UNDERSTANDING HUMAN ACTIONS IN STILL IMAGES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Bangpeng Yao
August 2013

This dissertation is online at: http://purl.stanford.edu/hb303pj9151

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Fei-Fei Li, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Daphne Koller

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Percy Liang

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Silvio Savarese

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Many human actions, such as "playing violin" and "taking a photo", can be well described by still images, because of the specific spatial relationship between humans and objects, as well as the specific human and object poses involved in these actions. Recognizing human actions in still images will potentially provide useful information in image indexing and visual search, since a large proportion of available images contain people. Progress on action recognition is also beneficial to object and scene understanding, given the frequent human-object and human-scene interactions. Further, as video processing algorithms often rely on some form of initialization from individual video frames, understanding human actions in still images will help recognize human actions in videos. However, understanding human actions in still images is a challenging task, because of the large appearance and pose variation in both humans and objects even for the same action.

In the first part of this thesis, we treat action understanding as an image classification task, where the goal is to correctly assign a class label such as "playing violin" or "reading book" to each human. Compared with traditional vision tasks such as object recognition, we show that it is critical to utilize detailed and structured visual information for action classification. To this end, we extract dense and structured visual descriptors for image representation, and propose to combine randomization and discrimination for image classification. The performance of our classification system can be further improved by integrating with other high-level features such as action attributes and objects.

The second part of this thesis aims at having a deeper understanding of human actions. Considering the specific types of human-object interactions for each action,

we first propose a conditional random field model which allows objects and human poses to serve as context of each other, and hence mutually improve each other's recognition results. Then, we move on to discover object functionality in a weakly supervised setting. For example, given a set of images containing human-violin interactions, where a human is either playing violin or holding a violin but not playing, our method builds a model of "playing violin" that corresponds to the functionality of the object, and clusters the input images accordingly.

Finally, we summarize our work and show our vision and preliminary results of how our work can benefit some new vision tasks, including fine-grained object recognition, video event categorization, and social role understanding.

To my family

my parents Jinchun
and Jiaqin
my wife Yaya
my sister Yaojuan

Acknowledgement

First and foremost, I would like to thank my advisor, Fei-Fei Li, for her continuous support through my entire PhD life. I feel truly privileged to have had the opportunity to work with Fei-Fei for one year in Princeton University and four years in Stanford University. With her guidance, I have learned how to select interesting problems, how to come up with effective solutions, how to evaluate results scientifically, and how to present the findings professionally. Fei-Fei is an advisor who always puts the needs of her students above anything else. Besides research, she has also provided me strong moral support and tremendous help on fellowship application and job search.

I feel honored to have Trevor Hastie, Daphne Koller, Percy Liang, and Silvio Savarese on my dissertation committee. I am also grateful to have Andrew Ng on my thesis proposal committee.

I owe much to the professors, researchers, and students I worked with in the past five years – Diane Beck, Gary Bradski, Leonidas Guibas, Xiaoye Jiang, Aditya Khosla, Pushmeet Kohli, Andy Lai Lin, Jiayuan Ma, Juan Carlos Niebles, Sebastian Nowozin, Vignesh Ramanathan, Carsten Rother, Toby Sharp, Hao Su, Kevin Tang, and Dirk Walther. Thank you all for the ideas, discussions, and contributions you have made to my research.

I also thank my wonderful colleagues in the Stanford Vision Lab (former Princeton Vision Lab) – Alexandre Alahi, Chris Baldassano, Barry Chai, Jia Deng, Michelle Greene, Zhiheng Huang, Catalin Iordan, Armand Joulin, Aditya Khosla, Jonathan Krause, Jia Li, Juan Carlos Niebles, Guido Pusiol, Vignesh Ramanathan, Olga Russakovsky, Sanjeev Satheesh, Hao Su, Min Sun, and Kevin Tang. Thank you all for the invaluable help in my research and life. I really enjoyed spending the last five

years with you in the vision lab. I would also like to thank many of the friends I met in Stanford and bay area. They made my Stanford life exciting and colorful.

I also owe a great deal to my master and undergraduate advisors in Tsinghua University, Haizhou Ai and Shao Li. Without their patient guidance, I would not be able to come to the most amazing universities, Princeton and Stanford, to pursue my PhD.

Many thanks to Microsoft Research and Systemanalyse und Programmentwicklung, who generously supported my research through the Microsoft Research PhD Fellowship and SAP Stanford Graduate Fellowship. My research is also supported by a various of grants to my advisor Fei-Fei Li – an NSF CAREER grant (IIS-0845230), an ONR MURI grant, the DARPA Mind's Eye program, the DARPA CSSG program, the DARPA VIRAT program, an NIH grant (1 R01 EY019429), a Google research award, a Intel research sponsorship, the Microsoft Research New Faculty Fellowship, and a Frank Moss Gift Fund.

Most importantly, I would like to thank my wonderful family. First, to my parents, Jinchun Xia and Jiaqin Yao, who made me who I am and always unreservedly support every decision I made. Second, to my elder sister, Yaojuan Xia, for her selfless support to me and taking care of our parents while I am studying aboard. Last but not least, to my wife, Yaya Xie, for her continued love and support to me ever since ten years ago when we met in college.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background: understanding humans

The ability to understand humans is one of the central functions of modern computer vision systems. Among the tens of thousands categories [34] of objects, human is the most important one in both real world and the vision world. Statistical studies have shown that almost 30% of internet images contain humans [114], and more than 40% pixels of YouTube videos are about humans [70].

In the past few decades, considerable research effort has been devoted to understanding humans in computer vision. Great advances have been achieved in human face detection [119, 57], recognition [139], and pedestrian detection [16, 37]. Human pose estimation is a challenging task in still images [38, 124], but depth sensors have made human pose estimation systems commercialized [106]. Thanks to these achievements, the research focus of understanding humans has been shifted to a higher level task in the past ten years – human action recognition.

Research on action recognition has been focusing on the scope of video sequences in the past decade. Before 2008, most work only deals with simple and repetitive actions such as walking, jogging, etc [105, 45]. Recently, recognizing actions in less contrived videos are attracting more and more attention, such as the work on the Hollywood dataset [69], the UCF YouTube dataset [77], the Olympic sports dataset [86], and the TRECVID MED dataset [110]. However, the goal of most research on these

(a) riding horse

(b) reading book

(c) phoning

(d) playing violin

Figure 1.1: Humans can easily recognize many human actions from just still images.

datasets is to simply assign a class label to each video sequence, where no detailed understanding is offered.

Despite the importance of motion cues in visual perception, humans can recognize many common actions from still images without difficulty, such as "reading a book", "playing a violin", etc. There can be a wide range of applications if computers have the same ability. Recognizing human actions in still images is expected to benefit image indexing and search, given the large amount of images containing humans. Given the frequent interactions of humans with objects and scenes, action recognition in still images will also potentially benefit other related problems such as object recognition and scene understanding. However, this problem has received little attention in computer vision community.

This thesis focuses its attention on understanding human actions in still images.

We propose a series of approaches, including feature representation and classifier design, which lead to a recognition system that achieves very promising performance on our collected datasets and the action classifier task of PASCAL VOC challenge [28, 29]. While most previous work deals with simple actions on small scale datasets of still images [49, 59], our work considers more complex and diverse actions [126, 130]. Further, we do not limit our scope to classifying human actions. Inspired by the psychology studies of human-object interaction [64] and object affordance [44], we also aim at having a detailed understanding of human poses and objects manipulated by humans, of which the interactions and object functionality plays a key role.

1.2 Contributions and thesis outline

In Chapter 2, 3, and 4, we treat action recognition as an image classification problem, and propose approaches that achieve very promising performance. In Chapter 5, 6, and 7, we go beyond action classification and aim at having a detailed understanding of human actions. We highlight our contributions in every chapter of this thesis.

Chapter 2 – Grouplet: A Structured Image Representation. We propose a new image representation for human actions called *grouplet*. A grouplet captures the structured information of an image by encoding a number of discriminative visual features and their spatial configurations. Using a dataset of different actions of people interacting with musical instruments, we show that grouplets are more effective in classifying and detecting human-object interactions than other state-of-the-art methods. In particular, our method can make a robust distinction between humans playing the instruments and humans co-occuring with the instruments without the playing action.

Chapter 3 – Combining Randomization and Discrimination. This chapter shows the importance of exploring fine image statistics and identifying discriminative image patches for action recognition. We achieve this goal by combining two ideas, discriminative feature mining and randomization. Discriminative feature mining allows us to model the detailed information that distinguishes different classes of images, while randomization allows us to handle the huge feature space and prevents

over-fitting. We propose a random forest with discriminative decision trees algorithm, where every tree node is a discriminative classifier that is trained by combining the information in this node as well as all upstream nodes. Experimental results show that our method identifies semantically meaningful visual information and outperforms state-of-the-art algorithms on various datasets. With this method, we won the action classification competition of the PASCAL VOC challenge in both 2011 and 2012.

Chapter 4 – Action Attributes and Parts. In this chapter, we propose to use attributes and parts for recognizing human actions in still images. We define action attributes as the verbs that describe the properties of human actions, while the parts of actions are objects and poselets that are closely related to the actions. We jointly model the attributes and parts by learning a set of sparse bases that are shown to carry much semantic meaning. Then, the attributes and parts of an action image can be reconstructed from sparse coefficients with respect to the learned bases. This dual sparsity provides theoretical guarantee of our bases learning and feature reconstruction approach. On the PASCAL action dataset and a new Stanford 40 Actions dataset, we show that our method extracts meaningful high-order interactions between attributes and parts in human actions while achieving state-of-the-art classification performance.

Chapter 5 – Mutual Context Model I: Single Object. We observe that objects and human poses can serve as mutual context to each other, where recognizing one facilitates the recognition of the other. In this chapter we propose a new random field model to encode the mutual context of objects and human poses in human-object interaction activities. We then cast the model learning task as a structure learning problem, of which the structural connectivity between the object, the overall human pose, and different body parts are estimated through a structure search approach, and the parameters of the model are estimated by a new max-margin algorithm. On a sports data set of six classes of human-object interactions, we show that our mutual context model significantly outperforms state-of-the-art in detecting very difficult

objects and human poses.

Chapter 6 – Mutual Context Model II: Multiple Objects. This chapter makes the mutual context model proposed in Chapter 5 able to deal with the actions in which humans interact with any number of objects. Besides the conventional tasks of action classification, object detection, and human pose estimation, we consider a new problem where we measure the similarity between action images. Experimental results show that our method not only improves action classification accuracy, but also learns a similarity measure that is largely consistent with human perception.

Chapter 7 – Discovering Object Functionality. Object functionality refers to the quality of an object that allows humans to perform some specific actions. In this chapter, we propose a weakly supervised approach to discover all possible object functionalities. This is different from most previous work on functionality that either assumes exactly one functionality for each object, or requires detailed annotation of human poses and objects. Our method takes any possible human-object interaction into consideration, and evaluates image similarity in 3D rather than 2D in order to cluster human-object interactions more coherently. Experimental results on a dataset of people interacting with musical instruments show the effectiveness of our approach.

1.3 Previously published work

Most contributions described in this dissertation have first appeared as various publications. These publications are: [126] (Chapter 2), [132] (Chapter 3), [130] (Chapter 4), [127] (Chapter 5), [131] (Chapter 6), [129] (Chapter 5 and 6), [133] (Chapter 7).

Besides the above publications, I also have the following publications during my PhD: [134, 128, 125, 99, 111]. However, they are beyond the scope of this dissertation, and therefore are not discussed in detail here.

Chapter 2

Grouplet: A Structured Image Representation

2.1 Introduction

In recent years, the computer vision field has made great progress in recognizing isolated objects, such as faces and cars. But a large proportion of our visual experience involves recognizing the interaction between objects. For example, seeing a human playing violin delivers a very different story than seeing a person chopping up a violin - one is a musician, the other is probably a contemporary artist. Psychologists have found that different brain areas are involved in recognizing different scenes of multiple objects [64] and in particular, there are neurons that react strongly upon seeing humans interacting with objects [58]. Such evidence shows that the ability to recognize scenes of human-object interactions is fundamental to human cognition.

The goal of this chapter[1] is to use structured visual features to recognize scenes in which a person is interacting with a specific object in a specific manner, such as playing musical instruments. Humans can recognize such activities based on only static images, most likely due to the rich structured information in the activities. For example, "playing violin" is defined not only by the appearance of a human and a violin and their co-occurrence, but also by the gesture of arms interacting with the

[1] An early version of this chapter has been presented in [126].

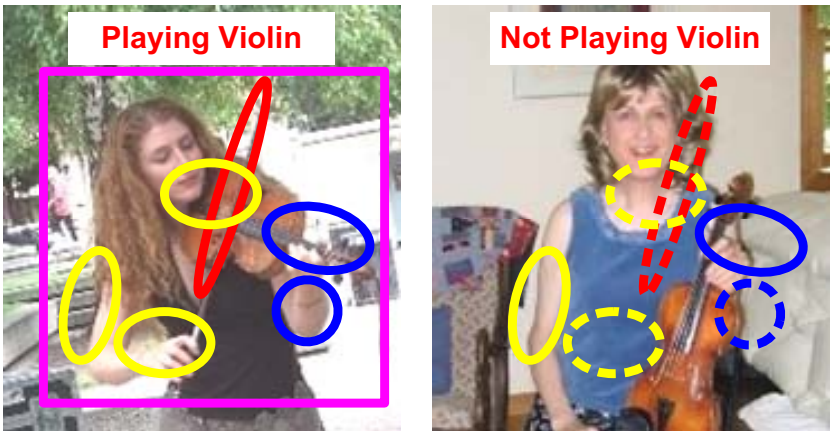


Figure 2.1: Recognizing a person playing violin versus not playing violin requires subtle discriminations of image features. We aim at discovering discriminative features that encode rich and structured information for such tasks. In the **left** figure, three sample grouplets are shown in three different colors. Dashed ellipses in the **right** indicate missing features.

pose of the violin, as shown in Fig.2.1.

In this chapter, we approach the problem by discovering image features that can characterize well different human-object interactions. We take the view in [64] that such human-object configurations are like different types of scenes. So similar to scene and object classification [39, 71], our features need to discover different classes of actions that carry intrinsically different visual appearance and spatial information. This problem offers us an opportunity to explore the following issues that have not been widely studied in generic object recognition tasks:

- Spatial relations among image patches. Recognizing that a person is playing violin is not simply discovering the co-occurrence of the violin and the human, which could also occur when a person just standing next to a violin. Our features need to capture the spatial relations that are crucial to define the human-object interactions.

- More subtle and discriminative features. Most of the current image features (and models) are tested on classes of objects that are very different from each other (e.g. bicycles vs. cows). The classes of human-object interactions are

much more similar, due to the dominant presence of humans in all classes. This demands more discriminative features to encode the image differences.

Focusing on the above issues, we propose a new image representation that encodes appearance, shape, and spatial relations of multiple image patches, termed "grouplet". The grouplets are discovered through a novel data mining approach, and could be further refined by a parameter estimation procedure. We show that the methods using grouplets outperform the state-of-the-art approaches in both human-object interaction *classification* and *detection* tasks.

The rest of this chapter first presents a human-object interaction data set in Section 2.2. Section 2.3 and Section 2.4 define the grouplets and introduce a method of obtaining discriminative grouplets respectively. Section 2.5 briefly describes the classification methods that use grouplets. Related work is discussed in Section 2.6. Experiment results are reported in Section 2.7.

