $\mathcal{P} = \{a_1, a_2, .., a_M\}$ which contains M attributes. The set that includes all combinations of pairwise classes is represented by $\mathcal{U} = \{u_1(1,1), u_2(1,2), ..., u_l(i,j), ..., u_L(C-1,C)\}$ where $u_l(i,j), i < j$ denotes the pairwise combination made up of classes i and j, l is the index of this combination in $\mathcal{U}$, and $L = C \times (C-1)/2$ is the total number of all possible pairwise classes. Here we propose to use the Fisher score to construct an ***attribute contribution matrix*** $A \in \mathbf{R}^{M \times L}$, where an entry $A_{d,l}$ represents

the discrimination capability of attribute a_d for differentiating the class pair (i, j) indexed by $u_l(i, j)$. Specifically, given the attribute a_d and class pair (i, j), let μ_k^d and σ_k^d be the mean and standard deviation of the k-th class and μ^d be the mean of samples from both classes i and j corresponding to the d-th attribute. The Fisher score of attribute a_d for differentiating the class pair (i, j) is computed as follows:

$$A_{d,l(i,j)} = \frac{\sum_{k=i,j} n_k(\mu_k^d - \mu^d)^2}{\sum_{k=i,j} n_k \sigma_k^2} \tag{5.1}$$

where l is the index of pairwise classes (i, j) in $\mathcal{U}$, and n_k is the number of points from class k. Note that different methods can be used to measure the discrimination capability of a_d, such as mutual information and T-test.

Given the attribute contribution matrix A, our goal is to select a subset of attributes denote as $\mathcal{S}$ from the original attributes set $\mathcal{P}$. As mentioned earlier in the introduction, we propose the following three selection criteria to select attributes in the subset $\mathcal{S}$:

- The selected attribute subset should provide as much discrimination capability as possible for each pairwise classes.
- The selected attribute subset should have similar discrimination capability for each pairwise classes.
- The sum of maximum discrimination capability that each pairwise classes can obtain from the selected attributes should be maximized.

Assume that we have already obtained the attributes $\mathcal{S}$ satisfying the above three selection criteria, we can obtain a row vector r from the attribute contribution

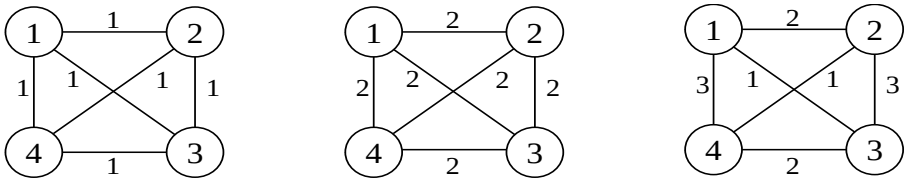
matrix A by summing up its elements from each column that are in rows corresponding to selected attributes $\mathcal{S}$. An example of vector r is shown in Table 5.1. In order to satisfy the above three selection criteria, we would like to have r and A satisfy the following three constraints respetively :

- Each entry of r should be as large as possible.
- The variance of all entries of r should be small.
- The sum of the maximum value of each column in the attribute contribution matrix A should be maximized.

The first constraint explicitly forces each pairwise class to have as much discrimination capability as possible from the selected attribute subset. The second constraint minimizes the variance of all entries of r. This will encourage each pairwise class to have equal or similar discrimination capability. The last constraint will maximize the sum of maximum discrimination capability that each pairwise classes can obtain from the selected attributes should be maximized. The first two constraints can be satisfied by maximizing the entropy rate of a random walk on the proposed graphs. For the third constraint, we will model it as a weighted maximum coverage problem and encourage $\mathcal{S}$ to have a maximum coverage of all pairwise classes.

5.3.2 Entropy Rate-based Attribute Selection

In order to optimize the first two criteria, we need to construct an undirected graph and maximize the entropy rate of a random walk on this graph. We aim to



(a) $\mathcal{S}_1$ (b) $\mathcal{S}_2$ (c) $\mathcal{S}_3$

Figure 5.2: The undirected graphs constructed based on Table 5.1. We show the role of the entropy rate in selecting attributes which have large and similar discrimination capability for each pair of classes. The circles with numbers denote the corresponding class vertices and the numbers next to the edge denote the edge weights, which is a measure of the discrimination capability of selected attribute subset. The self-loops are not displayed. The entropy rate of the graph with large edge weights in (c) has a higher objective value than that of a graph with smaller edge weights in (b). The entropy rate of graph with equal edge weights in (c) has a higher objective value than that of the graph with different edge weights in (d).

obtain a subset $\mathcal{S}$ so that the attribute-based representation has good discrimination power.

Graph Construction: We use $G = (V, E)$ to denote an undirected graph where V is the vertex set, and E is the edge set. The vertex v_i represents class i and the edge $e_{i,j}$ connecting class i and j represents that class i and j can be differentiated by the selected attribute subset $\mathcal{S}$ to some extent. The edge weight for $e_{i,j}$ is defined as $w_{i,j} = \sum_{d \in \mathcal{S}} A_{d,l}$, which represents the discrimination capability of $\mathcal{S}$ for differentiating class i from class j. The edge weights are symmetric, i.e. $w_{i,j} = w_{j,i}$. In addition, we add a self-loop $e_{i,i}$ for each vertex v_i of G. And the weight for self-loop $e_{i,i}$ is defined as $w_{i,i} = \sum_{d \in \mathcal{P} \setminus \mathcal{S}} A_{d,l}$. The total incident weight for each vertex is kept constant so that it produces a stationary distribution for the later proposed random walk on this graph. Note that the addition of these self-loops do not affect the selection of attributes and the graph will change with the selected subset $\mathcal{S}$.

Entropy Rate: We maximize the entropy rate of the random walk on the constructed graph to satisfy the first two selection criteria. The entropy rate quantifies the uncertainty of a stochastic process. Let $X = \{X_t | t \in T, X_t \in V\}$ be a random walk on the graph $G = (V, E)$ with nonnegative discrimination measure w. We use the random walk model from [14] with a transition probability $p_{ij}(\mathcal{S})$ defined as below:

$$p_{i,j}(\mathcal{S}) = \begin{cases} \frac{w_{i,j}}{w_i} = \frac{\sum_{d \in \mathcal{S}} A_{d,l}}{w_i} & \text{if } i \neq j \\ 1 - \frac{\sum_{k:k \neq i} w_{i,k}}{w_i} = \frac{\sum_{d \in \mathcal{P} \setminus \mathcal{S}} A_{d,l}}{w_i} & \text{if } i = j \end{cases} \tag{5.2}$$

where $\mathcal{S}$ is the selected attribute subset and $w_i = \sum_{m:e_{i,m} \in E} w_{i,m}$ is the sum of

incident weights of the vertex v_i including the self-loop. The stationary distribution for this random walk is given by

$$\mu = (\mu_1, \mu_2, ..., \mu_C)^T = (\frac{w_1}{w_0}, \frac{w_2}{w_0}, ..., \frac{w_C}{w_0}) \tag{5.3}$$

where $w_0 = \sum_{i=1}^{C} w_i$ is the sum of the total weights incident on all vertices. It can be verified through $\mu = P^T\mu$ where $P = [p]_{i,j}$ is the transition matrix.

For a stationary 1st-order Markov chain, the entropy rate which measures the uncertainty of the stochastic process X is given by:

$$\begin{aligned} \mathcal{H}(X) &= lim_{t\to\infty} H(X_t|X_{t-1}, X_{t-2}, ..., X_1) \\ &= lim_{t\to\infty} H(X_t|X_{t-1}) \\ &= H(X_2|X_1) \end{aligned} \tag{5.4}$$

The first equality is the definition of the entropy rate of the stationary 1st-order Markov chain, the last two equalities are due to the properties of 1st-order Markov process and stationarity respectively. More details can be found in [14]. Consequently, the entropy rate of the random walk X on our proposed graph $G = (V, E)$ can be written as a set function:

$$\begin{aligned} \mathcal{H}(\mathcal{S}) &= \sum_i u_i H(X_2|X_1 = v_i) \\ &= -\sum_i u_i \sum_j p_{i,j}(\mathcal{S}) log(p_{i,j}(\mathcal{S})) \\ &= -\sum_i \frac{w_i}{w_0} \sum_j \frac{w_{i,j}}{w_i} \log \frac{w_{i,j}}{w_i} \\ &= -\sum_i \sum_j \frac{w_{i,j}}{w_0} \log \frac{w_{i,j}}{w_0} + \sum_i \frac{w_i}{w_0} \log \frac{w_i}{w_0} \end{aligned} \tag{5.5}$$

Intuitively, the maximization of the entropy rate has two consequences. First, it encourages the maximization of $p_{i,j}(\mathcal{S})$ where $i = 1, ..., C$ and $i \neq j$. This can make

edge weights $w_{i,j}, i \neq j$ as large as possible, so class i can be easily differentiated from other classes j (i.e., satisfying the first criteria). Second, it makes all class vertices have transition probabilities similar to other connected class vertices, so the discrimination capabilities of class i from other classes are very similar (i.e., satisfying the second criteria). Maximizing the entropy rate of the random walk on the proposed graph can select a subset of attributes that are compact and discriminative for differentiating all pairwise classes, as shown in Figure 5.2.

Proposition 5.3.1. *The entropy rate of the random walk $\mathcal{H} : 2^M \rightarrow R$ is a submodular function under the proposed graph construction.*

The observation that adding an attribute in a later stage has a lower increase in the uncertainty establishes the submodularity of the entropy rate. This is because at a later stage, the increased edge weights from the added attribute will be shared with attributes which contribute to the differentiation of the same pair of classes. A detailed proof based on [70] is given in the supplementary section.

5.3.3 Weighted Maximum Coverage-based Attribute Selection

We consider a weighted maximum coverage function to achieve the last criteria that the selected subset $\mathcal{S}$ should maximize the coverage of all combinations of pairwise classes. For each attribute a_d, we define a coverage set $\mathcal{U}_d \subseteq \mathcal{U}$ which covers all the combinations of pairwise classes that attribute a_d can differentiate. Meanwhile, for each element (combination) $u_l \in \mathcal{U}$ that is covered by $\mathcal{U}_d$, we define a coverage weight $w(\mathcal{U}_d, u_l) = A_{d,l}$. Given the universe set $\mathcal{U}$ and these coverage

Attrs.	c_1/c_2	c_1/c_3	c_1/c_4	c_2/c_3	c_2/c_4	c_3/c_4
a_1	2	2	0	1	1	0
a_2	1	1	0	0	0	0
a_3	0	0	1	0	0	2
a_4	0	0	0	2	2	0

Table 5.2: Attribute contribution matrix A. c_i/c_j means class i versus class j.

sets $\mathcal{U}_d, d = 1, ..., M$, the weighted maximum coverage problem is to select at most K coverage sets, such that the sum of maximum coverage weight each element can obtain from $\mathcal{S}$ is maximized. The weighted maximum coverage function is defined as follows:

$$\begin{aligned} \mathcal{Q}(\mathcal{S}) &= \sum_{u_l \in U} \max_{d \in \mathcal{S}} w(U_d, u_l) \\ &= \sum_{u_l \in U} \max_{d \in \mathcal{S}} A_{d,l}, \quad \text{s.t. } N_{\mathcal{S}} \leq K \end{aligned} \tag{5.6}$$

where $N_{\mathcal{S}}$ is the number of attributes in $\mathcal{S}$. Note that the weighted maximum coverage problem is reduced to the well studied set-cover problem when all the coverage weights are equal to be ones.