2.2 The PPMI dataset

Most of the popular image data sets are collected for recognizing generic objects [33, 28] or natural scenes [90] instead of human and object interactions. We therefore collected a new data set called People-playing-musical-instruments (PPMI, Figure 2.2). PPMI[2] consists of 7 different musical instruments: bassoon, erhu, flute, French horn, guitar, saxophone, and violin. Each class includes ∼150 PPMI+ images (humans playing instruments) and ∼150 PPMI- images (humans holding the instruments without playing). As Figure 2.2 shows, images in PPMI are highly diverse and cluttered.

We focus on two problems on this data. One is to classify different activities of humans playing instruments; the other is to distinguish PPMI+ and PPMI- images for each instrument. The latter task is very different from traditional image classification tasks. Distinguishing PPMI+ and PPMI- images of the same instrument strongly

[2]The dataset is available at: `http://ai.stanford.edu/~bangpeng/ppmi.html`. Resources of the images include image search engines Google, Yahoo, Baidu, and Bing, and photo hosting websites Flickr and Picassa.

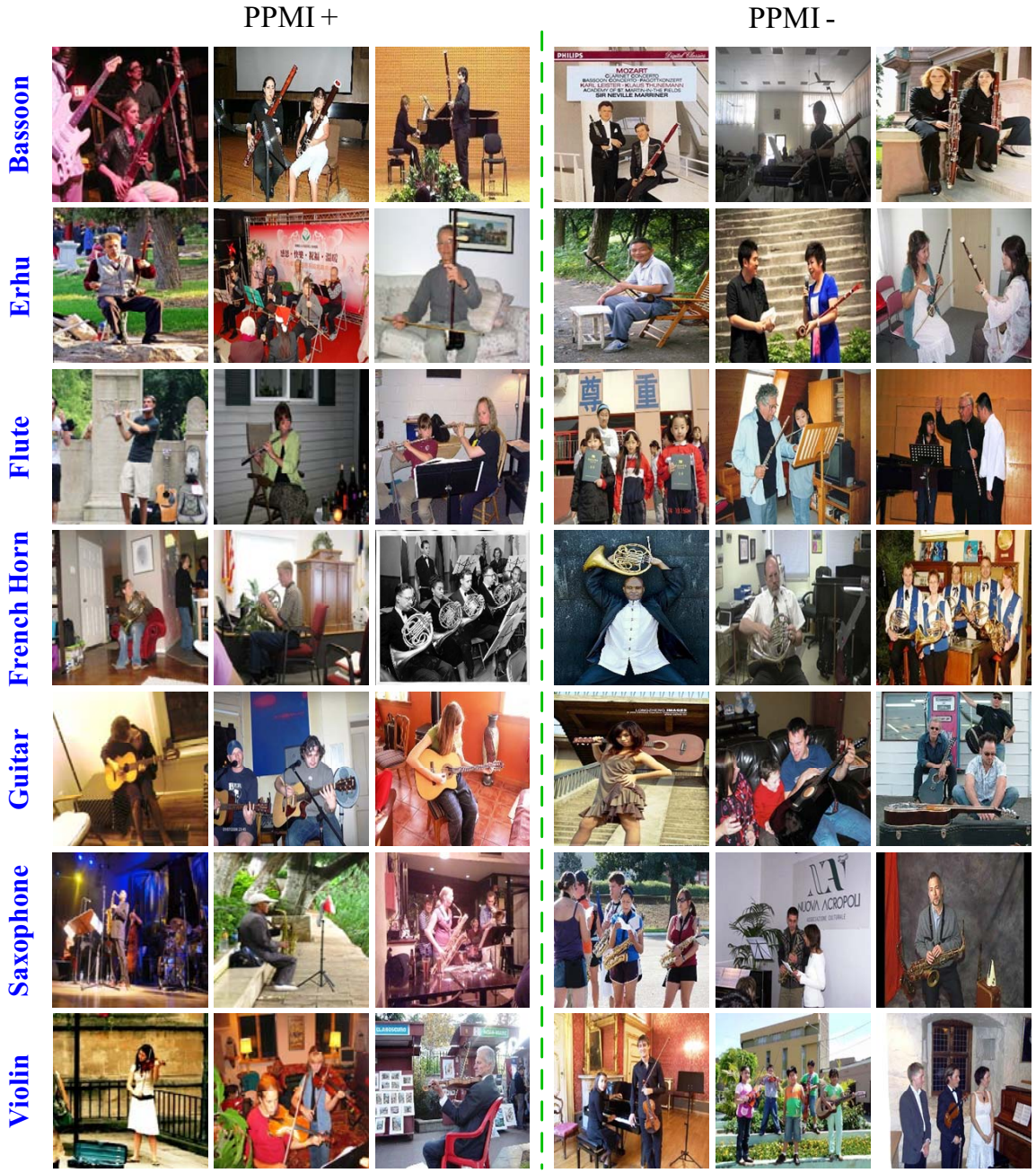


Figure 2.2: Example images of the People-Playing Musical Instrument (PPMI) dataset. PPMI+ indicate images of people playing instruments (PPMI+), while PPMI- indicate images of people co-occurring with but not playing the instruments.

depends on the structural information in the images, such as the spatial relations between the object and the human. This property of our data set cannot be captured by [48] and [49], which are possibly the only existing data sets of human-object interactions. Besides classification, we also show results of detecting people playing different instruments on the PPMI dataset.

2.3 Image building block - the grouplet

For recognizing human-object interactions, we discover a set of discriminative features that encode the structured image information. To address the two central issues introduced in Section 2.1, the grouplets have the following properties.

- Each grouplet contains a set of highly related image patches. It encodes the appearance, location, and shape of these patches, as well as their spatial relationship.
- In order to differentiate human and object interactions, for each action class, we apply a novel data mining approach to discover a set of discriminative grouplets. We then use a generative model to refine the grouplets which also gives us a classifier to distinguish different human actions.

A grouplet is defined by a one-layer AND/OR [13] structure on a set of *feature units*. A feature unit, denoted by $\{A, x, \boldsymbol{\sigma}\}$, indicates that a codeword of visual appearance A, is observed in the neighborhood of location x. A is a categorical variable describing the assignment of visual codewords, and x is a 2D location vector relative to a reference point. The *spatial extent* of A in the neighborhood of x is expressed as a 2D Gaussian distribution $\mathcal{N}(x, \boldsymbol{\sigma})$. In the AND/OR structure, the OR operation makes the grouplets invariant to small pose variations, while the AND operation encodes co-occurrence interactions of different image patches.

Figure 2.3 illustrates two grouplet features. Each grouplet lives in an image space where P indicates a reference location. Each grouplet is composed of a set of feature units. A feature unit, whose visual appearance is denoted by a shaded square patch,

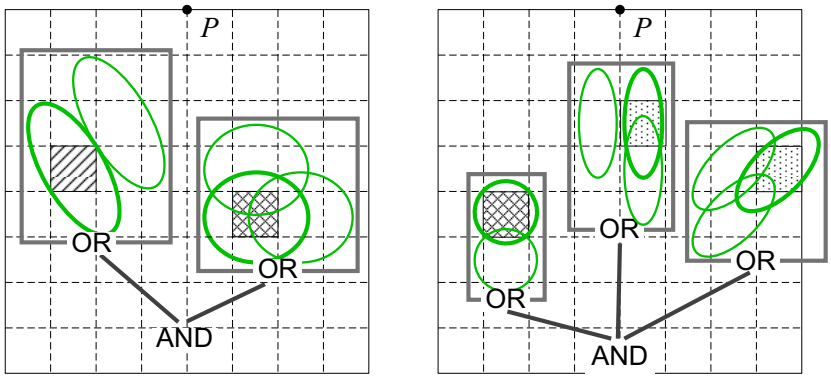


Figure 2.3: Graphical illustration of two examples of grouplets: **left** is a size-2 grouplet; and **right** is a size-3 grouplet. In each image, P indicates a reference point.

can shift around in a local neighborhood (indicated by smaller rectangular boxes). An ellipse surrounding the center of a feature unit indicates the spatial extent of the feature. Within the neighborhood, an OR operation is applied to select the feature unit that has the strongest signal (v, see Section 2.4.1), indicated by the ellipse of thicker lines. An AND operation collects all feature units to form the grouplet. The size of a grouplet is the number of OR operations it contains.

In the grouplet representation, each feature unit captures a specific appearance, location, and spatial extent information of an image patch. Together, the AND operation allows the grouplets to represent various interactions among a set of image patches, and the OR operation makes the grouplets resistent to small spatial variations. By definition, we do not exert any constraint on the appearance or location of the feature units, nor the size of the grouplets. Further, the spatial extent of each feature unit will be refined through a parameter estimation step, thus the grouplets can reflect any structured information among any number of image patches with any appearance. Examples of grouplets are shown in Figure 2.1 and Figure 2.9.

Implementation Details: In the grouplet representation, SIFT descriptors [78] are computed over a dense image grid of D rectangular patches, as in [71]. Using k-means clustering, we obtain a SIFT codebook which contains 250 codewords. Therefore, the visual appearance can be represented by $\{A_w\}_{w=1}^{W}$, where $W = 250$. The feature units in one OR operation should have the same visual codeword. Reference

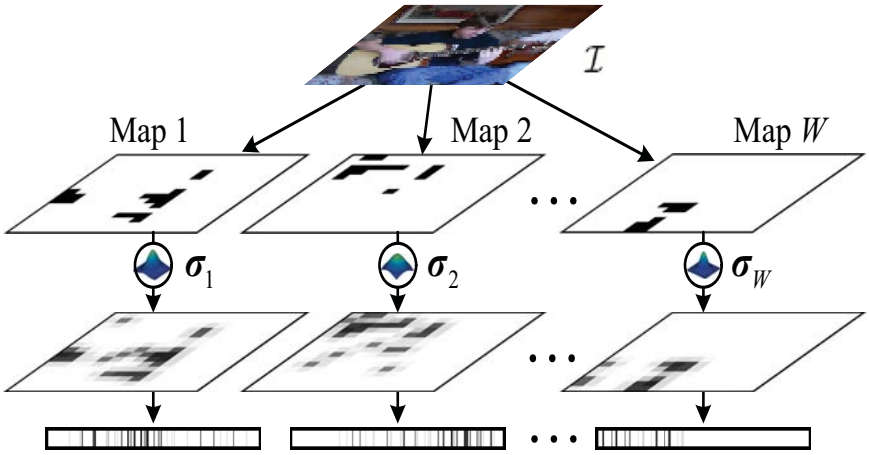


Figure 2.4: Computing the signals v of all feature units on an image $\mathcal{I}$.

points are chosen as the centers of the human faces.

2.4 Obtaining discriminative grouplets

To recognize subtly different scenes, we would like to find a rich set of grouplets that are not only highly characteristic of the image class, but also highly discriminative compared to other classes. We propose a novel data mining algorithm for discovering discriminative grouplets.

2.4.1 Defining discriminative grouplets

A grouplet Λ is discriminative for class c means that Λ has strong signals on images of class c, and has weak signals on images of other classes. In the rest of this section, we first describe how to compute the signal values of feature units and grouplets, and then elaborate on the definition of discriminative grouplets.

The signal v of a feature unit $\{A, x, \boldsymbol{\sigma}\}$ on an image $\mathcal{I}$ is the likelihood that $\{A, x, \boldsymbol{\sigma}\}$ is observed in $\mathcal{I}$:

$$v = \sum_{x' \in \Omega(x)} \Big[p(A|a') \cdot \mathcal{N}(x'|x, \boldsymbol{\sigma}) \Big] \tag{2.1}$$

where $\Omega(x)$ is the image neighborhood of location x, a' is the appearance of the image patch at x', $p(A|a')$ is the probability that a' is assigned to codeword A. As shown in Figure 2.4, first, a codeword assignment map is obtained for each codeword A_w. In Map w, a region is marked black if it is assigned to A_w. Then, each Map w is convolved with a 2D Gaussian distribution with covariance $\boldsymbol{\sigma}_w$. Finally the results are concatenated into a $(D\times W)$-dimensional vector of signal values, where each entry is the signal value of a feature unit on the image. Please refer to the implementation details of this section for more details. For a codeword A_w, we use a single variance $\boldsymbol{\sigma}_w$ to encode its spatial distribution in all positions of the image.

Given the signal values of the feature units in a grouplet, each OR operation selects a feature unit that has the strongest signal (see Figure 2.3). The overall signal of the grouplet, i.e. result of the AND operation, is the smallest signal value of the selected feature units. Intuitively, this decision ensures that even the relatively weakest feature unit needs to be strong enough for the grouplet to be strong (see Figure 2.5). In order to evaluate the discriminability of a grouplet, we introduce two terms, *support value*, $Supp(\cdot)$ and *confidence value*, $Conf(\cdot)$. A grouplet Λ is discriminative for a class c if both $Supp(\Lambda, c)$ and $Conf(\Lambda, c)$ are large. Given a set of training images where the signal of Λ on image $\mathcal{I}_i$ is denoted as r_i, $Supp(\Lambda, c)$ and $Conf(\Lambda, c)$ are computed by

$$Supp(\Lambda, c) = \frac{\sum_{c_i=c} r_i}{\sum_{c_i=c} 1}, \quad Conf(\Lambda, c) = \frac{Supp(\Lambda, c)}{\max\limits_{c'\neq c} Supp(\Lambda, c')} \tag{2.2}$$

where c_i is the class label of $\mathcal{I}_i$. Intuitively, a large $Supp(\Lambda, c)$ indicates that Λ generally has strong signals on images of class c, and a large $Conf(\Lambda, c)$ implies relatively weak signals of Λ on images of classes other than c.

Implementation Details: The size of $\Omega(x)$ is 5×5 patches. We assign each image patch to its nearest codeword: $p(A|a) = 1$ if and only if A is a's nearest codeword. We initialize $\boldsymbol{\sigma}_w$ to $[0.6, 0; 0, 0.6]$ for any A_w. $\boldsymbol{\sigma}_w$ will be updated in the parameter estimation step.

Figure 2.5: Example grouplets whose feature units are of different signal value strengths. One grouplet is presented in each image, where the ellipses indicate the location and spatial extent of the feature units. Thicker lines indicate stronger signal values. For the same flute-playing activity, it is intuitive to see that the grouplet on the **left** has overall stronger feature units than the mixed one in the **middle** and the weaker one on the **right**.

2.4.2 Iteratively mining discriminant grouplets

For each class, our goal is to find all the grouplets of large support and confidence values. One way is to evaluate these values on all possible grouplets. Assuming an image of D patches and a codeword vocabulary of size W, there are $D \times W$ possible feature units. The total number of grouplets is therefore $O(2^{D \times W})$ (in this chapter $D \times W = 240250$). Clearly, evaluating $Supp(\cdot)$ and $Conf(\cdot)$ of all the grouplets for each class is computationally infeasible.

We therefore develop a data mining algorithm for this task, which discriminatively explores the AND/OR structure of the grouplets in an Apriori mining [1] process. Furthermore, we introduce a novel parameter estimation method to better estimate the spatial distribution $\boldsymbol{\sigma}_w$ of each codeword A_w as well as to obtain a set of weights for the grouplets of each class. Our mining algorithm then iterates between the mining process and the parameter estimation process. An overview of the algorithm is shown in Algorithm 1, where l-grouplets indicate the grouplets of size l. We briefly describe the mining and the parameter estimation method in the rest of this section.