Proposition 5.3.2. *The weighted maximum coverage function $\mathcal{Q} : 2^M \rightarrow R$ is a monotonically increasing submodular function under the proposed set representation.*

For the weighted maximum coverage term, monotonicity is obvious because the addition of any attribute will increase the number of covered elements in $\mathcal{U}$. Submodularity results from the observation that the coverage weights of increased

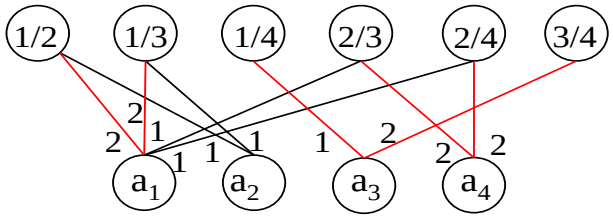


Figure 5.3: The coverage graph constructed based on the Table 5.2. We show the role of weighted maximum coverage term in selecting attributes which have large coverage weights. Two numbers separated by a backslash in the top circles denote a pair of classes, while the bottom circles denote different attributes. The number next to one edge is the coverage weight associated with the class pair when covered by the corresponding attribute. The edge which provides maximum coverage weight for each class pair is in red color. We consider three attribute subsets $\mathcal{S}_1 = \{a_1, a_2\}, \mathcal{S}_2 = \{a_1, a_3\}, \mathcal{S}_3 = \{a_1, a_4\}$. $\mathcal{S}_2$ has a higher objective value than $\mathcal{S}_1$ and $\mathcal{S}_3$ because the sum of maximum coverage weights for all class pairs obtained using attributes from subset $\mathcal{S}_2$ is largest.

covered elements will be less from adding an attribute in a later stage because some elements may be already covered by previously selected attributes. The proof is given in the supplementary section.

5.3.4 Objective Function and Optimization

Combing the entropy rate term and the weighted maximum coverage term, the overall objective function for attribute selection is formulated as follows:

$$\max \mathcal{F}(\mathcal{S}) = \max_{S} \mathcal{H}(\mathcal{S}) + \lambda \mathcal{Q}(\mathcal{S}) \text{ s.t. } N_{\mathcal{S}} \leq K \tag{5.7}$$

where λ controls the relative contribution between the entropy rate and the weighted maximum coverage term. The objective function is submodular because linear combination of two submodular functions with nonnegative coefficients preserves submodularity [79].

Direct maximization of a submodular function is an NP-hard problem. However, a greedy algorithm from [79] gives a near-optimal solution with a $(1 - 1/e)$-approximation bound. The greedy algorithm starts from an empty attribute set $\mathcal{S} = \emptyset$; and iteratively adds one attribute that provides the largest gain for $\mathcal{F}$ at each iteration. The iteration stops when the maximum number of selected attributes is obtained or $\mathcal{F}(\mathcal{S})$ decreases. Algorithm 1 presents the pseudo code of our algorithm. A naive implementation of this algorithm has the complexity of $O(|M|^2)$, because it needs to loop $O(|M|)$ times to add a new attribute and scan through the whole attribute list in each loop. By exploiting the submodularity of the objective function, we use the lazy greedy approach presented in [57] to speed up the optimization process.

Algorithm 4: Submodular Attribute Selection

1: **Input:** $G = (V, E), A$ and λ

2: **Output:** $\mathcal{S}$

3: **Initialization**: $\mathcal{S} \leftarrow \emptyset$

4: **for** $N_{\mathcal{S}} < K$ and $F(S \cup a) - F(S) \geq 0$ **do**

5: $\quad a_m = argmax_{\mathcal{S} \cup a_m} \mathcal{F}(\mathcal{S} \cup \{a_m\}) - \mathcal{F}(\mathcal{S})$

6: $\quad \mathcal{S} \leftarrow a_m$

7: **end for**

5.4 Human-labeled Attribute and Data-driven Attribute Extraction

In this section, we introduce algorithms for the detection of human-labeled attributes and extraction of data-driven attributes.

Visual classes can be characterized by a collection of human-labeled attributes. For example, the action "long-jump" in Olympic Sports Dataset [80] is associated with either the motion attributes (*jump forward, motion in the air*), or with the scene attributes (e.g., *outdoor, track*). Given an instance x, an attribute classifier $f_a : x \rightarrow \{0, 1\}$ predicts the confidence score of the presence of attribute a in the image or video. This classifier f_a is learned using the training samples of all action classes which have this attribute as positive and the rest as negative. Given a set of attribute classifiers $S = \{f_{a_i}(x)\}_{i=1}^{m}$, an instance $x \in R^d$ is mapped to the semantic space $\mathcal{O}$:

$$h : R^d \rightarrow \mathcal{O} = [0, 1]^m \tag{5.8}$$

where $h(x) = (h_1(x), ..., h_m(x))^T$ is a m-dimensional attribute score vector.

Previous works [69, 66] on data-driven attribute discovery used k-means or information theoretic clustering algorithms to obtain the clusters as the learned attributes. We propose to discover a large initial set of data-driven attributes using a dictionary learning method. Specifically, assume that we have a set of N data instances in a n-dimensional feature space $X = [x_1, ..., x_N], x_i \in R^n$, then a data-driven dictionary is learned by solving the following problem:

$$\arg\min_{D,Z} ||X - DZ||_2^2 \ s.t. \ \forall i, \ ||z_i||_0 \leq T \tag{5.9}$$

where $D = [d_1...d_K], d_i \in R^n$ is the learned attribute dictionary of size K, $Z = [z_i...z_N], z_i \in R^K$ are the sparse codes of X, and T specifies the sparsity that each video has fewer than T items in its decomposition. The objective function in (5.9) can be solved using the KSVD algorithm [1]. Since each dictionary atom is treated as a data-driven attribute, an entry z_{ij} in the sparse codes matrix Z is the assigned value for the i-th attribute (dictionary atom) to the j-th instance.

Compared to k-means clustering, this dictionary-based learning scheme avoids the hard assignment of cluster centers to data points. Moreover, it doesn't require the estimation of the probability density function of clusters in information theoretic clustering. Note that our attribute selection framework is very general and different initial attribute extraction methods can be used here.

5.5 Implementation Details

In this section, we provide the implementation details of our approach. The parameter λ is set to be 0.1 throughout the experiments. The effect of λ on the performance of our approach will be presented in the following experiment section.

For the AwA dataset, we followed [53] and used six different feature types: RGB color histograms, SIFT, rgSIFT, PHOG, SURF and local self-similarity histograms. We extracted the color histograms and PHOG feature vectors from 21 cells of a 3-level spatial pyramids ($1 \times 1, 2 \times 2, 4 \times 4$). For the color histograms, we concatenated 128-dimensional color descriptor extracted from each cell to construct a 2688-dimensional feature vector. For PHOG, we extracted 12-dimensional de-

scriptors from each cell and used the same method to construct the final histogram. For other feature types, we extracted 2000-dimensional histograms by using bag-of-words model. We concatenate the histograms from all the feature types to construct 10940-dimensional descriptor histograms.

For the aPascal datase, we followed [23] and used a bag-of-words model to extract features for four feature types: color, texture, visual words and edges. Specifically, color descriptors and texture descriptors are computed for each pixel, densely sampled and quantized to nearest 128 and 256 kmeans centers respectively. Visual words of HOG descriptors are extracted from a spatial pyramid using 8×8 blocks, a 4 pixel step size and 2 scales per octave. The final HOG descriptors are quantized to 1000 kmeans centers. The orientation of edges detected by a Canny edge detector are quantized into 8 signed bins. To encode the information of shapes and locations, we also divided the image into a grid of three vertical and two horizontal blocks, and generate histograms of each feature type for each cells. The final feature histogram is formed by concatenating the descriptors of the four feature types.

For the Olympic Sports datset, we followed the protocol in [66] to extract STIP features [17]. In order to detect interest points for the STIP feature, we applied a 2D Gaussian smoothing filter to video along the spatial dimension, followed by a pair of 1D Gabor filters temporally. Then we detect up to 200 interest points at the local maximum response from each action video. We extract the ST volumes around the interest points and obtain a a 100-dimensional gradient-based descriptors via PCA. Following [66], these interest points-based descriptors are further quantized into 2000 visual words by k-mean clustering and each action video is represented by a

2000-dimensional histogram.

For the UCF101 dataset, we compute the improved version of dense trajectories in [112] and extract three types of descriptors for each trajectory: histogram of oriented gradients (HOG), histogram of optical flow (HOF) and motion boundary histogram (MBH). HOG captures the static appearance information while HOF and MBH encode motion information by using optical flow. The three types of descriptors are normalized and concatenated to form the the trajectory descriptor. We use Fisher vector encoding [86] to obtain 101,376-dimensional histogram to represent each action video.

We consider three sets of attributes: human-labeled attribute set (**HLA set**), data-driven attribute set (**DDA set**) and the set mixing both types of attributes (**Mixed set**). For each human-labeled attribute, the original high dimensional features are used to learn the classifiers for predicting the presence of human-labeled attributes. In order to learn data-driven attributes, we first reduce the dimension of the original features by using the principle component analysis (PCA), and then learn a dictionary from the features of reduced dimension using the KSVD algorithm [1]. Each dictionary atom is treated as an attribute, and the sparse code with respect to this dictionary atom is treated as the attribute value, indicating the presence (or selection) of the associated dictionary atom for reconstruction. For the **Mixed** set, we concatenate the prediction scores of human-labeled attributes and the sparse codes associated with data-driven attributes to construct the new feature representations on this set.

To demonstrate the effectiveness of our selection framework, we compare the

result using the selected subset with the result based on the initial set. After attribute selection, prediction scores of selected human-labeled attributes from the **HLA** set are concatenated and normalized to form the new features for evaluation. Whereas for the **DDA** set, the sub-dictionary made up of the selected dictionary atoms (data-driven attributes) will be used to obtain new sparse representations for evaluation. For the **Mixed** set, we use the similar strategies to obtain new features based on the selected human-labeled and data-driven attribute subsets respectively. For all the attribute-based representations, a nonlinear SVM with a Gaussian or sigmoid kernel is trained for classifying unlabeled test data. The parameters C and γ are chosen from $\{0.1, 0.5, 1, 5, 10\}$.

We also compare our method with two other submodular approaches based on the *facility location* function (FL) and *saturated coverage* function (SC) discussed in [72]. These objective functions are defined as follows:

$$\mathcal{F}_{fa}(\mathcal{S}) = \sum_{i \in V} \max_{j \in S} w_{i,j}$$

$$\mathcal{F}_{sa}(\mathcal{S}) = \sum_{i \in V} \min\{C_i(\mathcal{S}), \alpha C_i(\mathcal{V})\}$$

where $w_{i,j}$ is a similarity between attribute i and j, $C_i(\mathcal{S}) = \sum_{j \in \mathcal{S}} w_{i,j}$ measures the degree that attribute i is "covered" by $\mathcal{S}$ and α is a hyperparameter that determines a global saturation threshold. For the two approaches compared against, we consider an undirected k-nearest neighbor graph and use a Gaussian kernel to compute pairwise similarities $w_{i,j} = \exp(-\beta d_{i,j}^2)$ where $d_{i,j}$ is the distance between attribute i and j, $\beta = (2\langle d_{i,j}^2 \rangle)^{-1}$ and $\langle \cdot \rangle$ denotes expectation over all pairwise distances. The value of k ranges from 5 to 10 for all the four datasets.

5.6 Experiments

In this section, we validate our method for both object and action recognition. We used the Animals with Attributes (AwA) dataset and aPscal Dataset introduced in [53] for object recognition, Olympic Sports dataset [80] and UCF101 [66] dataset for action recognition. For each dataset, we compare the result of the proposed approach with two other submodular selection methods on the **HLA**, **DDA** and **Mixed** sets respectively. Meanwhile, we compare the performance of attribute-based representation with several state-of-the-art approaches on the four datasets.

5.6.1 Object Recognition

5.6.1.1 Animal with Attributes Dataset

The Animal with Attributes (AwA) dataset [53] contains 30,475 images of 50 animal categories. The images are collected by querying four large internet search engines, Google, Microsoft, Yahoo and Flicker using the animal names as keywords. Associated with images, there exist 85 human labeled attributes. Figure 5.1a shows examples of some classes with the values of exemplary attributes assigned to this class.

To demonstrate that attributes-based representation does improve object recognition performance, we followed [134] to evaluate our approach for multi-class classification on 40 *known* categories of AwA dataset. We select different number of training images $K = 15, 20, 25, 30, 50$ per category as training data, 25 images per

methods	15	20	25	30	50
FL	21.6	24.4	28.2	28.7	37.0
SC	20.6	23.9	26.7	28.5	37.4
our method	**22.6**	**25.5**	**29.1**	**30.1**	**38.7**

Table 5.3: Recognition accuracy on the AwA dataset using human-labeled attributes.

category as test data, and 10 images per category for validation.

For each different combination of training and test data, we construct three attribute-based representations as follows: (1) **HLA set**: For each human-labeled attribute, we train a non-linear Support Vector Machine (SVM) classifier combined with the same kernel, which is the sum of individual χ^2 kernels for each feature type. For two D-dimensional feature vectors $x, y \in \mathbf{R}^D$, the χ^2 kernel is defined to be $k(x,y) = exp(-\gamma\chi^2(x,y))$ where $\chi^2(x,y) = \sum_{i=1}^{D} \frac{(x_i - y_i)^2}{x_i + y_i}$, the bandwidth parameter γ is set to be the five times inverse of median of the χ^2-distances over the training samples. We concatenate confidence scores from all these attribute classifiers into a 85-dimensional vector to represent this image. (2) **DDA set** : For data-driven attributes, we first apply PCA to reduce the dimension of histogram descriptors to be $D = 550, 750, 950, 1150, 1950$ respectively when the number of training images is $K = 15, 20, 25, 30, 50$ respectively. Then by using the KSVD algorithm [1], we learn a dictionary of different size which is forty times of the number training images per category. (3) **Mixed set**: The attribute set is made up of the combination of both **HLA set** and **DDA set**.

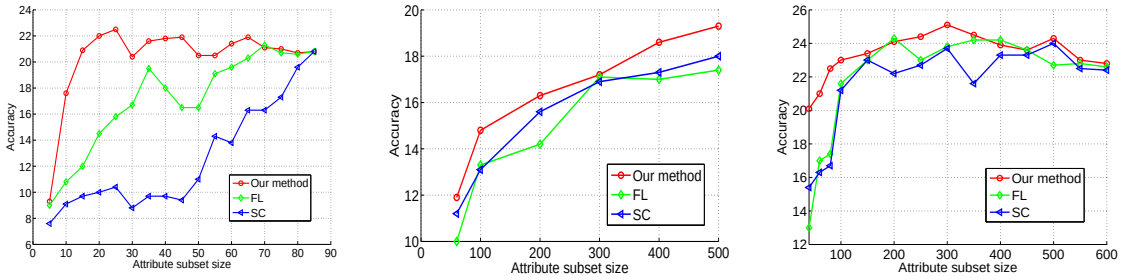
methods	15	20	25	30	50
FL [72]	17.4	21.2	20.4	21.6	25.1
SC [72]	18.0	21.3	21.9	22.5	25.9
our method	**19.3**	**22.7**	**23.6**	**24.1**	**27.1**

Table 5.4: Recognition accuracy on the AwA dataset using data-driven attributes.

methods	15	20	25	30	50
FL [72]	24.3	27.7	29.3	31.4	38.7
SC [72]	22.1	25.3	27.8	29.7	38.2
our method	**25.1**	**28.0**	**30.8**	**32.1**	**39.6**

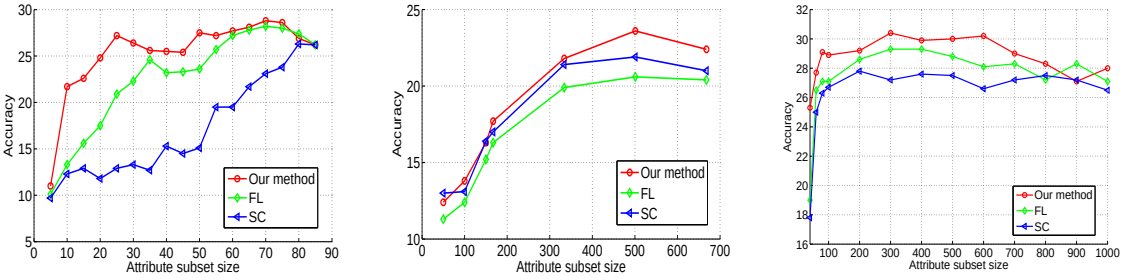
Table 5.5: Recognition accuracy on the AwA dataset using the mixed attribute set.

Tables 5.3, 5.4 and 5.5 show classification accuracies of attribute subsets selected by different submodular selection methods on the **HLA**, **DDA** and **Mixed** sets respectively. It can be seen that SC [72] outperforms FL [72] on the human-labeled attribute sets, but perform worse on the data-driven attribute sets. However, our method consistently yields a better performance than the other two submodular selection methods on all the three different attribute sets. This is because the attributes selected by our method have large and similar discrimination capability for differentiating pairwise classes, while the attributes selected by other two methods have large similarity to other attributes. It is also observed that the performance of all the three submodular selection methods increases as the number of training images per category increases. In addition, different approaches on the **HLA** set



(a) **HLA** set (b) **DDA** set (c) **Mixed** set

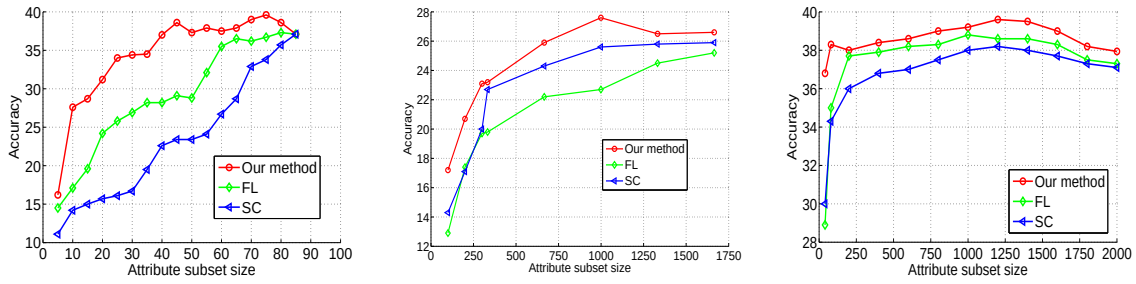
Figure 5.4: Recognition results by different submodular methods on the AwA dataset. The number of training images per category is 15.



(a) **HLA** set (b) **DDA** set (c) **Mixed** set

Figure 5.5: Recognition results by different submodular methods on the AwA datset. The number of training images percategory is 25.

perform better than on the **DDA** set. One possible reason is that we used PCA to reduce the dimension of the high dimensional features and lost some useful or discriminative information. Finally, we note that different approaches achieve the best performance on the **Mixed** set which combines both human-labeled attributes and data-driven attributes. This demonstrates that data-driven attributes are complementary to human-labeled attributes and can help improve the performance of visual recognition.



(a) **HLA** set (b) **DDA** set (c) **Mixed** set

Figure 5.6: Recognition results by different submodular methods on the AwA datset. The number of training images percategory is 50.

Figures 5.4, 5.5 and 5.6 present classification accuracies of attribute subsets of different sizes when the number of training images per category is 15, 25 and 25 respectively. When the number of training images per category is 15, only 15 attributes out of 85 human-labeled attributes selected by the proposed method achieves comparable performance of the total **HLA** set. However, the attribute subsets selected by other two submodular methods did not improve the performance of the total **HLA** set. When the number of training images per category is 25 or 50, only half of the 85 human-labeled attributes selected by the proposed method yield comparable or better performance of the total **HLA** set. We also found that the first selected 85 attributes out of the **Mixed** set by the proposed method are always human-labeled attributes. This is because the human-labeled attributes are more discriminative than data-driven attributes.