The Modified Apriori Mining Algorithm In each iteration of Algorithm 1, given the spatial distribution $\boldsymbol{\sigma}_w$ of each codeword A_w, we compute the signal values of all

foreach *Iteration* **do**
- • Compute signals of all feature units on each image;
- **foreach** *Class* **do**
  - ⋆ Obtain the feature units whose $Supp(\cdot) > T_{Supp}$;
  - ⋆ Generate 1-grouplets; Set $l = 2$;
  - **while** *The number of* $(l-1)$*-grouplets* $\geqslant 2$ **do**
    - Generate candidate l-grouplets; Remove l- grouplets whose $Supp(\cdot) < T_{Supp}$; $l = l + 1$;
  - **end**
  - ⋆ Remove the grouplets whose $Conf(\cdot) < T_{Conf}$.
- **end**
- • Parameter estimation to refine $\boldsymbol{\sigma}_w$ for each A_w and obtain a weight for each mined grouplet.

end

Algorithm 1: Obtaining discriminative grouplets.

the feature units on each image as in Figure 2.4. We are then ready to mine the discriminative grouplets for every class.

We modify the Apriori [1] mining method to explore the AND/OR structures to select the discriminative grouplets. The main idea of Apriori mining is compatible with the AND operation: if an l-grouplet has a large support value, then by removing the feature units in any of its OR operations, the remaining $(l-1)$-grouplets also have large support values. Therefore, we can generate l-grouplets based only on the mined $(l-1)$-grouplets, instead of considering all the possibilities. The OR operation is used to obtain the 1-grouplets. For each codeword, a hierarchical clustering is applied to the feature units that have large enough support values. Each cluster is then initialized as a 1-grouplet. The mining process is briefly shown in Algorithm 1.

Implementation Details: The hierarchical clustering is based on the maximum distance metric, of which the threshold is two times the patch size. The mining algorithm automatically adjusts the values of T_{Supp} and T_{Conf} for each class, so that the number of mined grouplets for different classes are approximately the same. Please refer to the full version of [126] for more details of the mining method.

Refining Grouplets Given a set of mined grouplets, we introduce a parameter estimation method to further refine the spatial distribution $\boldsymbol{\sigma}_w$ of each codeword A_w. With

the refined $\boldsymbol{\sigma}$, one can expect that more accurate signal values of the feature units can be computed, which in turn can be put into the mining process to obtain better grouplets in the next iteration. Furthermore, the algorithm computes a weight on each mined grouplet for each class. The combination of grouplets and the class-dependent weights can then be directly used for classification tasks (see Section 2.5).

Given an image $\mathcal{I}$ with class label c, we compute the likelihood of $\mathcal{I}$ given a set of parameters $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ contains the parameters for the spatial extent of each codeword and the importance of each grouplet.

$$p(\mathcal{I}, c|\boldsymbol{\theta}) = p(c|\boldsymbol{\theta}) \sum_m \left[p(\mathcal{I}|\Lambda^m, \boldsymbol{\theta}) p(\Lambda^m|c, \boldsymbol{\theta})\right] \tag{2.3}$$

where Λ^m indicates the m-th mined grouplet. $p(\mathcal{I}|\Lambda^m, \boldsymbol{\theta})$ denotes the likelihood of $\mathcal{I}$ given Λ^m. $p(\Lambda^m|c, \boldsymbol{\theta})$ models the importance of Λ^m for class c. We assume that the classes are uniformly distributed, and hence $p(c|\boldsymbol{\theta}) = \frac{1}{C}$, where C is the number of classes.

$p(\mathcal{I}|\Lambda^m, \boldsymbol{\sigma})$ denotes the likelihood of $\mathcal{I}$ given Λ^m. We assume that $p(\mathcal{I}|\Lambda^m, \boldsymbol{\sigma}) \propto p(\Lambda^m|\mathcal{I}, \boldsymbol{\sigma})$, and use the signal value of Λ^m on $\mathcal{I}$ to approximately describe $p(\mathcal{I}|\Lambda^m, \boldsymbol{\sigma})$. Furthermore, our goal is approximated by

$$v \approx p(A|a_h) \cdot \mathcal{N}(x_h|x, \boldsymbol{\sigma}) \tag{2.4}$$

where $\{a_h, x_h\} = \arg\max_{a', x'} p(A|a') \cdot \mathcal{N}(x'|x, \boldsymbol{\sigma})$. With this approximation, we can avoid computing marginalization within the "ln" operation in model learning. $p(\Lambda^m|c, \boldsymbol{\pi})$ models the importance of Λ^m for class c. It is expressed as a multinomial distribution,

$$\begin{aligned} p(\Lambda^m|c, \boldsymbol{\theta}) &= \prod_{c'=1}^{C} \mathrm{Mult}(\Lambda^m|\boldsymbol{\pi}_{:,c'})^{\delta(c,c')} \\ &= \prod_{c'=1}^{C} (\boldsymbol{\pi}_{m,c'})^{\delta(c,c')} \end{aligned} \tag{2.5}$$

where $\delta(c, c')$ equals 1 if $c = c'$ and otherwise 0. We can see that $\boldsymbol{\pi}$ is a $M \times C$ matrix.

We use an expectation-maximization (EM) algorithm to estimate the parameters $\boldsymbol{\theta}$. On a PC with a 2.66GHz CPU, our algorithm can process around 20000 grouplets under 3 minutes per EM iteration.

2.5 Using grouplets for classification

Having obtained the discriminative grouplets, we are ready to use them for classification tasks. In this chapter, we show that grouplets can be used for classification either by a generative or a discriminative classifier.

A Generative Classifier. Recall that when refining grouplets, our probabilistic parameter estimation process outputs the importance of each grouplet for each class. This can, therefore, be directly used for classification. Given a new image $\mathcal{I}$, its class label c is predicted as follows,

$$c = \arg\max_{c'} p(c'|\mathcal{I}, \boldsymbol{\theta}) = \arg\max_{c'} p(c', \mathcal{I}|\boldsymbol{\theta}) \tag{2.6}$$

A Discriminative Classifier. Discriminative classifiers such as SVM can be applied by using groupets. Given an image, the input feature vector to SVM classifiers is the signal values of the mined grouplets.

2.6 Related work

Many features have been proposed for various vision tasks in the past decade [116]. It is out of the scope of this chapter to discuss all of them. Instead, we discuss the image representations that have directly influenced our work.

One of the most popular image feature representation schemes is bag of words (BoW) and its derivations (e.g. [71]). These methods have shown promising results in holistic image classification tasks. But by assuming little or no spatial relationships among image patches, these representations are not sufficient for more demanding tasks such as differentiating human and object interactions.

In order to remedy BoW, some methods have been proposed to either encode

longer range image statistics [109, 103] or explicitly model spatial relationships among image patches [39, 38, 87]. But most of such approaches uncover image features in a generative way, which might result in some features that are not essential for recognition. In [37], a deformable part model is presented for discriminatively detecting objects in cluttered scenes. This method, however, assumes that the target object consists of a small number of deformable parts, which might not be able to model the subtle difference between similar image categories.

Our feature is similar in spirit to [8], though independently developed. We differ from [8] in that our features are automatically discovered instead of supervised by humans, making it a more scalable and convenient algorithm. Furthermore, we emphasize the dependence among image features, which is critical for demanding recognition tasks such as human and object interactions.

There has been a lot of work on discriminative feature selection [57, 63]. But most of the methods are not able to manage such a huge number of features (2 to the power of millions) as in the grouplets. Our algorithm is inspired by previous works [95, 136, 135] that also use data mining methods for feature selection. But compared to these previous methods, we take a step further to encode much more structured information in the feature representation.

2.7 Experiment

We first conduct experiments to analyze the properties of grouplets (Section 2.7.1). The rest of this section then focuses on comparing using grouplets for human-object interaction classification and detection with a number of existing state-of-the-art methods. Apart from Section 2.7.5, all experiments use the PPMI dataset introduced in Section 2.2. In Section 2.7.4 we use the original PPMI images. Data sets that are used from Section 2.7.1 to 2.7.3 are obtained as follows. We first run a face detector [57] on all PPMI images. For each instrument, we manually select 200 detection results from PPMI+ and PPMI- images respectively. We then crop a rectangle region of the upper body of each selected detection result and normalize the region to 256×256 pixels so that the face size is 32×32.

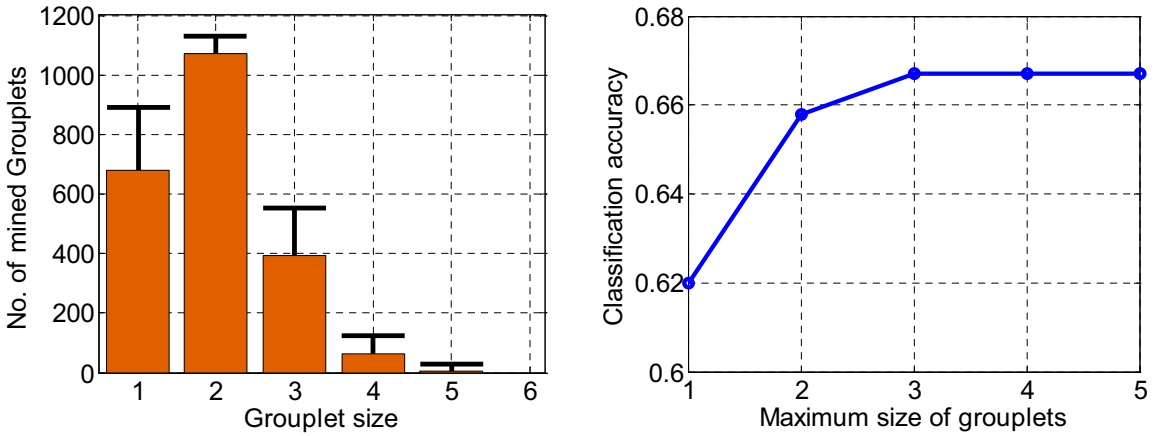


Figure 2.6: Analysis of the properties of grouplets. **left:** Average distribution of grouplets containing different number of feature units in the PPMI images. Error bars indicate standard deviations. **right:** 7-class classification accuracy with respect to the number of feature units in included grouplets.

2.7.1 Analysis of the properties of grouplets

Effects of the grouplet size We use a 7-class classification task to analyze the properties of the mined grouplets (experiment details in Section 2.7.2). Here we use an SVM with the histogram intersection kernel for classification. We use LIBLINEAR [30] for SVM implementation.

Figure 2.6(left) shows the average distribution of different sizes of grouplets. Because the AND operation takes the smallest signal value of all feature units, it is unlikely that grouplets with a very large size can be mined. We observe that a majority of the mined grouplets contain 1, 2, or 3 feature units. Figure 2.6(right) shows the classification performance as the size of the grouplets increases. We see a big increase in accuracy using grouplets from size 1 to size 3. After this, the accuracy stabilizes even when including grouplets of bigger sizes. Two reasons might account for this observation: 1) the number of grouplets containing more than 3 feature units is small, and hence the overall contribution to classification is small; 2) much information in such grouplets is already contained in the grouplets of smaller sizes.

Effect of the Iterative Learning Procedure Given a set of training images, our algorithm iterates between a mining and a parameter estimation step. The idea

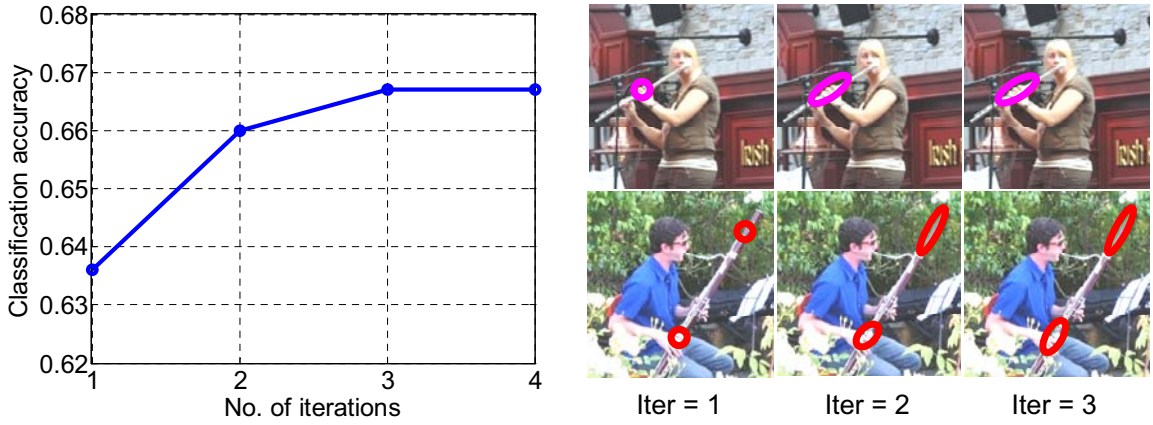


Figure 2.7: **left:** Classification accuracy with respect to the number of iterations (outer loop of Algorithm 1) of grouplet mining and parameter estimation. **right:** Spatial extent of grouplets mined in the 1st, 2nd, and 3rd iteration.

is that each iteration offers a better refinement of the grouplet parameters (e.g. spatial extent of codeword), hence of the overall dscriminability. Figure 2.7(left) shows that the classification accuracy increases with the iteration number. We observe the biggest gain between the first two iterations, indicating that with only two iterations, the method can obtain a good estimation of the spatial extent of each grouplet. Figure 2.7(right) shows that the estimation of the spatial extent of the grouplets align better with the visual features as the iteration increases, resulting in better grouplets.

2.7.2 Classification of playing different instruments

Here we use our algorithm (grouplet+SVM and grouplet+Model, Section 2.5) to classify images of people playing seven different musical instruments. For each class, 100 normalized PPMI+ images are randomly selected for training and the remaining 100 images for testing. We use three iterations of the iterative learning framework to mine around 2000 grouplets for each class. Figure 2.8(left) shows the confusion table obtained by grouplet+SVM with the histogram intersection kernel. We observe that the histogram intersection kernel performs better than the other kernels.

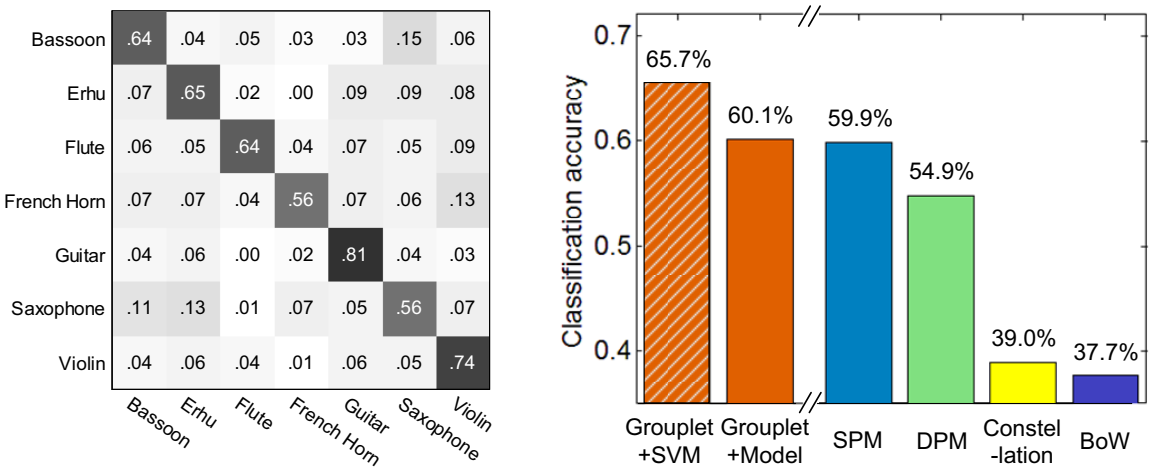


Figure 2.8: 7-class classification using the normalized PPMI+ images. **left:** Confusion matrix obtained by grouplet+SVM. The classification accuracy is 65.7%, whereas chance is 14%. **right:** Classification results of different methods. Y-axis indicates the classification accuracy of each method on the 7 classes.