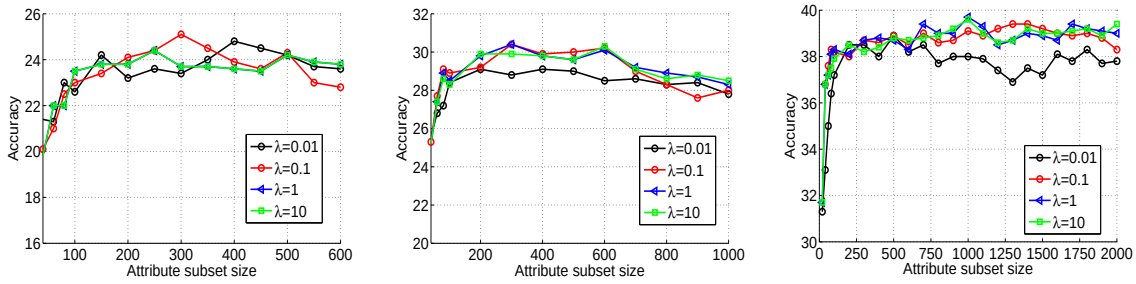
We also compare our approach with several state-of-the-art approaches on this dataset: (1) low-level features, on which a SVM classifier is trained. (2) Classemes [106], which used the output of a large number of weakly trained ob-

methods	15	20	25	30	50
low-level features	22.6	22.7	24.4	25.0	27.0
CLA [134]	21.0	22.0	24.5	26.0	29.5
LDA [8]	22.6	23.4	27.0	30.4	31.8
classemes [106]	23.8	26.0	27.6	30.4	32.2
[28]	29.0	29.2	29.6	31.0	33.3
our method	**25.1**	**28.0**	**30.8**	**32.1**	**39.6**

Table 5.6: Recognition accuracy of different comparing methods on the AwA dataset.

ject category classifiers as attributes. (3) Category-level attribute designing approach (CLA) [134], which designed discriminative category-level attributes. (4) LDA-based attribute learning approach [8], which automatically learned attributes for each object class by using latent dirichlet allocation. (5) [28], which mined visual prototypes of attributes by clustering with Gaussian mixtures from multi-scale salient areas in noisy Web images.

Table 5.6 shows the results of different approaches. We observe that attribute-based representations obtained by different approaches can achieve higher accuracy against the low-level-feature-based approach. It can also be seen that our approach consistently outperforms CLA[134], LDA[8] and Classemes[106], which demonstrates the effectiveness of the proposed attribute selection approach that can select discriminative and compact attribute subset from the original noisy and redundant set. In addition, the proposed approach achieves comparable recognition accuracy to [28]



(a) $K = 15$ (b) $K = 25$ (c) $K = 50$

Figure 5.7: The effect of λ on the performance of the proposed approach on the AwA dataset when the number of training images percategory is 15, 25 and 50 respectively.

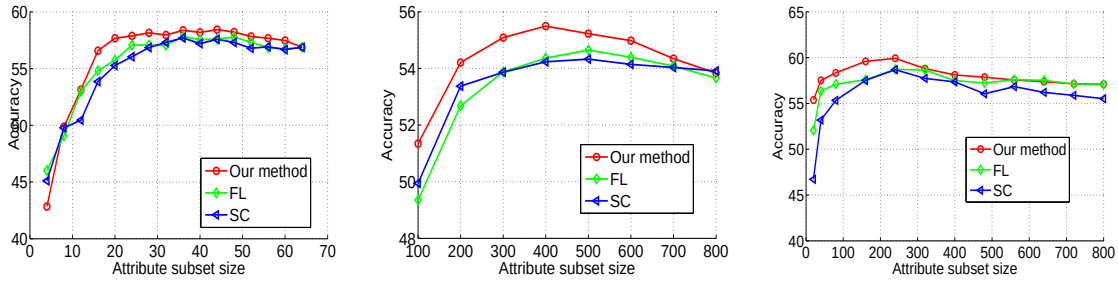
when the number of training images per category is less than 25, but surpasses it when the number of training images per category is larger than 25.

Figure 5.7 shows the performance curves for a range of λ. We observe that when λ is larger than 0.1, our approach obtains similar performance for different values of λ.

5.6.1.2 aPascal Dataset

The aPascal dataset introduced in [23] consists of a subset of 12,695 images from 20 classes selected from the PASCAL VOC 2008 dataset. Attributes are annotated on the image level and each image is annotated with 64 binary attributes. These attributes characterize shape, material and presence of important parts of the visual object.

In order to evaluate our approach for multi-class classification, we follow [23] to use the Pascal training set as the training set and the Pascal validation set as the test



(a) **HLA** set (b) **DDA** set (c) **Mixed** set

Figure 5.8: Recognition results by different submodular methods on the aPascal dataset.

set. Three attribute-based representations are constructed as follows: (1) **HLA set**: For each human-labeled attribute, we trained a linear SVM and the parameter C was chosen from {0.001, 0.01, 0.1, 1, 10, 100, 200}. We concatenate confidence scores from all these attribute classifiers into a 64-dimensional vector to represent this image. (2) **DDA set**: For data-driven attributes, we first apply PCA to reduce the dimension of histogram descriptors to be 3000, and learn a dictionary of size 800 from all video features using KSVD [1]. Each video is represented by a 800-dimensional sparse coefficient vector. (3) **Mixed set**: This attribute set is obtained by combining **HLA set** and **DDA set**. Figure 5.8 shows classification accuracies of attribute subsets selected by different submodular slection methods. It can be observed that our approach consistently outperform other two methods, which demonstrates that the attributes selected by our approach are more discriminative.

Tables 5.7 shows classification accuracies of attribute subsets selected by the different submodular selection methods on the **HLA**, **DDA** and **Mixed** sets respectively. It can be seen that the attribute subsets selected by the different submodular

Attributes set	ALL	Subset(FL)	Subset(SC)	ours
HLA	56.1	57.5	57.8	58.5
DDA	53.5	54.6	54.3	55.4
Mixed	57.0	58.7	58.6	59.9

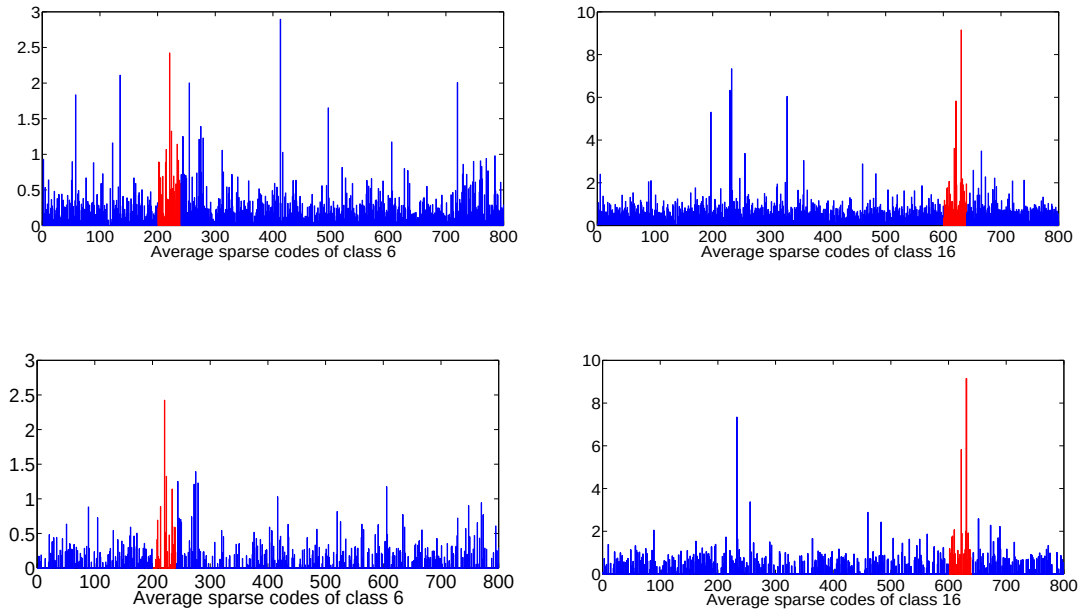
Table 5.7: Recognition results of different attribute-based representations. "All" denotes the original attribute sets and "Subset" denote the selected subsets.

Method	logistic regression [23]	SVM [23]	latent space [2]	ours
HLA	53.4	58.3	59.6	59.9

Table 5.8: Recognition results of different approaches.

methods outperform the initial attribute set. The proposed method consistently yields a better performance than the other two submodular selection methods on all the three different attribute sets.

We also compare our approach with several state-of-the-art approaches on this dataset: (1) a classifier trained using logistic regression [23]. (2) a linear SVM classifier [23]. Note that logistic regression and SVM [23] not only used the semantic attributes, but also used another type of discriminative attributes proposed in [23]. (3) latent space [2], which used partial least squares to find a suitable latent attribute space to learn the semantic attributes. Table 5.8 shows the comparison result of different approaches. We observe that the proposed approach perform better than other comparing approaches, which validates the effectiveness of the proposed submodular attribute selection method.



(a) Before selection (b) After selection

Figure 5.9: Sparse codes of class 6 and 16 before and after selection respectively. (a)The sparse codes in red correspond to the sub-dictionary D_6. (b) The sparse codes in read correspond to the sub-dictionary D_{16}.

For data-driven attributes, we visualize the attribute values (or sparse codes) of test samples from two different classes in Figure 5.9. Specifically, the learned attribute dictionary D is made up of a set of class-specific dictionaries, i.e. $D = [D_1, D_2, ..., D_K]$, where D_k is the sub-dictionary corresponding to class k. The sparse codes corresponding to the k-th sub-dictionary is most discriminative for differentiating class k from other classes. Before selection, we sum up the absolute value of sparse codes of test samples from each class. After the selection, we keep the sparse codes corresponding to selected dictionary atoms and set the remaining sparse codes to be zeros. It can be seen that the subset of discriminative sparse codes (in red) are mostly kept after selection, while some noisy and redundant sparse codes are removed. In addition, we normalize the sparse codes of each class to have a sum of one. Since the normalized sparse codes can been seen as a distribution function, we calculate the entropy of normalized sparse codes before and after selection respectively. And we found that the entropy decreases after selection, which demonstrates that the sparse codes are more discriminative after selection.

5.6.2 Action Recognition

5.6.2.1 Olympic Sports Dataset

The Olympic Sports dataset [80] contains 783 YouTube video clips of athletes practicing different sports. It has 16 sports activities which includes high jump, long jump, triple jump, pole vault, discus throw, hammer throw, javelin throw, shot put, basketball lay-up, bowling, tennis serve, platform diving, springboard diving,

(a) Olympic Sports dataset (b) UCF101 dataset

Figure 5.10: Exemplar frames of four action classes from the Olympic Sports dataset and UCF101 dataset respectively.

snatch, clean and jerk, gymnastic vault. Figure 5.10a shows exemplar frames of four action classes. We use 40 human-labeled attributes provided by [66].