We compare our grouplet+SVM and grouplet+Model methods with some other approaches, including a four-level spatial pyramid matching (SPM) [71], the deformable part model (DPM) [37], the constellation model [39], and bag-of-words (BoW). The results are shown in Figure 2.8(right). Both BoW and SPM [71] use the histogram representation, where BoW does not consider spatial information in image features while SPM accounts for some level of coarse spatial information by building histograms in different regions of the image. The BoW representation is followed by an SVM classifier with the histogram intersection kernel. Both DPM [37] and the constellation model [39] are part-based models, where DPM trains the classifier discriminatively and constellation model adopts a generative way.

We observe that our grouplet+SVM outperforms the other methods by a large margin. This suggests the effectiveness of the structural information in the mined grouplets. Furthermore, the method that combines grouplets with a generative model achieves comparable performance with SPM. This demonstrates that (1) the discriminatively mined grouplets carry the information that can distinguish images of different classes; (2) our parameter estimation step can effectively learn the weights of each mined grouplet.

2.7.3 Discriminating playing from not playing

Our algorithm aims to learn discriminative structured information of human-object interactions. To demonstrate this, we conduct a classification experiment on PPMI+ vs. PPMI- datasets. For each instrument, we perform a binary classification task: whether the picture contains a person playing the instrument or a person not playing the instrument. Note that all images contain person(s) and instrument(s). The distinction between PPMI+ and PPMI- is only the way the person is interacting with the instrument.

We have 7 binary classification problems. In each problem, 100 normalized PPMI+ and 100 PPMI- images are randomly selected for training, and the other 200 images are used for testing. We mine around 4000 grouplets for both PPMI+ and PPMI- images of each instrument. In Table 2.1, our method is compared with the other approaches described in Section 2.7.2. Due to space limitation, results of the constellation model, which performs on par with BoW, are not listed in Table 2.1. We can see that our method outperforms the other methods on almost all the classes, especially on bassoon, flute, and violin, where our approach improves the accuracy by almost 10%. The only exception is guitar, where DPM achieves the best performance. The reason is that in the normalized images of people playing guitar, the guitar always occupies a big region at the left-bottom part of the image (Figure 2.9). Therefore it is not difficult for the part-based methods (DPM, SPM) to localize the guitar in each

Instruments	SPM [71]	DPM [37]	BoW	Grouplet+Model	Grouplet+SVM
Bassoon	71.5%	68.5%	64.5%	75.0%	**78.0%**
Erhu	78.0%	75.5%	77.5%	**78.5%**	**78.5%**
Flute	84.5%	79.0%	78.0%	85.0%	**90.5%**
French Horn	78.5%	75.5%	71.5%	77.0%	**80.5%**
Guitar	79.5%	**81.0%**	68.0%	73.0%	75.5%
Saxophone	76.0%	76.5%	73.0%	75.0%	**78.5%**
Violin	78.5%	75.5%	74.0%	83.5%	**85.0%**

Table 2.1: Classification results of PPMI+ (playing musical instrument) vs. PPMI- (co-occurring but not playing the instrument). The best performance on each instrument is marked with bold font.

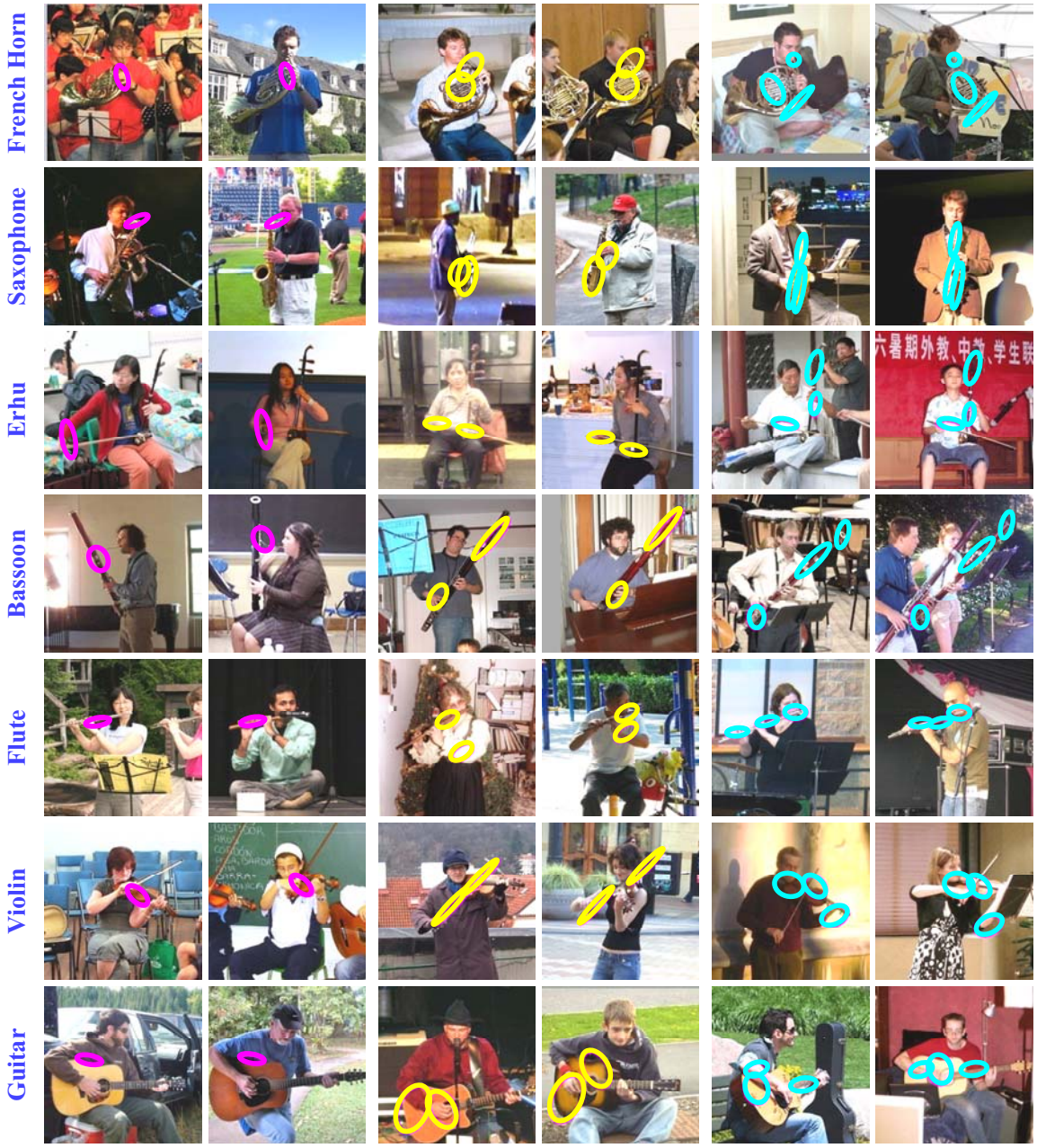


Figure 2.9: Examples of 1, 2, and 3-grouplets on two images of each instrument.

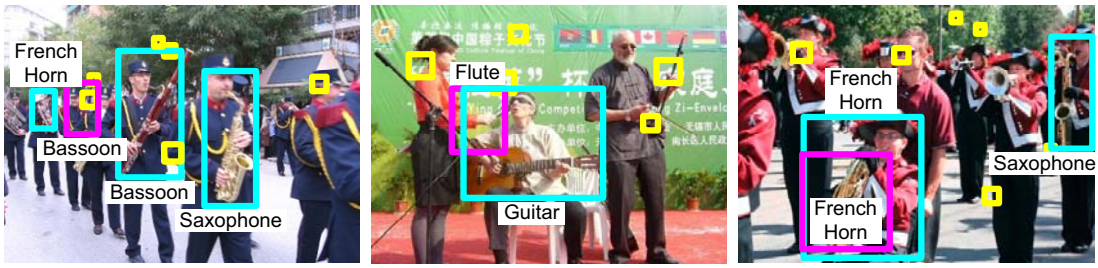
image. Figure 2.10 shows some PPMI+ and PPMI- images with the grouplets that are mined for the corresponding PPMI+ images of the same instrument, where much fewer grouplets are observed on PPMI- images.

Figure 2.10: On each image, we show all grouplets selected by the algorithm for this class whose signal values are stronger than a threshold. We can see that PPMI- images usually have a much smaller number of grouplets with strong signals than PPMI+ images.

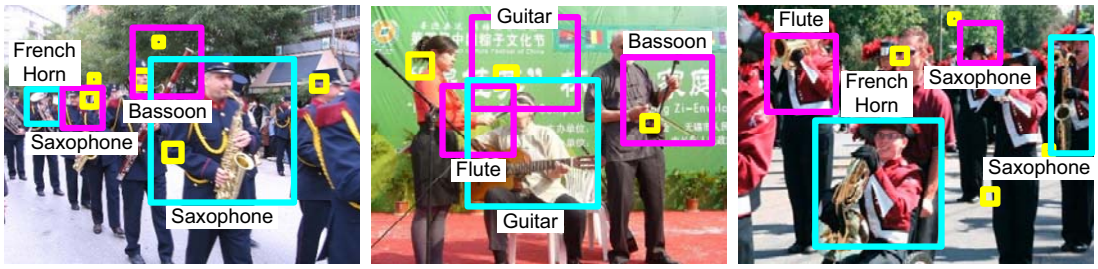
2.7.4 Detecting human and object interactions

Here, we test our approach's ability to detect activities in cluttered scenes. We use the original PPMI images as shown in Figure 2.2. In this experiment, 80 PPMI+ and 80 PPMI- randomly selected images of each instrument are used for training, and the remaining images for testing.

We first run a face detector on all images. We set a relatively low detection threshold to guarantee that almost all human faces are detected. Figure 2.11 shows that many false alarms occur after this step, at positions where no face is present or on a person who is not playing an instrument. Given each face detection, we crop out the neighboring region. Based on these regions, we mine the grouplets that are discriminative for detecting people playing each instrument. Then, an 8-class SVM classifier is trained to determine whether this detection contains a person playing one



(a) Examples of detection results by our method.



(b) Examples of detection results by spatial pyramid matching (SPM) [71].

Figure 2.11: Examples of detection results by our method and SPM. Cyan and magenta rectangles denote the detection results and false alarms respectively. Bounding boxes in (a) are drawn by including all the grouplets that have large signals on the image region. Yellow rectangles show the face detection results which are classified as background.

of the 7 instruments or not. This is a very challenging task (see Figure 2.11). The preliminary experiment result shows that, measured with area under the precision-recall curve, our algorithm significantly outperforms the SPM method [71]: we obtain a 45.7% performance, while SPM is 37.3%. We show examples of both successes and failures of our algorithm and SPM in Figure 2.11, from which we can see that SPM produces more false alarms than our method.

2.7.5 Result on other dataset - Caltech 101

Not only grouplets can be used for recognizing human-object interactions, but it is also a general framework to mine structured visual features in images. Therefore we also test our algorithm in an object recognition task using Caltech101 [33], in the same setting as in [47]. Table 2.2 compares our results with some previous methods.

Method	[4]	[47]	[137]	[42]	Grouplet+SVM
Accuracy	48%	59%	65%	77%	62%

Table 2.2: Recognition results on Caltech 101. The performance is measured by the average accuracy of the 101 classes.

Other than the method in [42], our model performs on par with most of the state-of-the-art algorithms. It is important to note that this experiment is carried out without any additional tuning of the algorithm designed for activity classification. To accommodate objects that are not characterized by specific spatial structures (e.g. articulated animals), some design modifications should be applied to mine the grouplets.

2.8 Summary

In this chapter, we proposed a grouplet feature for recognizing human-object interactions. Grouplets encode detailed and structured information in the image data. A data mining method incorporated with a parameter estimation step is applied to mine the discriminative grouplets. One future research direction would be to link the mined grouplets with semantic meanings in the images to obtain deeper understanding of the scenes of human-object interactions.

Chapter 3

Classification: Combining Randomization and Discrimination

The grouplet is a feature representation that captures subtle and structured visual information. But feature mining and classifier training steps are separated when using grouplet for action classification. In this chapter[1], we treat action recognition as a fine-grained image classification problem, and propose to combine randomization and discrimination for joint feature selection and classifier training, which leads to encouraging results on action recognition and the other fine-grained image classification tasks.

3.1 Introduction

Psychologists have shown that the ability of humans to perform basic-level categorization (e.g. cars vs. dogs; kitchen vs. highway) develops well before their ability to perform subordinate-level categorization, or fine-grained visual categorization (e.g. Golden retrievers vs. Labrador) [62]. It is interesting to observe that computer vision research has followed a similar trajectory. Basic-level object and scene recognition has seen great progress [37, 71, 90, 120] while fine-grained categorization has received little attention. Unlike basic-level recognition, even humans might have difficulty with

[1] An early version of this chapter has been presented in [132].

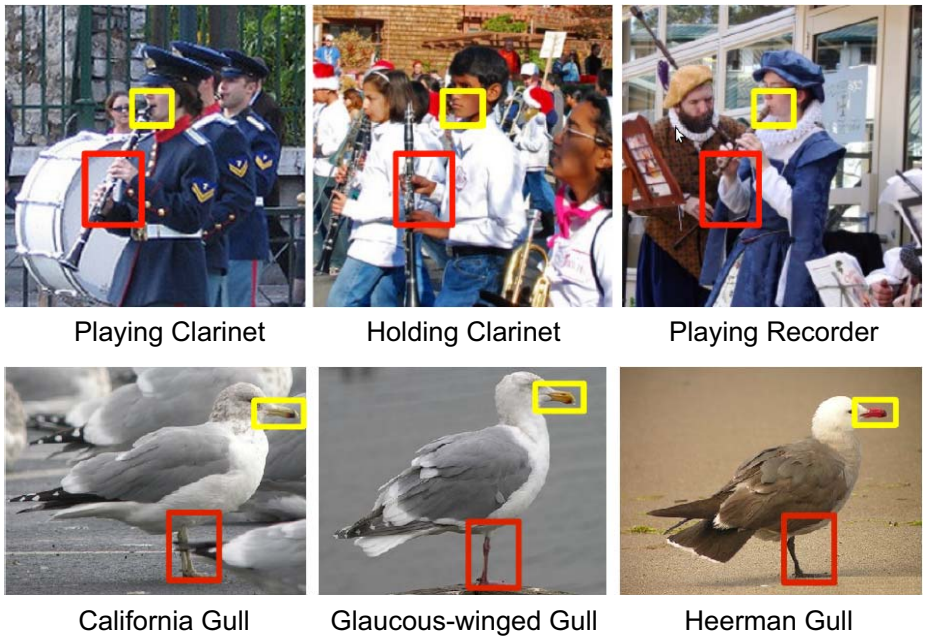


Figure 3.1: Human action recognition (top row) is a fine-grained image classification problem, where the human body dominates the image region. In some sense, it is similar to the subordinate object classification problem. Bounding boxes indicate discriminative image patches.

some of the fine-grained categorization [122]. Thus, an automated visual system for this task could be valuable in many applications.