Three attribute-based representations are constructed as follows: (1) **HLA set**: For each human-labeled attribute, we train a binary SVM with a histogram intersection kernel. We concatenate confidence scores from all these attribute classifiers into a 40-dimensional vector to represent this video. (2) **DDA set**: For data-driven attributes, we learn a dictionary of size 457 from all video features using KSVD [1] and each video is represented by a 457-dimensional sparse coefficient vector. (3) **Mixed set**: This attribute set is obtained by combining **HLA set** and **DDA set**.

We compare the performance of features based on selected attributes with those based on the initial attribute set. For all the different attribute-based features, we use an SVM with Gaussian kernel for classification. Table 5.9 shows clas-

dataset	**HLA**		**DDA**		**Mixed**	
	All	**Subset**	**All**	**Subset**	**All**	**Subset**
Olympic	61.8	**64.1**	49.0	**53.8**	63.1	**66.7**
UCF101	81.7	**83.4**	79.0	**81.6**	82.3	**85.2**

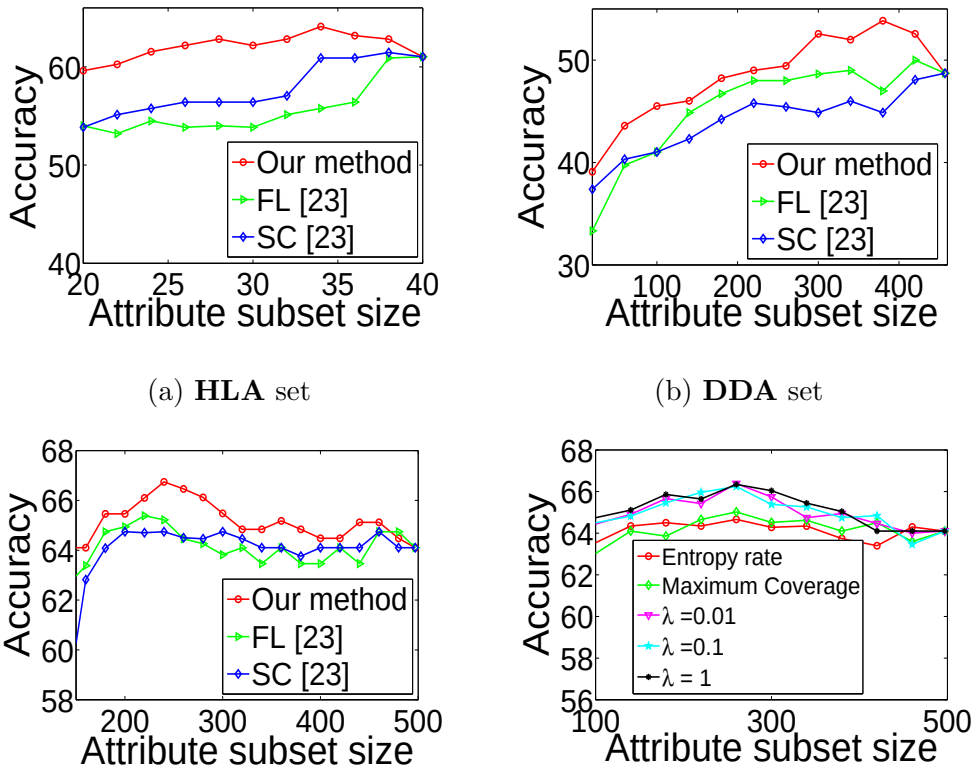
Table 5.9: Recognition results of different attribute-based representations. "All" denotes the original attribute sets and "Subset" denote the selected subsets.

sification accuracies of different attribute-based representations. Compared with the initial attribute set, the selected attributes have greatly improved the classification accuracy, which demonstrates the effectiveness of the proposed method for selecting a subset of discriminative attributes. Moreover, features based on the **Mixed** set outperform features based on either **HLA** set or **DDA** set. This shows that data-driven attributes are complementary to human-labeled attributes and together they offer a better description of actions.

Table 5.10 shows the per-category average precision (AP) and mean AP of different approaches. It can be seen that the proposed method achieves the best performance. This illustrates the benefits of selecting discriminative attributes and removing noisy and redundant attributes. Note that our method outperforms the method that is most similar to ours [66] which uses complex latent SVMs to combine low-level features, human-labeled attributes and data-driven attributes. Moreover, compared with other dynamic classifiers [80, 60] which account for the dynamics of bag-of-features or action attributes, our method still obtains comparable results. This is because the provided human-labeled attributes are very noisy and they

Activity	[54]	[80]	[104]	[66]	[60]	**HLA**	**DDA**	**Mixed**
high-jump	52.4	68.9	18.4	**93.2**	82.2	80.4	66.4	83.1
long-jump	66.8	74.8	81.8	82.6	92.5	88.8	85.3	**93.9**
triple-jump	36.1	52.3	16.1	48.3	52.1	61.4	60.7	**73.6**
pole-vault	47.8	82.0	**84.9**	74.4	79.4	55.1	45.5	56.8
gym. vault	88.6	86.1	85.7	86.7	83.4	98.2	84.2	**98.4**
short-put	56.2	62.1	43.3	**76.2**	70.3	63.7	39.5	72.2
snatch	41.8	69.2	**88.6**	71.6	72.7	74.5	34.2	79.8
clean-jerk	83.2	84.1	78.2	79.4	85.1	73.8	57.9	**82.6**
javelin throw	61.1	74.6	79.5	62.1	**87.5**	36.0	26.4	36.5
hammer throw	65.1	77.5	70.5	65.5	74.0	76.9	77.2	**80.4**
discuss throw	37.4	58.5	48.9	**68.9**	57.0	53.9	45.6	56.0
diving-plat.	91.5	87.2	93.7	77.5	86.0	94.8	55.3	**99.2**
diving-sp. bd.	80.7	77.2	79.3	65.2	78.3	79.7	59.7	**90.4**
bask. layup	75.8	77.9	85.5	66.7	78.1	88.7	89.7	**90.7**
bowling	66.7	**72.7**	64.3	72.0	52.5	43.0	55.3	55.4
tennis-serve	39.6	49.1	49.6	55.2	38.7	78.8	35.3	**83.7**
mean-AP	62.0	72.1	66.8	71.6	73.2	72.1	57.2	**77.0**

Table 5.10: Average precisions for activity recognition on the Olympic Sporst dataset.



(a) **HLA** set

(b) **DDA** set

(c) **Mixed** set

(d) Effect of λ in **Mixed** set

Figure 5.11: Recognition results by different submodular methods on the Olympic Sports dataset.

can greatly affect the training of latent SVM and representation of the attribute dynamics.

Figures 5.11a 5.11b 5.11c show classification accuracies of attribute subsets selected by different submodular selection methods. It can be seen that our method outperforms the other two submodular selection methods for the three different attribute sets. This is because our method prefers attributes with large and similar discrimination capability for differentiating pairwise classes, while the other two methods prefer attributes with large similarity to other attributes (i.e. repre-

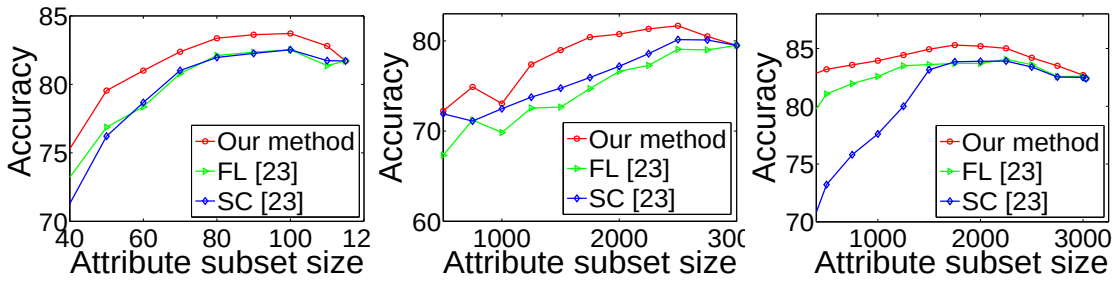
sentative), without explicitly considering the discrimination capabilities of selected attributes. Figure 5.11d shows the performance curves for a range of λ. We observe that the combination of entropy rate term and maximum coverage term obtains a higher classification accuracy than when only one of them is used. In addition, our approach is insensitive to the selection of λ on the Olympic Sports dataset.

5.6.2.2 UCF101 Dataset

UCF101 dataset contains over 10,000 video clips from 101 different human action categories. Figure 5.10b shows exemplar frames of four action classes.

Three different attribute sets and corresponding attribute-based representations are constructed as follows: (1) **HLA set**: Due to the high dimensionality of features and large number of samples, the linear SVM is trained for the detection of each human-labeled attribute. We concatenate confidence scores from all these attribute classifiers into a 115-dimensional vector to represent a video. (2) **DDA set**: For data-driven attributes, we first apply PCA to reduce the dimension of histogram descriptors to be 3300 and then learn a dictionary of size 3030. The features based on data-driven attributes are 3030-dimensional sparse coefficient vectors. (3) **Mixed set**: **HLA set** plus **DDA set**.

Following the training and testing dataset partitions proposed in [102], we train a linear SVM and report classification accuracies of different attribute-based representations in Table 5.9. The selected attribute subset outperforms the initial attribute set again which demonstrates the effectiveness of our proposed attribute



(a) **HLA** set (b) **DDA** set (c) **Mixed** set

Figure 5.12: Recognition results by different submodular methods on UCF101 dataset.

selection method. Figure 5.12 shows the results of attribute subsets selected by different submodular selection methods. Note that this dataset is highly challenging because the training and test videos of the same action have different backgrounds and actors. It can be seen that our method still substantially outperforms the other two submodular methods. This is because some redundant attributes dominated the selection process and the attributes selected by approaches in the comparison group had very unbalanced discrimination capability for different classes. However, the attributes selected by the proposed method have strong and similar discrimination capability for each class.

Table 5.11 presents the classification accuracies of several state-of-the-art approaches on this dataset. Our method achieves comparable results to the best result 85.9% from [112] which uses complex spatio-temporal pyramids to embed structure information in features. Note that our method also outperforms other methods which make use of complicated and advanced feature extraction and encoding techniques.

splits	[112]	[122]	[143]	[44]	[94]	**HLA**	**DDA**	**Mixed**
1	83.03	83.11	79.41	65.22	63.41	82.45	80.35	**84.19**
2	84.22	84.60	81.25	65.39	65.37	83.27	82.16	**85.51**
3	84.80	84.23	82.03	67.24	64.12	84.60	82.42	**86.30**
Avg	84.02	83.98	80.90	65.95	64.30	83.44	81.64	**85.24**

Table 5.11: Recognition results of different approaches on UCF101 dataset.