Action recognition in still images can be regarded as a fine-grained classification problem [55]. Unlike traditional object or scene recognition problems where different classes can be distinguished by different parts or coarse spatial layout [39, 71, 37], more detailed visual distinctions need to be explored for fine-grained image classification. The bounding boxes in Figure 3.1 demarcate the distinguishing characteristics between closely related bird species, or different musical instruments or human poses that differentiate the different playing activities. Models and algorithms designed for basic-level object or image categorization tasks are often unprepared to capture such subtle differences among the fine-grained visual classes. In this chapter, we approach this problem from the perspective of finding a large number of image patches with arbitrary shapes, sizes, or locations, as well as interactions between pairs of patches that

carry discriminative image statistics [25, 126] (Section 3.3). However, this approach poses a fundamental challenge: without any feature selection, even a modestly sized image will yield millions or billions of image patches. Furthermore, these patches are highly correlated because many of them overlap significantly. To address these issues, we propose the use of *randomization* that considers a random subset of features at a time.

In this chapter, we propose a *random forest with discriminative decision trees* algorithm to discover image patches and pairs of patches that are highly discriminative for fine-grained categorization tasks. Unlike conventional decision trees [23, 10, 7], our algorithm uses strong classifiers at each node and combines information at different depths of the tree to effectively mine a very dense sampling space. Our method significantly improves the strength of the decision trees in the random forest while still maintaining low correlation between the trees. This allows our method to achieve low generalization error according to the theory of random forest [10].

Besides human action recognition in still images [126, 28, 29], we also evaluate our method on subordinate categorization of closely related animal species [122]. We show that our method achieves state-of-the-art results. Furthermore, our method identifies semantically meaningful image patches that closely match human intuition. Additionally, our method tends to automatically generate a coarse-to-fine structure of discriminative image regions, which parallels the human visual system [14].

The remaining part of this chapter is organized as follows: Section 3.2 discusses related work. Section 3.3 describes our dense feature space and Section 3.4 describes our algorithm for mining this space. Experimental results are discussed in Section 3.5, and Section 3.6 summarizes this chapter.

3.2 Related work

Image classification has been studied for many years. Most of the existing work focuses on basic-level categorization such as objects [34, 7, 37] or scenes [90, 35, 71]. In this chapter we focus on fine-grained image categorization [55, 9], which requires an approach to capture the fine and detailed information in images.

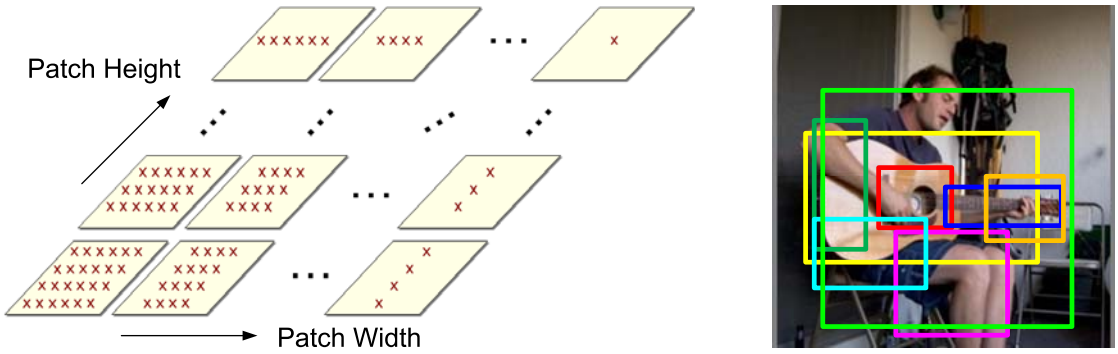
In this chapter, we explore a dense feature representation to distinguish fine-grained image classes. The previous chapter has shown the advantage of dense features ("Grouplet" features [126]) in classifying human activities. Instead of using the generative local features as in Grouplet, here we consider a richer feature space in a discriminative setting where both local and global visual information are fused together. Inspired by [25, 126], our approach also considers pairwise interactions between image regions.

We use a random forest framework to identify discriminative image regions. Random forests have been used successfully in many vision tasks such as object detection [7], segmentation [107] and codebook learning [83]. Inspired from [115], we combine discriminative training and randomization to obtain an effective classifier with good generalizability. Our method differs from [115] in that for each tree node, we train an SVM classifier from one of the randomly sampled image regions, instead of using AdaBoost to combine weak features from a fixed set of regions. This allows us to explore an extremely large feature set efficiently.

A classical image classification framework [120] is *Feature Extraction* $\rightarrow$ *Coding* $\rightarrow$ *Pooling* $\rightarrow$ *Concatenating*. *Feature extraction* [78] and better *coding* and *pooling* methods [120] have been extensively studied for object recognition. In this work, we use discriminative feature mining and randomization to propose a new feature *concatenating* approach, and demonstrate its effectiveness on fine-grained image categorization tasks.

3.3 Dense sampling space

Our algorithm aims to identify fine image statistics that are useful for fine-grained categorization. For example, in order to classify whether a human is playing a guitar or holding a guitar without playing it, we want to use the image patches below the human face that are closely related to the human-guitar interaction (Figure 3.2(b)). An algorithm that can reliably locate such regions is expected to achieve high classification accuracy. We achieve this goal by searching over rectangular image patches of arbitrary width, height, and image location. We refer to this extensive set of image



(a) (b)

Figure 3.2: Illustration of the proposed dense sampling space. (a) We densely sample rectangular image patches with varying widths and heights. The regions are closely located and have significant overlaps. The red × denote the centers of the patches, and the arrows indicate the increment of the patch width or height. (b) Illustration of some image patches that may be discriminative for "playing-guitar". All those patches can be sampled from our dense sampling space.

regions as the *dense sampling space*, as shown in Figure 3.2(a). This figure has been simplified for visual clarity, and the actual density of regions considered in our algorithm is significantly higher. We note that the regions considered by spatial pyramid matching [71] is a very small subset lying along the diagonal of the height-width plane that we consider. Further, to capture more discriminative distinctions, we also consider interactions between pairs of arbitrary patches. The pairwise interactions are modeled by applying concatenation, absolute of difference, or intersection between the feature representations of two image patches.

However, the dense sampling space is very huge. Sampling image patches of size 50×50 in a 400×400 image every four pixels leads to thousands of patches. This increases many-folds when considering regions with arbitrary widths and heights. Further considering pairwise interactions of image patches will effectively lead to trillions of features for each image. In addition, there is much noise and redundancy in this feature set. On the one hand, many image patches are not discriminative for distinguishing different image classes. On the other hand, the image patches are highly overlapped in the dense sampling space, which introduces significant redundancy among these features. Therefore, it is challenging to explore this high-dimensional,

noisy, and redundant feature space. In this work, we address this issue using randomization.

3.4 Discriminative random forest

In order to explore the dense sampling feature space for fine-grained visual categorization, we combine two concepts: (1) *Discriminative training* to extract the information in the image patches *effectively*; (2) *Randomization* to explore the dense feature space *efficiently*. Specifically, we adopt a random forest [10] framework where each tree node is a discriminative classifier that is trained on one or a pair of image patches. In our setting, the discriminative training and randomization can benefit from each other. We summarize the advantages of our method below:

- The random forest framework allows us to consider a subset of the image regions at a time, which allows us to explore the dense sampling space efficiently in a principled way.

- Random forest selects a best image patch in each node, and therefore it can remove the noise-prone image patches and reduce the redundancy in the feature set.

- By using discriminative classifiers to train the tree nodes, our random forest has much stronger decision trees. Further, because of the large number of possible image regions, it is likely that different decision trees will use different image regions, which reduces the correlation between decision trees. Therefore, our method is likely to achieve low generalization error (Section 3.4.4) compared with the traditional random forest [10] which uses weak classifiers in the tree nodes.

An overview of the random forest framework we use is shown in Algorithm 2. In the following sections, we first describe this framework (Section 3.4.1). Then we elaborate on our feature sampling (Section 3.4.2) and split learning (Section 3.4.3)

foreach *tree* t **do**
- Sample a random set of training examples $\mathcal{D}$;
- `SplitNode(`$\mathcal{D}$`)`;
- **if** *needs to split* **then**
  - i. Randomly sample the candidate (pairs of) image regions (Section 3.4.2);
  - ii. Select the best region to split $\mathcal{D}$ into two sets $\mathcal{D}_1$ and $\mathcal{D}_2$ (Section 3.4.3);
  - iii. Go to `SplitNode(`$\mathcal{D}_1$`)` and `SplitNode(`$\mathcal{D}_2$`)`.
- **else**
  - Return $P_t(c)$ for the current leaf node.
- **end**

end

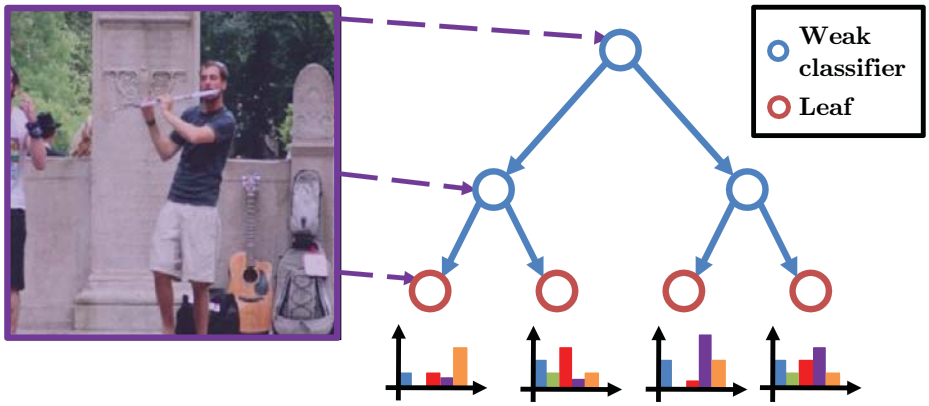
Algorithm 2: Overview of the process of growing decision trees in the random forest framework.

strategies in detail, and describe the generalization theory [10] of random forest which guarantees the effectiveness of our algorithm (Section 3.4.4).

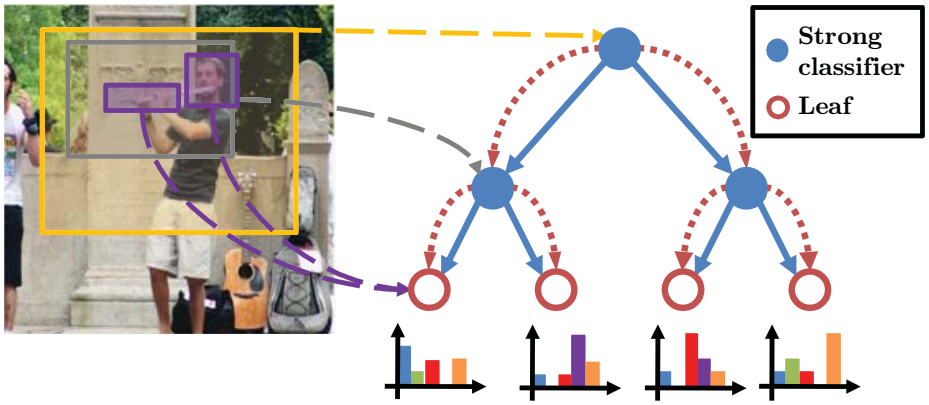
3.4.1 The random forest framework

Random forest is a multi-class classifier consisting of an ensemble of decision trees where each tree is constructed via some randomization. As illustrated in Figure 3.3(a), the leaf nodes of each tree encode a distribution over the image classes. All internal nodes contain a binary test that splits the data and sends the splits to its children nodes. The splitting is stopped when a leaf node is encountered. An image is classified by descending each tree and combining the leaf distributions from all the trees. This method allows the flexibility to explore a large feature space effectively because it only considers a subset of features in every tree node.

Each tree returns the posterior probability of an example belonging to the given classes. The posterior probability of a particular class at each leaf node is learned as the proportion of the training images belonging to that class at the given leaf node. The posterior probability of class c at leaf l of tree t is denoted as $P_{t,l}(c)$. Thus, a test image can be classified by averaging the posterior probability from the leaf node



(a) Conventional random decision tree.



(b) The proposed discriminative decision tree.

Figure 3.3: Comparison of conventional random decision trees with our discriminative decision trees. Solid blue arrows show binary splits of the data. Dotted lines from the shaded image regions indicate the region used at each node. Conventional decision trees use information from the entire image at each node, which encodes no spatial or structural information, while our decision trees sample single or multiple image regions from the dense sampling space (Figure 3.2(a)). The histograms below the leaf nodes illustrate the posterior probability distribution $P_{t,l}(c)$ (Section 3.4.1). In (b), dotted red arrows between nodes show our nested tree structure that allows information to flow in a top-down manner. Our approach uses strong classifiers in each node (Section 3.4.3), while the conventional method uses weak classifiers.

of each tree:

$$c^* = \arg\max_c \frac{1}{T} \sum_{t=1}^{T} P_{t,l_t}(c), \tag{3.1}$$

where c^* is the predicted class label, T is the total number of trees, and l_t is the leaf node that the image falls into.

In the following sections, we describe the process of obtaining $P_{t,l}(c)$ using our algorithm. Readers can refer to previous works [10, 7, 107] for more details of the conventional decision tree learning procedure.

3.4.2 Sampling the dense feature space

As shown in Figure 3.3(b), each internal node in our decision tree corresponds to a single or a pair of rectangular image regions that are sampled from the dense sampling space (Section 3.3), where the regions can have many possible widths, heights, and image locations. In order to sample a candidate image region, we first normalize all images to unit width and height, and then randomly sample (x_1, y_1) and (x_2, y_2) from a uniform distribution $U([0, 1])$. These coordinates specify two diagonally opposite vertices of a rectangular region. Such regions could correspond to small areas of the image (e.g. the purple bounding boxes in Figure 3.3(b)) or even the complete image. This allows our method to capture both global and local information in the image.

In our approach, each sampled image region is represented by a histogram of visual descriptors. For a pair of regions, the feature representation is formed by applying histogram operations (e.g. concatenation, intersection, etc.) to the histograms obtained from both regions. Furthermore, the features are augmented with the decision value $\mathbf{w}^T\mathbf{f}$ (described in Section 3.4.3) of this image from its parent node (indicated by the dashed red lines in Figure 3.3(b)). Therefore, our feature representation combines the information of all upstream tree nodes that the corresponding image has descended from. We refer to this idea as "nesting". Using feature sampling and nesting, we obtain a candidate set of features, $\mathbf{f} \in \mathbb{R}^n$, corresponding to a candidate image region of the current node.

Implementation details. Our method is flexible to use many different visual descriptors. In this work, we densely extract SIFT [78] descriptors on each image with a spacing of four pixels. The scales of the grids to extract descriptors are 8, 12,

16, 24, and 30. Using k-means clustering, we construct a vocabulary of codewords[2]. Then, we use Locality-constrained Linear Coding [120] to assign the descriptors to codewords. A bag-of-words histogram representation is used if the area of the patch is smaller than 0.2, while a 2-level or 3-level spatial pyramid is used if the area is between 0.2 and 0.8 or larger than 0.8 respectively. Note that all parameter here are empirically chose. Using other similar parameters will lead to very similar results.

During sampling (step i of Algorithm 2), we consider four settings of image patches: a single image patch and three types of pairwise interactions (concatenation, intersection, and absolute of difference of the two histograms). We sample 25 and 50 image regions (or pairs of regions) in the root node and the first level nodes respectively, and sample 100 regions (or pairs of regions) in all other nodes. Sampling a smaller number of image patches in the root can reduce the correlation between the resulting trees.

3.4.3 Learning the splits

In this section, we describe the process of learning the binary splits of the data using SVM (step ii in Algorithm 2). This is achieved in two steps: (1) Randomly assigning all examples from each class to a binary label; (2) Using SVM to learn a binary split of the data.

Assume that we have C classes of images at a given node. We uniformly sample C binary variables, $\mathbf{b}$, and assign all examples of a particular class c_i a class label of b_i. As each node performs a binary split of the data, this allows us to learn a simple binary SVM at each node. This improves the scalability of our method to a large number of classes and results in well-balanced trees. Using the feature representation $\mathbf{f}$ of an image region (or pairs of regions) as described in Section 3.4.2, we find a binary split of the data:

$$\begin{cases} \mathbf{w}^T\mathbf{f} \leq 0, \text{go to left child} \\ \text{otherwise, go to right child} \end{cases}$$

[2]A dictionary size of 1024, 256, 256 is used for PASCAL action [28, 29], PPMI [126], and Caltech-UCSD Birds [122] datasets respectively.

where $\mathbf{w}$ is the set of weights learned from a linear SVM.

We evaluate each binary split that corresponds to an image region or pairs of regions with the information gain criteria [7], which is computed from the complete training images that fall at the current tree node. The splits that maximize the information gain are selected and the splitting process (step iii in Algorithm 2) is repeated with the new splits of the data. The tree splitting stops if a pre-specified maximum tree depth has been reached, or the information gain of the current node is larger than a threshold, or the number of samples in the current node is small.

3.4.4 Generalization error of random forests

In [10], it has been shown that an upper bound for the generalization error of a random forest is given by

$$\rho(1 - s^2)/s^2, \tag{3.2}$$

where s is the strength of the decision trees in the forest, and ρ is the correlation between the trees. Therefore, the generalization error of a random forest can be reduced by making the decision trees stronger or reducing the correlation between the trees.

In our approach, we learn discriminative SVM classifiers for the tree nodes. Therefore, compared to the traditional random forests where the tree nodes are weak classifiers of randomly generated feature weights [7], our decision trees are much stronger. Furthermore, since we are considering an extremely dense feature space, each decision tree only considers a relatively small subset of image patches. This means there is little correlation between the trees. Therefore, our random forest with discriminative decision trees algorithm can achieve very good performance on fine-grained image classification, where exploring fine image statistics discriminatively is important. In Section 3.5.5, we show the strength and correlation of different settings of random forests with respect to the number of decision trees, which justifies the above arguments. Please refer to [10] for details about how to compute the strength and correlation values for a random forest.

3.5 Experiments

In this section, we first evaluate our algorithm on two fine-grained image datasets: actions of people-playing-musical-instrument (PPMI) [126] (Section 3.5.1) and a subordinate object categorization dataset of 200 bird species [122] (Section 3.5.2). Experimental results show that our algorithm outperforms state-of-the-art methods on these datasets. Further, we use the proposed method to participate the action classification competition of the PASCAL VOC challenge, and obtain the winning award in both 2011 [28] and 2012 [29]. Detailed results and analysis are shown in Section 3.5.3 and Section 3.5.4. Finally, we evaluate the strength and correlation of the decision trees in our method, and compare the result with the other settings of random forests to show why our method can lead to better classification performance (Section 3.5.5).

3.5.1 People-playing-musical-instrument (PPMI)

The people-playing-musical-instrument (PPMI) data set is introduced in [126]. This data set puts emphasis on understanding subtle interactions between humans and objects. Here we use a full version of the dataset which contains twelve musical instruments; for each instrument there are images of people playing the instrument and holding the instrument but not playing it. We evaluate the performance of our method with 100 decision trees on the 24-class classification problem. We compare our method with many previous results[3], including bag of words, grouplet [126], spatial pyramid matching (SPM) [71], locality-constrained linear coding (LLC) [120]. The grouplet method uses one SIFT scale, while all the other methods use multiple SIFT scales described in Section 3.4.2. Table 3.1 shows that we significantly outperform the a various of previous approaches.

Table 3.2 shows the result of our method on the 12 binary classification tasks where each task involves distinguishing the activities of playing and not playing for

[3]The baseline results are available from the dataset website `http://ai.stanford.edu/~bangpeng/ppmi`.

Method	BoW	Grouplet [126]	SPM [71]	LLC [120]	Ours
mAP (%)	22.7	36.7	39.1	41.8	**47.0**

Table 3.1: Mean Average Precision (% mAP) on the 24-class classification problem of the PPMI dataset. The best result is highlighted with bold fonts.

the same instrument. Despite a high baseline of 89.2% mAP, our method outperforms by 2.9% to achieve a result of 92.1% overall. We also perform better than the grouplet approach [126] by 7%, mainly because the random forest approach is more expressive. While each grouplet is encoded by a single visual codeword, each node of the decision trees here corresponds to an SVM classifier. Furthermore, we outperform the baseline methods on nine of the twelve binary classification tasks. In Figure 3.4, we visualize the heat map of the features learned for this task. We observe that they show semantically meaningful locations of where we would expect the discriminative regions of people playing different instruments to occur. For example, for flute, the region around the face provides important information while for guitar, the region to the left of the torso provides more discriminative information. It is interesting to note that despite the randomization and the algorithm having no prior information, it is able to locate the region of interest reliably.

Furthermore, we also demonstrate that the method learns a coarse-to-fine region of interest for identification. This is similar to the human visual system which is believed to analyze raw input in order from low to high spatial frequencies or from large global shapes to smaller local ones [14]. Figure 3.5 shows the heat map of the area selected by our classifier as we consider different depths of the decision tree. We observe that our random forest follows a similar coarse-to-fine structure. The average area of the patches selected reduces as the tree depth increases. This shows that the classifier first starts with more global features or high frequency features to discriminate between multiple classes, and finally zeros in on the specific discriminative regions for some particular classes.

Instrument	BoW	Grouplet [126]	SPM [71]	LLC [120]	Ours
Bassoon	73.6	78.5	84.6	85.0	**86.2**
Erhu	82.2	87.6	88.0	89.5	**89.8**
Flute	86.3	95.7	95.3	97.3	**98.6**
French horn	79.0	84.0	93.2	93.6	**97.3**
Guitar	85.1	87.7	**93.7**	92.4	93.0
Saxophone	84.4	87.7	89.5	88.2	**92.4**
Violin	80.6	93.0	93.4	**96.3**	95.7
Trumpet	69.3	76.3	82.5	86.7	**90.0**
Cello	77.3	84.6	85.7	82.3	**86.7**
Clarinet	70.5	82.3	82.7	84.8	**90.4**
Harp	75.0	87.1	92.1	**93.9**	92.8
Recorder	73.0	76.5	78.0	79.1	**92.8**
Average	78.0	85.1	88.2	89.2	**92.1**

Table 3.2: Comparison of mean Average Precision (% mAP) of the results obtained by different methods on the PPMI binary classification tasks of people playing and holding different musical instruments. Each column shows the results obtained from one method. The best results are highlighted with bold fonts.

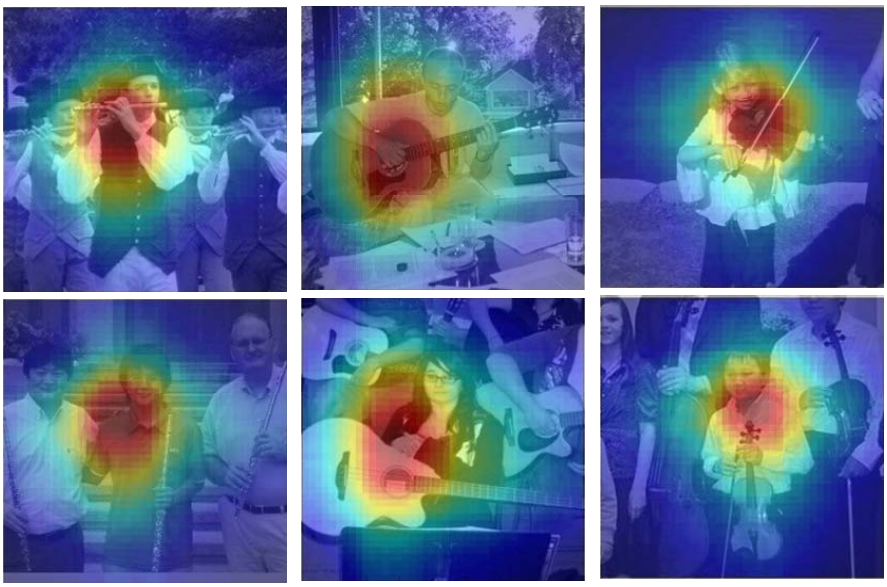
Method	MKL [9]	LLC [120]	Ours
Accuracy	19.0%	18.0%	**19.2%**

Table 3.3: Comparison of the mean classification accuracy of our method and the baseline results on the Caltech-UCSD Birds 200 dataset. The best performance is indicated with bold fonts.

3.5.2 Caltech-UCSD birds 200 (CUB-200)

The Caltech-UCSD Birds (CUB-200) dataset contains 6,033 annotated images of 200 different bird species [122]. This dataset has been designed for subordinate image categorization. It is a very challenging dataset as the different species are very closely related and have similar shape/color. There are around 30 images per class with 15 for training and the remaining for testing. The test-train splits are fixed (provided on the website).

The images are cropped to the provided bounding box annotations. These regions are resized such that the smaller image dimension is 150 pixels. As color provides important discriminative information, we extract C-SIFT descriptors [117] in the



(a) flute (b) guitar (c) violin

Figure 3.4: (a) Heat map of the dominant regions of interest selected by our method for playing flute on images of playing flute (top row) and holding a flute without playing it (bottom row). (b,c) shows similar images for guitar and violin, respectively. The heat maps are obtained by aggregating image regions of all the tree nodes in the random forest weighted by the probability of the corresponding class. Red indicates high frequency and blue indicates low frequency.

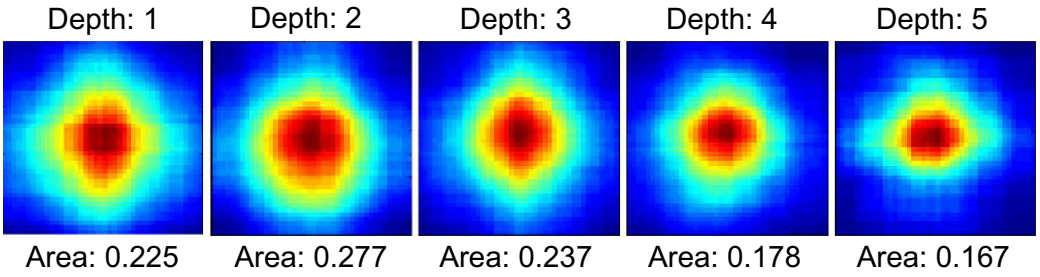


Figure 3.5: Heat map for "playing trumpet" class with the weighted average area of selected image regions for each tree depth. Please refer to Figure 3.4 for how the heat maps are obtained.

same way described in Section 3.4.2. We use 300 decision trees in our random forest. Table 3.3 compares the performance of our algorithm against the LLC baseline and the state-of-the-art result (multiple kernel learning (MKL) [9]) on this dataset.

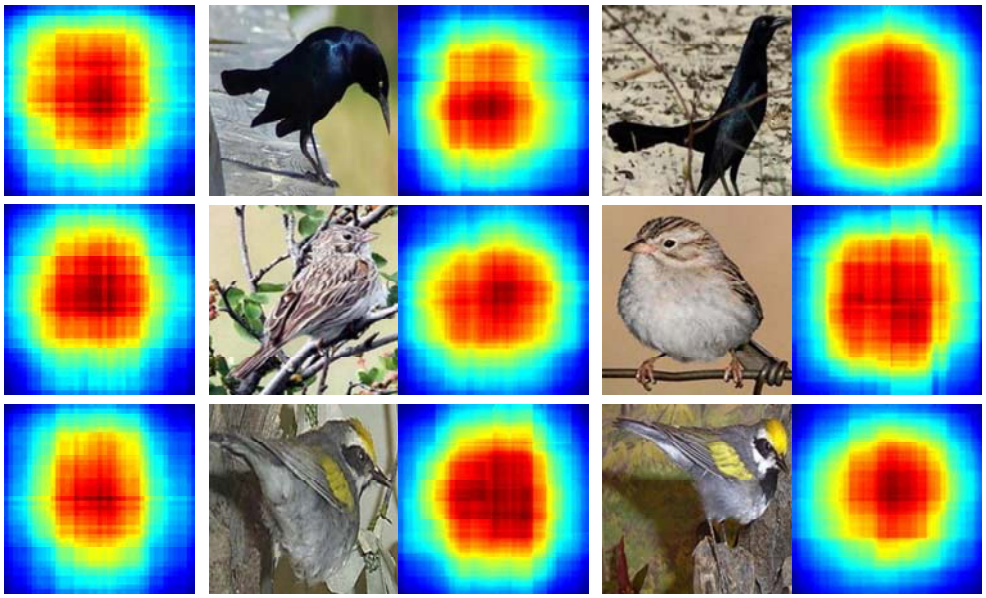


Figure 3.6: Each row represents visualizations for a single class of birds (from top to bottom): boat tailed grackle, brewer sparrow, and golden winged warbler. For each class, we visualize: (a) Heat map for the given bird as described in Figure 3.4; (b,c) Two example images of the corresponding bird and the distribution of image patches selected for the specific image.

Our method outperforms LLC and achieves comparable performance with the MKL approach. We note that [9] uses multiple features e.g. geometric blur, gray/color SIFT, full image color histograms etc. It is expected that including these features can further improve the performance of our method. Furthermore, we show in Figure 3.6 that our method is able to capture the intra-class pose variations by focusing on different image regions for different images.

3.5.3 PASCAL 2011 action classification

The most recent PASCAL VOC challenge incorporated the task of recognizing actions in still images. The images describe ten common human activities: “Jumping”, “Phoning”, “Playing a musical instrument”, “Reading”, “Riding a bicycle or motorcycle”, “Riding a horse”, “Running”, “Taking a photograph”, “Using a computer”,

and "Walking". Each person that we need to classify is indicated by a bounding box and is annotated with one of the nine actions they are performing. There are also humans performing actions that do not belong to any of the ten aforementioned categories. There actions are all labeled as "Other".