5.7 Summary

We exploited human-labeled attributes and data-driven attributes for improving the performance of both object and action recognition algorithms. We first presented three attribute selection criteria for the selection of discriminative and compact attributes. Then we formulated the selection procedure as one of optimizing a submodular function based on the entropy rate of a random walk and weighted maximum coverage function. Our selected attributes not only have strong and similar discrimination capability for all pairwise classes, but also maximize the sum of largest discrimination capability that each pairwise classes can obtain from the selected attributes. Experimental results on four challenging dataset show that the proposed method significantly outperforms many state-of-the art approaches.

Our approach has two limitations that need to be addressed. First, the data-driven attributes are learned independently from the human-labeled attributes, it is possible that some of the learned data-driven attributes are redundant and can not help improve the performance of visual recognition. One possible future work

includes extending our approach to model the relationship between human-labeled attributes and data-driven attributes, such that the learned data-driven attributes should further reduce the confusion among classes given the human-labeled attributes. Second, our our approach only exploits the linear relationship between attributes in the entropy rate term, and the first-order relationship in the weighted maximum coverage term. Another possible future work is to model and exploit high-order relationship among attributes for improving the performance of visual recognition.

Chapter 6: Directions for Future Work

In this chapter, we outline several potential directions in which the problems addressed in this dissertation can be explored further.

6.1 Grassmman Manifold-based Domain Adaptation

In the manifold-based approach [33], only the source and target data are available and we generated intermediate representations by sampling along the geodesic that connects the source and target domains. It is not clear why these intermediate representations could help decrease the mismatch between the two domains and improve the cross-domain classification task. We would like to validate the quality of these intermediate representations. Given a subset of the real and intermediate samples, corresponding to domain shifts that lie between source and target domains, we will develop subspace-based representations from them to evaluate the fidelity of the intermediate data synthetically generated by sampling the geodesic. The real and intermediate samples can also be used to regularize the construction of geodesic-based intermediate representations. Figure 6.1 shows the difference between subspaces obtained from intermediate samples and subspaces sampled from the geodesic on the Grassmann manifold.

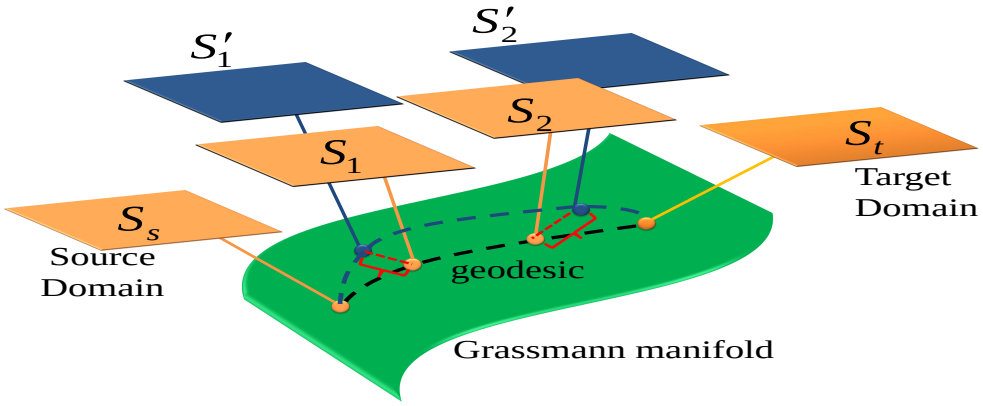


Figure 6.1: **Synthetic intermediate representations versus real intermediate representations.** We apply PCA in the source and target domains and obtain two subspaces S_s, S_t. intermediate subspace S_1 and S_2 are obtained by sampling along the geodesic connecting the source and target domains. Since intermediate samples are available, we can obtain the intermediate subspace S_1', S_2' by applying PCA similarly. S_1, S_2 are called synthetic intermediate subspaces while S_1', S_2' are called real intermediate subspaces.

6.2 Measures of Domain Shifts

We will investigate measures to characterize the nature and type of domain shift so that appropriate adaptation methods can be developed and evaluated. For example, pose variations correspond to geometric domain shifts, while appearance shifts due to illumination variations provide photometric domain shifts. We will integrate physical models to handle domain shifts due to pose and illumination variations. Statistical models will be developed to address domain shifts due to occlusions as these could be random. We will develop principled methods to predict the adaptability of one domain to another. Public data sets, such as the CMU PIE

and CMU MultiPIE data, which consists of faces at different poses illuminated by multiple sources, can be the initial data sets that will enable this investigation.

6.3 Vision applications

Our previous efforts have focused on object recognition, face recognition and activity recognition problems. We will continue to evaluate the effectiveness of domain adaptation methods for object recognition problems in unconstrained conditions. We will also investigate domain adaptation methods for other computer vision algorithms, such as object detection and tracking. Object detection algorithms require adaptation to objects and background clutter.

6.4 Hierarchical Latent Domain Adaptation

Large-scale image classification systems that are able to identify objects among thousands of possible labels are receiving significant attention in recent years. However, we are often confronted with the situation that the test data only covers a semantically related subset of all the objects whereas the training data contains millions of samples from all the objects. This means that the training data and test data have different label space and the label space of the test data are a subset of that of training data. On the one hand, the general classifiers trained using all the available training data is not optimal to the specific test tasks. On the other hand, it is inefficient and suboptimal to retrain the classifiers whenever a test task is given. We will focus on optimally adapting the general classifiers to specific task as shown

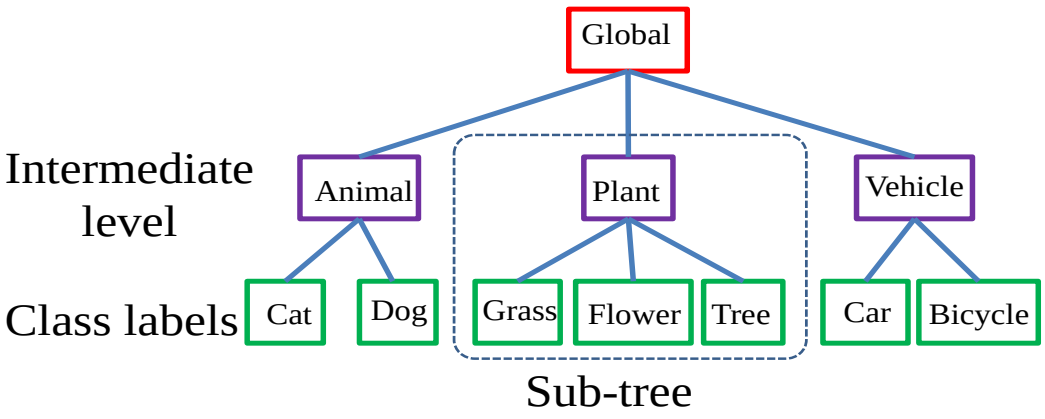


Figure 6.2: Hierarchical latent domain adaptation. General classifiers are trained over all the class labels in the bottom. The specific test task which are implicitly given as a set of image queries come from a semantically related subset of all the class labels. The goal is to adapt the general classifiers to specific test task.

in Figure 6.2.

A probabilistic model that jointly identifies the underlying test task and performs prediction with a linear-time probabilistic inference algorithm was proposed by [40]. However, this generative model has the following disadvantages:(1)Tree structure of category labels is used for task discovery only but not for training of the general classifiers. (2) It is not reasonable to model latent task space using the Erland prior. (3) Each category label is treated equally during the label refinement and the similarity of sibling nodes is not exploited. Future work will focus on deriving discriminative model to adapt the general classifiers to specific tasks.

Appendix A: Appendix

Here we give proofs of proposition 5.3.1 and 5.3.2 in Chapter 5.

A.1 Proof of Submodularity of Entropy Rate

Recall our definition of $\mathcal{H}(\mathcal{S})$:

$$\mathcal{H}(\mathcal{S}) = -\sum_i u_i \sum_j p_{i,j}(\mathcal{S}) log(p_{i,j}(\mathcal{S})) \tag{A.1}$$

where u_i is the stationary probability of v_i in the stationary distribution and $p_{i,j}(\mathcal{S})$ is the transition probability from v_i to v_j with respect to $\mathcal{S}$. T

Proof. We prove the submodularity by showing

$$\mathcal{H}(\mathcal{S} \cup \{a_1\}) - \mathcal{H}(\mathcal{S}) \geq \mathcal{H}(\mathcal{S} \cup \{a_1, a_2\}) - \mathcal{H}(\mathcal{S} \cup \{a_2\}). \tag{A.2}$$

It is known that the transition probability with respect to $\mathcal{S}$ is given as follows:

$$p_{i,j}(\mathcal{S}) = \begin{cases} \frac{w_{i,j}}{w_i} = \frac{\sum_{d \in \mathcal{S}} A_{d,l}}{w_i} & \text{if } i \neq j \\ \frac{w_{i,i}}{w_i} = \frac{\sum_{d \in \mathcal{P} \setminus \mathcal{S}} A_{d,l}}{w_i} & \text{if } i = j \end{cases} \tag{A.3}$$

where $w_i = \sum_{m:e_{i,m} \in E} w_{i,m}$ is the sum of incident weights of the vertex v_i and $w_{i,i} = w_i - \sum_{j \neq i} w_{i,j}$, l is the index of the combination of pairwise classes (i, j) in $\mathcal{U}$. Without loss of generality, we assume that after the addition of attribute a_n into

$\mathcal{S}$, the transition probability becomes

$$p_{i,j}(\mathcal{S} \cup \{a_1\}) = \begin{cases} \frac{w_{i,j}}{w_i} + \frac{A_{n,l}}{w_i} & \text{if } i \neq j \\ \frac{w_{i,i}}{w_i} - \frac{\sum_{j \neq i} A_{n,l}}{w_i} & \text{if } i = j. \end{cases} \tag{A.4}$$

For simplicity of notation, we let $p_{i,j}(\mathcal{S}) = p_{i,j}$ and $p_{i,j}(\mathcal{S} \cup \{a_n\}) = p_{i,j} + \Delta^n_{i,j}, n = 1, 2,$ where $\Delta^n_{i,j}$ is symmetric,i.e. $\Delta^n_{i,j} = \Delta^n_{j,i}$. We note that $\Delta^n_{i,j \neq i} \geq 0$ and $\Delta^n_{i,i} = -\sum_{j \neq i} \Delta^n_{i,j} \leq 0$. $\Delta^n_{i,j} = 0$ means that the addition of a_n does not increase the edge weight $e_{i,j}$ while $\Delta^n_{i,j} > 0$ means that the addition of a_n increase $w_{i,j}$. Similarly, we let $p_{i,j}(\mathcal{S} \cup \{a_1, a_2\}) = p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j}$.

$$\mathcal{H}(\mathcal{S} \cup \{a_1\}) - \mathcal{H}(\mathcal{S}) \tag{A.5}$$

$$= -\sum_i u_i \sum_j (p_{i,j} + \Delta_{i,j}) \log((p_{i,j} + \Delta_{i,j}) + \sum_i u_i \sum_j p_{i,j} \log p_{i,j} \tag{A.6}$$

$$= -\sum_i \sum_j \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} \log(p_{i,j} + \Delta_{i,j}) - \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} \log \frac{w_i}{w_0} \tag{A.7}$$

$$+ \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i}{w_0} + \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log p_{i,j} \tag{A.8}$$

$$= -\sum_i \sum_j \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} + \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} \tag{A.9}$$

$$= -\sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} + \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} \tag{A.10}$$

$$- \sum_i \sum_j \frac{w_i \Delta_{i,j}}{w_0} \log \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} \tag{A.11}$$

Now we prove the first two terms and the las term are larger than zeros respectively.

$$- \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0} + \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} \tag{A.12}$$

$$= \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{\frac{w_i p_{i,j}}{w_0}}{\frac{w_i(p_{i,j} + \Delta_{i,j})}{w_0}} \tag{A.13}$$

$$\geq \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{\sum_i \sum_j \frac{w_i p_{i,j}}{w_0}}{\sum_i \sum_j \frac{w_i (p_{i,j}+\Delta_{i,j})}{w_0}} \tag{A.14}$$

$$= \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log 1 = 0 \tag{A.15}$$

by the definition of transition probability $\sum_j (p_{i,j} + \Delta_{i,j}) = \sum_j p_{i,j} = 1$ and the *Log-sum inequality* stated as follows.