We participated the competition using the method proposed in this chapter, and won the winning award in both 2011 [28][4] and 2012 [29][5]. We introduce the details of our results in the 2011 challenge [28] in the rest of this subsection. Section 3.5.4 will cover our results in the 2012 challenge [29].

There are around 2,500 training/validation images and a similar number of testing images in the 2011 dataset. As in [18], we obtain a foreground image for each person by extending the bounding box of the person to contain 1.5× the original size of the bounding box, and resizing it such that the larger dimension is 300 pixels. We also resize the original image accordingly. Therefore for each person, we have a "person image" as well as a "background image". We only sample regions from the foreground and concatenate the features with a 2-level spatial pyramid of the background. We use 100 decision trees in our random forest.

Classification results measured by mean Average Precision (mAP) are shown in Table 3.4. Our method achieves the best result on six out of the ten actions. Note that we achieved this accuracy based on only grayscale SIFT descriptors, without using any other features or contextual information like object detectors.

Figure 3.7 shows the frequency of an image patch being selected by our method. For each activity, the figure is obtained by considering the features selected in the tree nodes weighted by the proportion of samples of this activity in this node. From the results, we can clearly see the difference of distributions for different activities. For example, the image patches corresponding to human-object interactions are usually highlighted, such as the patches of bikes and books. We can also see that the image patches corresponding to background are not frequently selected. This demonstrates our algorithm's ability to deal with background clutter.

[4]A summary of the results in 2011 PASCAL challenge is in `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/workshop/index.html`.

[5]A summary of the results in 2012 PASCAL challenge is in `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/workshop/index.html`.

Action	CAENLEAR DSAL	CAENLEAR HOBJ_DSAL	NUDT CONTEXT	NUDT SEMANTIC	Ours
Jumping	62.1%	**71.6%**	65.9%	66.3%	66.0%
Phoning	39.7%	**50.7%**	41.5%	41.3%	41.0%
Playing instrument	60.5%	**77.5%**	57.4%	53.9%	60.0%
Reading	33.6%	37.8%	34.7%	35.2%	**41.5%**
Riding bike	80.8%	86.5%	88.8%	88.8%	**90.0%**
Riding horse	83.6%	89.5%	90.2%	90.0%	**92.1%**
Running	80.3%	83.8%	**87.9%**	87.6%	86.6%
Taking photo	23.2%	25.1%	25.7%	25.5%	**28.8%**
Using computer	53.4%	58.9%	54.5%	53.7%	**62.0%**
Walking	50.2%	59.2%	59.5%	58.2%	**65.9%**

Table 3.4: Comparison of the mean Average Precision of our method and the other approaches in the action classification competition of PASCAL VOC 2011. Each column shows the result from one method. The best results are highlighted with bold fonts. We skipped the results of MISSOURI_SSLMF and WVU_SVM-PHOW, which did not outperform on any class, due to space limitations.

3.5.4 PASCAL 2012 action classification

The action classification competition of the 2012 PASCAL VOC challenge [29] contains more than 5,000 training/validation images and a similar number of testing images, which is around 90% increase in size over 2011. We use our proposed method with two improvements.

- Besides the SIFT image descriptor [78] used in the 2011 challenge, we also consider four other descriptors: HOG [16], color naming [118], local binary pattern [89], and object bank [74]. We build decision trees for each feature independently.

- We use training images to build decision trees, and then evaluate the performance of each decision tree on the validation data. A tree that corresponds to

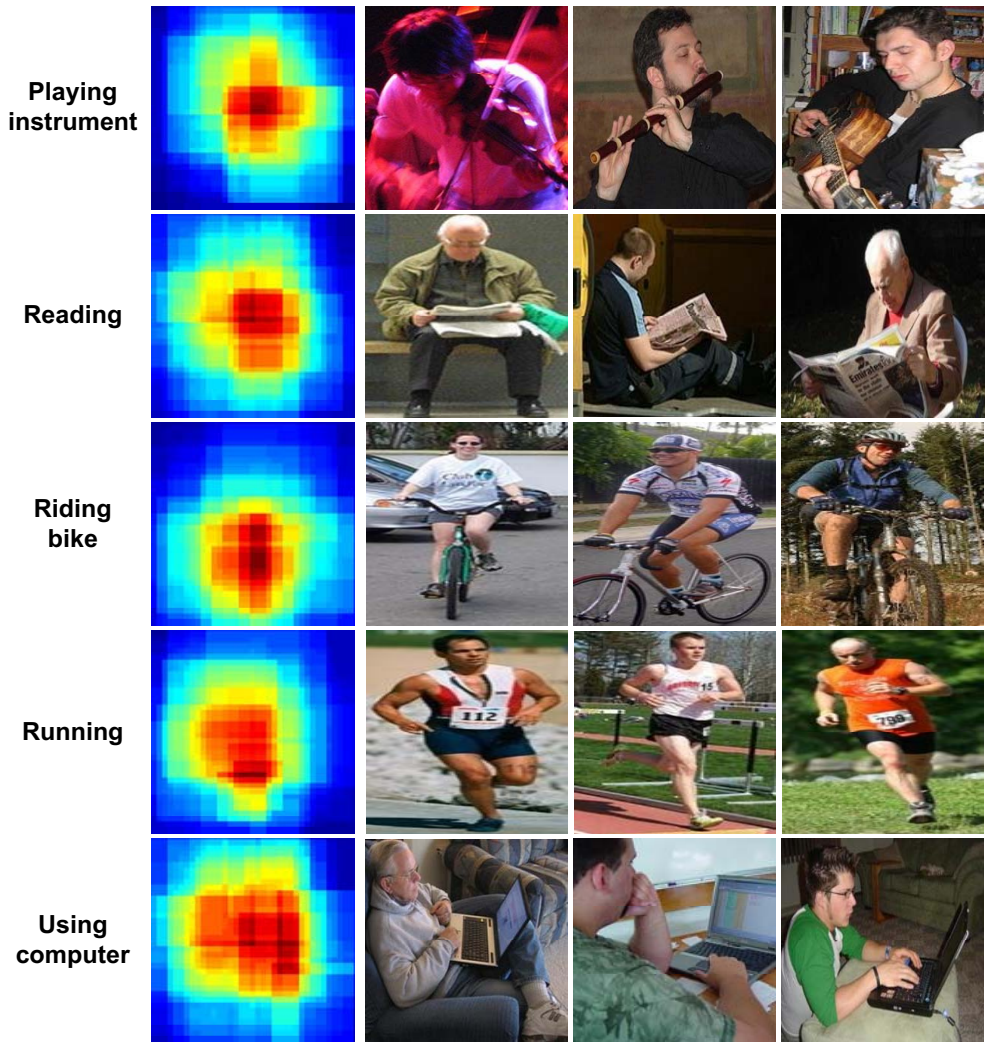


Figure 3.7: Heat maps that show distributions of frequency that an image patch is selected in our method.

a low validation scores will be assigned a low weight. This is different from our 2011 method where all the trees have the weight.

The results are shown in Table 3.5. In 2012 we only have one competitor (DPM_RF_SVM), and our method outperforms this approach on eight out of the ten action classes. Further, comparing "Ours 2012" with "Ours 2011", we can see that combining multiple features and using a tree selection approach improve the performance by 6%.

Action	DPM_RF_SVM	Ours 2011	Ours 2012
Jumping	73.8%	71.1%	**75.7%**
Phoning	**45.0%**	41.2%	44.8%
Playing instrument	62.8%	61.9%	**66.6%**
Reading	41.4%	39.3%	**44.4%**
Riding bike	93.0%	92.4%	**93.2%**
Riding horse	93.4%	92.5%	**94.2%**
Running	**87.8%**	86.1%	87.6%
Taking photo	35.0%	31.3%	**38.4%**
Using computer	64.7%	60.4%	**70.6%**
Walking	73.5%	68.9%	**75.6%**

Table 3.5: Comparison of the mean Average Precision of our method and the other approaches in the action classification competition of PASCAL VOC 2012. "Ours 2011" indicates our approach used for the 2011 challenge. The best results are highlighted with bold fonts.

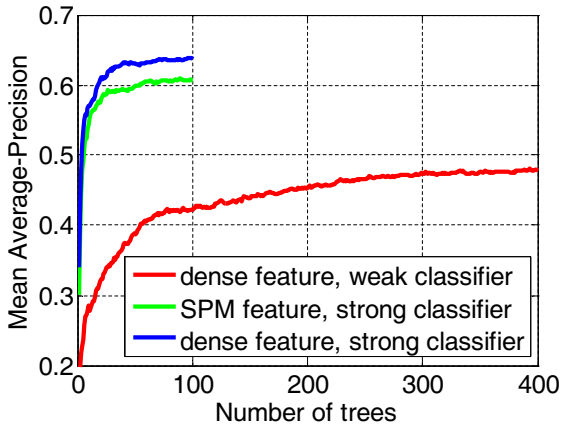
3.5.5 Strength and correlation of decision trees

We compare our method against two control settings of random forests on the PASCAL action dataset. Here we use the PASCAL VOC 2010 dataset [27] where there are fewer images than that on 2011 to make our experiments easier to conduct.

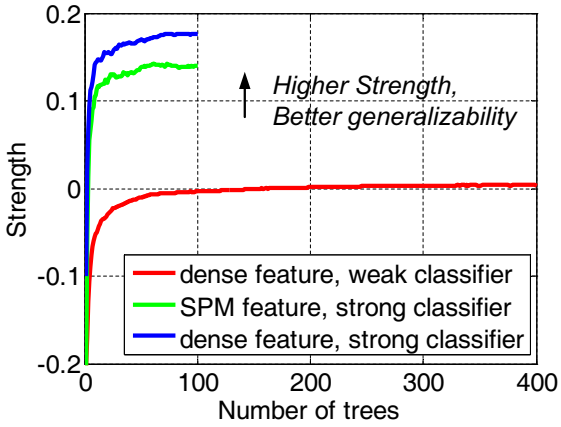
- *Dense feature, weak classifier*: For each image region or pairs of regions sampled from our dense sampling space, replace the SVM classifier in our method with a weak classifier as in the conventional decision tree learning approach [23, 10], i.e. randomly generating 100 sets of feature weights and select the best one.

- *SPM feature, strong classifier*: Use SVM classifiers to split the tree nodes as in our method, but the image regions are limited to that from a 4-level spatial pyramid.

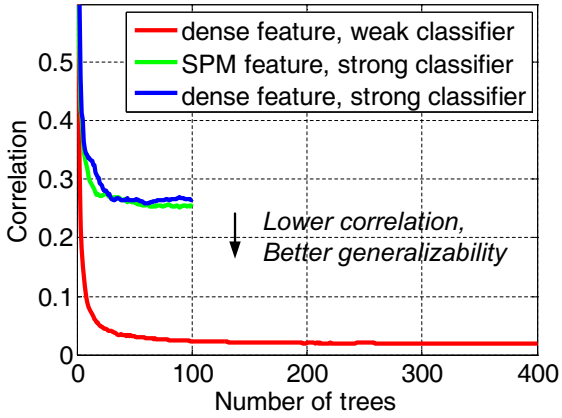
Note that all other settings of the above two approaches remain unchanged as compared to our method (as described in Section 3.4). Figure 3.8 shows that on this dataset, a set of strong classifiers with relatively high correlation can lead to better performance than a set of weak classifiers with low correlation. We can see that the



(a) Mean average precision (mAP).



(b) Strength of the decision trees.



(c) Correlation between the decision trees

Figure 3.8: Comparison of different random forest settings. (a) We compare the classification performance (mAP) obtained by our method dense feature, strong classifier with two control settings. Please refer to Section 3.5.5 for details of these settings. (b,c) We also compare the strength of the decision trees learned by these approaches and correlation between these trees (Section 3.4.4), which are highly related to the generalization error of random forests.

performance of random forests can be significantly improved by using strong classifiers in the nodes of decision trees. Compared to the random forests that only sample spatial pyramid regions, using the dense sampling space obtains stronger trees without significantly increasing the correlation between different trees, thereby improving the classification performance. Furthermore, the performance of the random forests using discriminative node classifiers converges with a small number of decision trees,

indicating that our method is more efficient than the conventional random forest approach. In our experiment, the two settings and our method need a similar amount of time to train a single decision tree.

Additionally, we show the effectiveness of random binary assignment of class labels (Section 3.4.3) when we train classifiers for each tree node. Here we ignore this step and train a one-vs-all multi-class SVM for each sampled image region or pairs of regions. In this case C sets of weights are obtained when there are C classes of images at the current node. The best set of weights is selected using information gain as before. This setting leads to deeper and significantly unbalanced trees, and the performance decreases to 58.1% with 100 trees. Furthermore, it is highly inefficient as it does not scale well with increasing number of classes.

3.6 Summary

In this chapter, we propose a random forest with discriminative decision trees algorithm to explore a dense sampling space for fine-grained image categorization. Experimental results on subordinate classification and activity classification show that our method achieves state-of-the-art performance and discovers much semantically meaningful information.

Chapter 4

Learning Bases of Action Attributes and Parts

The previous two chapters recognize human actions based on low-level image descriptors. In this chapter,[1] we propose to use higher level image representations, including action attributes, objects, and human poses, for action recognition.

4.1 Introduction

As shown in the previous two chapters, a straightforward solution for this problem is to use the whole image to represent an action and treat action recognition as a general image classification problem [59, 126, 18, 132]. Such methods have achieved promising performance on the recent PASCAL challenge using spatial pyramid [71, 18] or random forest [132] based methods. These methods do not, however, explore the semantically meaningful components of an action, such as human poses and the objects that are closely related to the action.

There is some recent work which uses objects [49, 127, 22, 94] interacting with the person or human poses [123, 80] to build action classifiers. However, these methods are prone to problems caused by false object detections or inaccurate pose estimations. To alleviate these issues, some methods [127] rely on labor-intensive annotations of

[1] An early version of this chapter has been presented in [130].

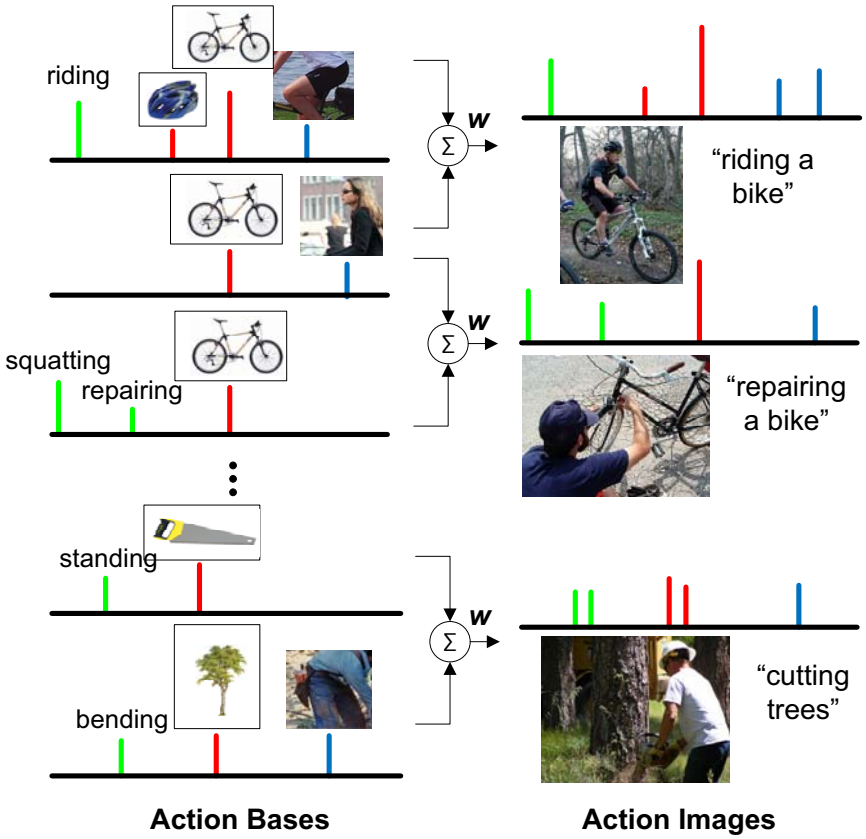


Figure 4.1: We use attributes (verb related properties) and parts (objects and poselets [8]) to model action images. Given a large number of image attributes and parts, we learn a number of sparse action bases, where each basis encodes the interactions between some highly related attributes, objects, and poselets. The attributes and parts of an image can be reconstructed from a sparse weighted summation of those bases. The colored bars indicate different attributes and parts, where the color code is: green - attribute, red - object, blue - poselet. The height of a bar reflects the importance of this attribute or part in the corresponding basis.

objects and human body parts during training time, posing a serious concern towards large scale action recognition.