Proposition A.1.1. *(Log-sum inequality) For non-negative numbers* $a_1, a_2, ..., a_n$ *and* $b_1, b_2, ..., b_n$

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq (\sum_{i=1}^{n} a_i) \log \frac{\sum_{i=1}^{n} a_i}{sum_{n=1}^{n} b_i} \tag{A.16}$$

with equality if and only if $\frac{a_i}{b_i}$ *= constant.*

$$- \sum_i \sum_j \frac{w_i \Delta_{i,j}}{w_0} \log \frac{w_i (p_{i,j} + \Delta_{i,j})}{w_0} \tag{A.17}$$

$$= - \sum_i \sum_{j \neq i} \frac{w_i \Delta_{i,j}}{w_0} \log \frac{w_i (p_{i,j} + \Delta_{i,j})}{w_0} - \sum_i \frac{w_i \Delta_{i,i}}{w_0} \log \frac{w_i (p_{i,i} + \Delta_{i,i})}{w_0} \tag{A.18}$$

$$= - \sum_i \sum_{j \neq i} \frac{w_i \Delta_{i,j}}{w_0} \log \frac{w_i (p_{i,j} + \Delta_{i,j})}{w_0} + \sum_i \sum_{j \neq i} \frac{w_i \Delta_{i,j}}{w_0} \log \frac{w_i (p_{i,i} + \Delta_{i,i})}{w_0} \tag{A.19}$$

$$= \sum_i \sum_{j \neq i} \frac{w_i \Delta_{i,j}}{w_0} \log \frac{p_{i,i} + \Delta_{i,i}}{p_{i,j} + \Delta_{i,j}} \tag{A.20}$$

A.1.1 Submodularity

Proof. We prove the submodularity by showing

$$\mathcal{H}(\mathcal{S} \cup \{a_1\}) - \mathcal{H}(\mathcal{S}) \geq \mathcal{H}(\mathcal{S} \cup \{a_1, a_2\}) - \mathcal{H}(\mathcal{S} \cup \{a_2\}). \tag{A.21}$$

Similarly, for simplicity of notation, we let $p_{i,j}(\mathcal{S} \cup \{a_1\}) = p_{i,j} + \Delta^1_{i,j}$ and

$p_{i,j}(\mathcal{S} \cup \{a_1, a_2\}) = p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j}$.

$$\mathcal{H}(\mathcal{S} \cup \{a_1\}) - \mathcal{H}(\mathcal{S}) - \mathcal{H}(\mathcal{S} \cup \{a_1, a_2\}) + \mathcal{H}(\mathcal{S} \cup \{a_2\}) \tag{A.22}$$

$$= -\sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} + \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} \tag{A.23}$$

$$+ \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.24}$$

$$- \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.25}$$

$$= -\sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \tag{A.26}$$

$$+ \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.27}$$

$$+ \sum_i \sum_j \frac{w_i \Delta^2_{i,j}}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.28}$$

$$- \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.29}$$

$$+ \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} \tag{A.30}$$

$$= -\sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \tag{A.31}$$

$$+ \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.32}$$

$$+ \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.33}$$

$$- \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \log \frac{w_i(p_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.34}$$

$$+ \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i p_{i,j}}{w_0} - \sum_i \sum_j \frac{w_i p_{i,j}}{w_0} \log \frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0} \tag{A.35}$$

$$= \sum_i \sum_j \frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0} \log \frac{\frac{w_i(p_{i,j} + \Delta^1_{i,j} + \Delta^2_{i,j})}{w_0}}{\frac{w_i(p_{i,j} + \Delta^1_{i,j})}{w_0}} \tag{A.36}$$

$$+\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^2_{i,j})}{w_0}\log\frac{\frac{w_i(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})}{w_0}}{\frac{w_i(p_{i,j}+\Delta^2_{i,j})}{w_0}} \tag{A.37}$$

$$+\sum_i\sum_j \frac{w_i p_{i,j}}{w_0}\log\frac{\frac{w_i p_{i,j}}{w_0}}{\frac{w_i(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})}{w_0}} \tag{A.38}$$

$$\geq \sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^1_{i,j})}{w_0}\log\frac{\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})}{w_0}}{\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^1_{i,j})}{w_0}} \tag{A.39}$$

$$+\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^2_{i,j})}{w_0}\log\frac{\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})}{w_0}}{\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^2_{i,j})}{w_0}} \tag{A.40}$$

$$+\sum_i\sum_j \frac{w_i p_{i,j}}{w_0}\log\frac{\sum_i\sum_j \frac{w_i p_{i,j}}{w_0}}{\frac{\sum_i\sum_j w_i(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})}{w_0}} \tag{A.41}$$

$$=\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^1_{i,j})}{w_0}\log 1+\sum_i\sum_j \frac{w_i(p_{i,j}+\Delta^2_{i,j})}{w_0}\log 1 \tag{A.42}$$

$$+\sum_i\sum_j \frac{w_i p_{i,j}}{w_0}\log 1 \tag{A.43}$$

$$=0. \tag{A.44}$$

by the definition of the transition probability

$$\sum_j p_{i,j}=\sum_j(p_{i,j}+\Delta^1_{i,j})=\sum_j(p_{i,j}+\Delta^1_{i,j}+\Delta^2_{i,j})=1 \tag{A.45}$$

A.2 Proof of Monotonically Increasing Submodularity of Coverage Term

The proof contains two parts. The first part proves $\mathcal{Q}(\mathcal{S})$ is monotonically increasing. In the second part, we show that $\mathcal{Q}(\mathcal{S})$ is submodular.

A.2.1 Proof of the monotonically increasing property

Proof. Let $\mathcal{S}$ be a subset of attributes and $a_1 \in \mathcal{P}$ be any attribute. We prove the monotonically increasing property

$$\mathcal{Q}(\mathcal{S} \cup \{a_1\}) - \mathcal{Q}(\mathcal{S}) \geq 0. \tag{A.46}$$

$$\mathcal{Q}(\mathcal{S} \cup \{a_1\}) - \mathcal{Q}(\mathcal{S}) = \sum_{u_l \in \mathcal{U}} \max_{d \in \mathcal{S} \cup \{a_1\}} A_{d,l} - \sum_{u_l \in \mathcal{U}} \max_{d \in \mathcal{S}} A_{d,l} \tag{A.47}$$

$$= \sum_{u_l \in \mathcal{U}} [\max(\max_{d \in \mathcal{S}} A_{d,l}, A_{1,l}) - \max_{d \in \mathcal{S}} A_{d,l}] \geq 0 \tag{A.48}$$

A.2.2 Proof of the submodularity

Proof. We prove the submodularity by showing

$$\mathcal{Q}(\mathcal{S} \cup \{a_1\}) - \mathcal{Q}(\mathcal{S}) \geq \mathcal{Q}(\mathcal{S} \cup \{a_1, a_2\}) - \mathcal{Q}(\mathcal{S} \cup \{a_2\}). \tag{A.49}$$

.

$$\mathcal{Q}(\mathcal{S} \cup \{a_1\}) - \mathcal{Q}(\mathcal{S}) \geq \mathcal{Q}(\mathcal{S} \cup \{a_1, a_2\}) - \mathcal{Q}(\mathcal{S} \cup \{a_2\}) \tag{A.50}$$

$$= \sum_{u_l \in \mathcal{U}} [\max(\max_{d \in \mathcal{S}} A_{d,l}, A_{1,l}) - \max_{d \in \mathcal{S}} A_{d,l} - \max(\max_{d \in \mathcal{S}} A_{d,l}, A_{1,l}, A_{2,l}) \tag{A.51}$$

$$+ \max(\max_{d \in \mathcal{S}} A_{d,l}, A_{2,l})]. \tag{A.52}$$

Depending on which term from the three terms $\max_{d \in \mathcal{S}} A_{d,l}$, $A_{1,l}$ and $A_{2,l}$ is largest, we consider three cases and prove that

$$\mathcal{Q}_l = \max(\max_{d \in \mathcal{S}} A_{d,l}, A_{1,l}) - \max_{d \in \mathcal{S}} A_{d,l} - \max(\max_{d \in \mathcal{S}} A_{d,l}, A_{1,l}, A_{2,l}) + \max(\max_{d \in \mathcal{S}} A_{d,l}, A_{2,l}) \geq 0 \tag{A.53}$$

for given $u_l \in \mathcal{U}$.

Case 1: Assume that $\max_{d\in\mathcal{S}} A_{d,l}$ is the largest, i.e. $\max_{d\in\mathcal{S}} A_{d,l} \geq A_{1,l}, \max_{d\in\mathcal{S}} A_{d,l} \geq A_{2,l}$, then

$$Q_l = \max_{d\in\mathcal{S}} A_{d,l} - \max_{d\in\mathcal{S}} A_{d,l} - \max_{d\in\mathcal{S}} A_{d,l} + \max_{d\in\mathcal{S}} A_{d,l} = 0. \tag{A.54}$$

Case 2: Assume that $A_{1,l}$ is the largest, i.e. $A_{1,l} \geq \max_{d\in\mathcal{S}} A_{d,l}, A_{1,l} \geq \max_{d\in\mathcal{S}}$, then

$$Q_l = A_{1,l} - \max_{d\in\mathcal{S}} A_{d,l} - A_{1,l} + \max_{d\in\mathcal{S}} A_{d,l} \tag{A.55}$$

$$= \max(\max_{d\in\mathcal{S}} A_{d,l}, A_{2,l}) - \max_{d\in\mathcal{S}} A_{d,l} \geq 0. \tag{A.56}$$

Case 3: Assume that $A_{2,l}$ is the largest, i.e. $A_{2,l} \geq \max_{d\in\mathcal{S}} A_{d,l}, A_{2,l} \geq \max_{d\in\mathcal{S}}$, then

$$Q_l = \max(\max_{d\in\mathcal{S}} A_{d,l}, A_{1,l}) - \max_{d\in\mathcal{S}} A_{d,l} - A_{2,l} + A_{2,l} \tag{A.57}$$

$$= \max(\max_{d\in\mathcal{S}} A_{d,l}, A_{1,l}) - \max_{d\in\mathcal{S}} A_{d,l} \geq 0. \tag{A.58}$$

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K -SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 2006.

[2] Z. Al-Halah, T. Gehrig, and R. Stiefelhagen. Learning semantic attributes via a common latent space. In *International Conference on Computer Vision Theory and Applications*, 2014.

[3] L. T. Alessandro Bergamo. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010.

[4] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):288–303, 2010.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. Springer, 2006.

[6] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *IEEE International Conference on Computer Vision*, volume 2, pages 1395–1402. IEEE, 2005.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[9] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[10] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.

[11] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011.

[12] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse learning. In *UAI*, 2011.

[13] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *Computer Vision and Pattern Recognition Workshops, 2008.*, pages 1–8. IEEE, 2008.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[15] A. Das, A. Dasgupta, and R. Kumar. Selecting diverse features via spectral regularization. In *NIPS*, 2012.

[16] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[17] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[18] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3474–3481. IEEE, 2012.

[19] L. Duan, I. W.-H. Tsang, D. Xu, and S. J. Maybank. Domain transfer SVM for video concept detection. In *CVPR*, 2009.

[20] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[21] H. J. Escalante, C. A. Hernández, J. A. González, A. López-López, M. M. y Gómez, E. F. Morales, L. E. Sucar, L. V. Pineda, and M. Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer Vision and Image Understanding*, 2010.

[22] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *ACM Multimedia*, 2004.

[23] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[24] A. Farhadi and M. K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008.

[25] A. Farhadi, M. K. Tabrizi, I. Endres, and D. A. Forsyth. A latent model of discriminative aspect. In *ICCV*, 2009.

[26] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[27] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. Technical Report arXiv, 2010.

[28] J. Fu, J. Wang, X.-J. Wang, Y. Rui, and H. Lu. What visual attributes characterize an object class? In *Asian Conference on Computer Vision*, 2014.

[29] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *Computer Vision–ECCV 2012*, pages 530–543. Springer, 2012.

[30] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *IEEE Workshop on Statistical Signal Processing*, pages 315 – 318, 2003.

[31] S. Gao, L.-T. Chia, and I. W.-H. Tsang. Multi-layer group sparse coding - for concurrent image classification and annotation. In *CVPR*, 2011.

[32] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[33] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[34] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, 2008.

[35] Y. Han, F. Wu, J. Shao, Q. Tian, and Y. Zhuang. Graph-guided sparse reconstruction for region tagging. In *CVPR*, 2012.

[36] C.-H. Huang, Y.-R. Yeh, and Y.-C. F. Wang. Recognizing actions across cameras by exploring the correlated subspace. In *ECCV Workshops*, 2012.

[37] R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *ICML*, 2013.

[38] A. Jain, A. Gupta, M. Rodriguez, and L. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013.

[39] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 2011.

[40] Y. Jia and T. Darrell. Latent task adaptation with large-scale hierarchies. In *ICCV*, 2013.

[41] W. Jiang, E. Zavesky, S.-F. Chang, and A. C. Loui. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, 2008.

[42] Z. Jiang and L. S. Davis. Submodular salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.

[43] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011.

[44] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. In *PAMI*, 2013.

[45] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE Trans. Pattern Anal. Mach. Intell.*

[46] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008.

[47] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, 2008.

[48] S. Kong and D. Wang. A dictionary learning approach for classification: Separating the particularity and the commonality. In *ECCV*, 2012.

[49] A. Krause and V. Cevher. Submodular dictionary selection for sparse representation. In *ICML*, 2010.

[50] A. Krause, A. Singh, C. Guestrin, and C. Williams. Near-optimal sensor placements in gaussian processes. In *ICML*, 2005.

[51] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[52] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):1962–1977, 2011.

[53] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[54] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[55] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition*, 2008.

[56] I. Laptev and P. Pérez. Retrieving actions in movies. In *International Conference on Computer Vision*, 2007.

[57] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[58] B. Li, O. I. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *CVPR*, 2012.

[59] R. Li and T. Zickler. Discriminative virtual views for cross-view action recognition. In *CVPR*, 2012.

[60] W. Li and N. Vasconcelos. Recognizing activities by attribute dynamics. In *NIPS*, 2012.

[61] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of ACL*, 2011.

[62] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[63] D. Liu, S. Yan, Y. Rui, and H.-J. Zhang. Unified tag analysis with multi-edge graph. In *ACM Multimedia*, 2010.

[64] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[65] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[66] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011.

[67] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[68] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[69] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *CVPR*, 2009.

[70] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy-rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. In *PAMI*, 2014.

[71] X. Liu, B. Cheng, S. Yan, J. Tang, T.-S. Chua, and H. Jin. Label to region by bi-layer sparsity priors. In *ACM Multimedia*, 2009.

[72] Y. Liu, K. Wei, K. Kirchhoff, Y. Song, and J. Bilmes. Submodular feature selection for high-dimensional acoustic score spaces. In *ICASSP*, 2013.

[73] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[74] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

[75] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009.

[76] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2008.

[77] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *CVPR*, 2007.

[78] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *ECCV*, 2008.

[79] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 1978.

[80] J. C. Niebles, C. wei Chen, and L. Fei-fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[81] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*.

[82] V. Parameswaran and R. Chellappa. Human action-recognition using mutual invariants. *CVIU*, 2005.

[83] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1681–1688. IEEE, 2011.

[84] D. Parikh and K. Grauman. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 503–510. IEEE, 2011.

[85] A. Parkash and D. Parikh. Attributes for classifier feedback. In *Computer Vision–ECCV 2012*, pages 354–368. Springer, 2012.

[86] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[87] D.-S. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, 2008.

[88] S. Ramagiri, R. Kavi, and V. Kulathumani. Real-time multi-view human action recognition using a wireless camera network. In *Distributed Smart Cameras (ICDSC), 2011 Fifth ACM/IEEE International Conference on*. IEEE, 2011.

[89] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 2002.

[90] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.

[91] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *Computer Vision–ECCV 2012*, pages 876–889. Springer, 2012.

[92] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 910–917. IEEE, 2010.

[93] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *Trends and Topics in Computer Vision*, pages 1–14. Springer, 2012.

[94] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[95] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[96] R. Salakhutdinov, A. Torralba, and J. B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.

[97] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *International Conference on Multimedia*, 2007.

[98] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 2009.

[99] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 801–808. IEEE, 2011.

[100] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *TPAMI*, 25(12):1615 – 1618, December 2003.

[101] S. Singh, S. A. Velastin, and H. Ragheb. Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, 2010.

[102] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[103] M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, 2008.

[104] K. D. Tang, F.-F. Li, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[105] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1994.

[106] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Computer Vision–ECCV 2010*, pages 776–789. Springer, 2010.

[107] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008.

[108] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*.

[109] P. K. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *TPAMI*, 33(11):2273–2286, 2011.

[110] A. ul Haq, I. Gondal, and M. Murshed. On dynamic scene geometry for view-invariant action matching. In *CVPR*, 2011.

[111] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 537–544. IEEE, 2009.

[112] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[113] J. Wang, H. Zheng, J. Gao, and J. Cen. Cross-view action recognition based on a statistical translation framework. In *IEEE Transactions on Circuits and Systems for Video Technolgoy*, 2014.

[114] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[115] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *Computer Vision–ECCV 2010*, pages 155–168. Springer, 2010.

[116] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, 2007.

[117] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010.

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

[119] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Computer Vision–ECCV 2008*, pages 650–663. Springer, 2008.

[120] H. Wold. Partial least squares. *International Journal of Cardiology*, 147(2):581–591, 1985.

[121] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[122] J. Wu, Y. Zhang, and W. Lin. Towards good practices for action video encoding. In *ICCV*, 2013.

[123] X. Wu and Y. Jia. View-invariant action recognition using latent kernelized structural svm. In *Computer Vision–ECCV 2012*. Springer, 2012.

[124] P. Yan, S. M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008.

[125] C. Yang, M. Dong, and F. Fotouhi. Region based image annotation through multiple-instance learning. In *ACM Multimedia*, 2005.

[126] C. Yang, M. Dong, and J. Hua. Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In *CVPR*, 2006.

[127] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.

[128] J. Yang and M.-H. Yang. Top-down visual saliency via joint crf and dictionary learning. In *CVPR*, 2012.

[129] J. Yang, K. Yu, and T. S. Huang. Supervised translation-invariant sparse coding. In *CVPR*, 2010.

[130] Y. Yang, Y. Yang, Z. Huang, H. T. Shen, and F. Nie. Tag localization with spatial correlations and joint group sparsity. In *CVPR*, 2011.

[131] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1331–1338. IEEE, 2011.

[132] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.

[133] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009.

[134] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 771–778. IEEE, 2013.

[135] J. Yuan, J. Li, and B. Zhang. Exploiting spatial context constraints for automatic image region annotation. In *ACM Multimedia*, 2007.

[136] M. Yuan, M. Yuan, Y. Lin, and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 2006.

[137] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, 2010.

[138] J. Zheng and Z. Jiang. Learning view-invariant and sparse representations for cross-view action recognition. In *ICCV*, 2013.

[139] J. Zheng and Z. Jiang. Tag taxonomy aware dictionary learning for region tagging. In *CVPR*, 2013.

[140] J. Zheng, Z. Jiang, R. Chellappa, and P. J. Phillips. Submodular attribute selection for action recognition in video. In *Advances in Neural Information Processing Systems*, 2014.

[141] J. Zheng, Z. Jiang, J. Phillips, and R. Chellappa. Cross-view action recognition via a transferable dictionary pair. In *BMVC*, 2012.

[142] J. Zheng, M.-Y. Liu, R. Chellappa, and J. P. M.-H. Yang. A grassmann manifold-based domain adaptation approach. In *ICPR*, 2012.

[143] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *ICCV*, 2013.

[144] Z. J. Zhu, Fan and L. Shao. Submodular object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.