Inspired by the recent work on using objects and body parts for action recognition as well as global and local attributes [31, 68, 5, 92] for object recognition, in this chapter, we propose an *attributes* and *parts* based representation of human actions in a weakly supervised setting. The action attributes are holistic image descriptions of

human actions, usually associated with verbs in the human language such as "riding" and "sitting" (as opposed to "repairing" or "lifting") for the action "riding bike". The action parts include objects that are related to the corresponding action (e.g. "bike", "helmet", and "road" in "riding bike") as well as different configurations of local body parts (we use poselet described in [8]). Given an image of a human action, many attributes and parts[2] contribute to the recognition of the corresponding action.

Given an image collection of many different actions, there is a large number of possible attributes, objects and poselets. Furthermore, there is a large number of possible interactions among these attributes and parts in terms of co-occurrence statistics. For example, the "riding" attribute is likely to co-occur with objects such as "horse" and "bike", but not "laptop", while the "right arm extended upward" poselet is more likely to co-occur with objects such as "volleyball" and the attribute "hitting". We formulate these interactions of action attributes and parts as *action bases* for expressing human actions. A particular action in an image can therefore be represented as a weighted summation of a subset of these bases, as shown in Figure 4.1.

This representation can be naturally formulated as a reconstruction problem. Our challenge is to: 1) represent each image by using a sparse set of action bases that are meaningful to the content of the image, 2) effectively learn these bases given far-from-perfect detections of action attributes and parts without meticulous human labeling. To resolve these challenges, we propose a *dual sparsity* reconstruction framework to simultaneously obtain sparsity in terms of both the action bases as well as the reconstruction coefficients for each image. We show that our method has theoretical foundations in sparse coding and compressed sensing [141, 61]. To test the performance of our approach, we collected a new dataset "Stanford 40 Actions". On the PASCAL action dataset [27] and the new "Stanford 40 Actions" dataset, our attributes and parts representation significantly outperforms state-of-the-art methods. Furthermore, we visualize the bases obtained by our framework and show semantically meaningful interpretations of the images.

[2]Our definition of action attributes and parts are different from the attributes and parts in common object recognition literature. Please refer to Section 4.2 for details. In this work we use "action attribute" and "attribute", "action part" and "part" interchangeably, if not explicitly specified.

The remaining part of this chapter is organized as follows. Related work are described in Section 4.2. The attributes and parts based representation of actions and the method to learn action bases are elaborated in Section 4.3 and Section 4.4 respectively. The "Stanford 40 Actions" dataset and experiment results are shown and discussed in Section 4.5 and Section 4.6.

4.2 Related work

Most of the action recognition approaches [126, 18, 27] for still images treat the problem as a pure image classification problem. There are also algorithms which model the objects or human poses for action classification, such as the mutual context model [127] and poselets [8, 80]. However, the mutual context model requires supervision of the bounding boxes of objects and human body parts, which are expensive to obtain especially when there is a large number of images. Also, we want to put the objects and human poses in a more discriminative framework so that the action recognition performance can be further improved. While poselets have achieved promising performance on action recognition [80], it is unclear how to jointly explore the semantic meanings of poselets and the other concepts such as objects for action recognition.

In this chapter, we propose to use attributes and parts for action classification. Inspired by the recent work of learning attributes for object recognition [31, 68, 5, 92] and action recognition in videos [76], the attributes we use are linguistically related description of the actions. We use a global image based representation to train a classifier for each attribute. Compared to the attributes for objects which are usually adjectives or shape related, the attributes we use to describe actions are mostly related to verbs. The parts based models have been successfully used in object detection [37] and recognition [39]. However unlike these approaches that use low-level descriptors, the action parts we use are objects and poselets with pre-trained detectors as in [74, 80]. The discriminative information in those detectors can help us alleviate the problem of background clutter in action images and give us more semantic information of the images [74].

In the attributes and parts based representation, we learn a set of sparse action

bases and estimate a set of coefficients on these bases for each image. This dual sparsity makes our problem different from traditional dictionary learning and sparse coding problems [113, 72, 79], given that our action bases are sparse (in the large set of attributes and parts, only a small number of them are highly related in each basis) and far from being mutually orthogonal (consider the two bases "riding - sitting - bike" and "riding - sitting - horse"). In this work, we solve this dual sparsity problem using the elastic-net constrained set [141], and show that our approach has theoretical foundations in the compressed network theorem [61].

4.3 Action recognition with attributes & parts

4.3.1 Attributes and parts in human actions

Our method jointly models different attributes and parts of human actions, which are defined as follows.

Attributes: The attributes are linguistically related descriptions of human actions. Most of the attributes we use are related to verbs in human language. For example, the attributes for describing "riding a bike" can be "riding" and "sitting (on a bike seat)". It is possible for one attribute to correspond to more than one action. For instance, "riding" can describe both "riding a bike" and "riding a horse", while this attribute can differentiate the intentions and human gestures in the two actions with the other ones such as "drinking water". Inspired by the previous work on attributes for object recognition [31, 68, 5], we train a discriminative classifier for each attribute.

Parts: The parts we use are composed of objects and human poses. We assume that an action image consists of the objects that are closely related to the action and the descriptive local human poses. The objects are either manipulated by the person (e.g. "bike" in "riding a bike") or related to the scene context of the action (e.g. "road" in "riding a bike", "reading lamp" in "reading a book"). The human poses are represented by poselets [8], where the human body parts in different images described by the same poselet are tightly clustered in both appearance space and configuration space. In our approach, each part is modeled by a pre-trained object

detector or poselet detector.

To obtain our features, we run all the attribute classifiers and part detectors on a given image. A vector of the normalized confidence scores obtained from these classifiers and detectors is used to represent this image.

4.3.2 Action bases of attributes and parts

Our method learns high-order interactions of image attributes and parts. Each interaction corresponds to the co-occurrence of a set of attributes and parts with some specific confidence values (Figure 4.1). These interactions carry richer information about human actions and are thus expected to improve recognition performance. Furthermore, the components in each high-order interaction can serve as context for each other, and therefore the noise in the attribute classifiers and part detectors can be reduced. In our approach, the high-order interactions are regarded as the bases of the representations of human actions, and each image is represented as a sparse distribution with respect to all the bases. Examples of the learned action bases are shown in Figure 4.5. We can see that the bases are sparse in the whole space of attributes and parts, and many of the attributes and parts are closely correlated in human actions, such as "riding - sitting - bike" and "using - keyboard - monitor - sitting" as well as the corresponding poselets.

Now we formalize the action bases in a mathematical framework. Assume we have P attributes and parts, and let $\mathbf{a} \in \mathbb{R}^P$ be the vector of confidence scores obtained from the attribute classifiers and part detectors. Denoting the set of action bases as $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_M]$ where each $\boldsymbol{\phi}_m \in \mathbb{R}^P$ is a basis, the vector $\mathbf{a}$ can be represented as

$$\mathbf{a} = \sum_{m=1}^{M} w_m \boldsymbol{\phi}_m + \boldsymbol{\varepsilon} \tag{4.1}$$

where $\mathbf{w} = \{w_1, \cdots, w_M\}$ are the reconstruction coefficients of the bases, and $\boldsymbol{\varepsilon} \in \mathbb{R}^P$ is a noise vector. Note that in our problem, the vector $\mathbf{w}$ and $\{\boldsymbol{\phi}_m\}_{m=1}^M$ are all sparse. This is because on one hand, only a small number of attributes and parts are highly related in each basis of human actions; on the other hand, a small proportion of the

action bases are enough to reconstruct the set of attributes and parts in each image.

4.3.3 Action classification using the action bases

From Eqn.4.1, we can see that the attributes and parts representation $\mathbf{a}$ of an action image can be reconstructed from the sparse factorization coefficients $\mathbf{w}$. $\mathbf{w}$ reflects the distribution of $\mathbf{a}$ on all the action bases $\mathbf{\Phi}$, each of which encodes a specific interaction between action attributes and parts. The images that correspond to the same action should have high coefficients on the similar set of action bases. In this chapter, we use the coefficients vector $\mathbf{w}$ to represent an image, and train an SVM classifier for action classification.

The above classification approach resolves the two challenges of using attributes and parts (objects and poselets) for action recognition that we proposed in Section 4.1. Since we only use the learned action bases to reconstruct the feature vector, our method *can correct some false detections of objects and poselets* by removing the noise component ε in Eqn.4.1. Also, those action bases correspond to some high-order interactions in the features, and therefore they *jointly model the complex interactions between different attributes, objects, and poselets.*

4.4 Learning action bases and coefficients

Given a collection of training images represented as $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N\}$ as described in Section 4.3.2, where each $\mathbf{a}_i$ is the vector of confidence scores of attribute classifications and part detections computed from image i. Intuitively, there exists a latent dictionary of bases where each basis characterizes frequent co-occurrence of attributes, objects, and poselets involved in an action, e.g. "cycling" and "bike", such that each observed data $\mathbf{a}_i$ can be sparsely reconstructed with respect to the dictionary. Our goal is to identify a set of sparse bases $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_M]$ such that each $\mathbf{a}_i$ has a sparse representation with respect to the dictionary, as shown in Eqn.4.1.

During the bases learning stage, we need to learn the bases $\mathbf{\Phi}$ and find the reconstruction coefficients $\mathbf{w}_i$ for each $\mathbf{a}_i$. This is achieved by

$$\min_{\mathbf{\Phi}\in\mathcal{C},\mathbf{w}_i\in\mathbb{R}^M} \sum_{i=1}^{N} \Big(\frac{1}{2}\|\mathbf{a}_i - \mathbf{\Phi}\mathbf{w}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_1\Big), \tag{4.2}$$

$$\mathcal{C} = \{\mathbf{\Phi} \in \mathbb{R}^{P\times M}, \text{s.t. } \forall j,\ \|\mathbf{\Phi}_j\|_1 + \frac{\gamma}{2}\|\mathbf{\Phi}_j\|_2^2 \leq 1\}, \tag{4.3}$$

where $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_N] \in \mathbb{R}^{M\times N}$, λ and γ are regularization parameters, and $\mathcal{C}$ is the convex set that $\mathbf{\Phi}$ belongs to. The l_1-norm regularization makes the reconstruction coefficients $\mathbf{w}_i$ tend to be sparse.

In our setting, the bases $\mathbf{\Phi}$ should also be sparse, even though the given $\mathcal{A}$ might be quite noisy due to the error-prone object detectors and poselet detectors. To address this issue, we construct the convex set $\mathcal{C}$ as in 4.3. Including both l_1-norm and l_2-norm to define the convex set $\mathcal{C}$, the sparsity requirement of the bases are encoded. This is called the *elastic-net constraint set* [141]. Furthermore, the sparsity on $\mathbf{\Phi}$ implies that different action bases have small overlaps, therefore the coefficients learned from Eqn.4.2 are guaranteed to generalize to the testing case, according to the compressed network theorem [61].

Given a new image represented by $\mathbf{a}$, we want to find a sparse $\mathbf{w}$ such that $\mathbf{a}$ can be reconstructed from the learned $\mathbf{\Phi}$. This is achieved by

$$\min_{\mathbf{w}\in\mathbb{R}^M} \frac{1}{2}\|\mathbf{a} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1. \tag{4.4}$$

The dual sparsity on both action bases and reconstruction coefficients in Eqn.4.2 and Eqn.4.3 also enables the uniqueness of the attributes and parts reconstruction in the testing step (Eqn.4.4). Uniqueness is important in the Lasso problem especially when we look for interpretable bases for action recognition. Otherwise if the solution for the problem is not unique, one might reconstruct the attributes and parts of an action image from other confusing bases which also optimize our objective but are totally irrelevant to the action in the image.

It has been shown that the ℓ_1-norm minimization problem has a unique sparse solution, if the basis matrix satisfies the so-called *Restricted Isometry Property (RIP)*

condition, which requires that every subset of columns in the support of the sparse signal are nearly orthogonal [11]. In [140], the *Irrepresentable Condition* (*IRR*) was proposed for stably recovering a sparse signal $\mathbf{w}^*$ by solving the Lasso problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{a} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1. \tag{4.5}$$

The basis matrix $\mathbf{\Phi}$ satisfies the IRR condition with respect to $S = \{\forall j, w_j^* \neq 0\}$, if $\mathbf{\Phi}_S^T\mathbf{\Phi}_S$ is invertible and

$$\|\mathbf{\Phi}_{S^c}^T\mathbf{\Phi}_S \left(\mathbf{\Phi}_S^T\mathbf{\Phi}_S\right)^{-1}\|_\infty < 1. \tag{4.6}$$

where $\mathbf{\Phi}_S$ is a sub-matrix of $\mathbf{\Phi}$ with S selecting the columns, $\mathbf{\Phi}_S^T$ is the transpose of $\mathbf{\Phi}_S$, S^c is the complement of S.

The IRR condition does not hold for general matrices. However, it has been shown that when the basis matrix $\mathbf{\Phi}$ is sparse, it turns out that IRR still holds in many different situations [61]. Please refer to [61] for further materials explaining the conditions to guarantee the success of solving the Lasso problem. In our problem, we impose sparsity on $\mathbf{\Phi}$ so that a unique sparse solution of $\mathbf{w}$ can be obtained for most of the vectors $\mathbf{a}$.

In our two optimization problems, Eqn.4.4 is convex while Eqn.4.2 is non-convex. However Eqn.4.2 is convex with respect to each of the two variables $\mathbf{\Phi}$ and $\mathbf{W}$ when the other one is fixed. We use an online learning algorithm [79] which scales up to large datasets to solve this problem.

4.5 Stanford 40 Actions dataset

To test the performance of action recognition on more categories of actions, we collected a new dataset called *Stanford 40 Actions*[3]. The dataset contains 40 diverse daily human actions, such as "brushing teeth", "cleaning the floor", "reading book", "throwing a frisbee", etc. A summary of the dataset is shown in Table 4.1. All the

[3]The dataset can be downloaded from `http://vision.stanford.edu/Datasets/40actions.html`.

Action Name	# imgs	Action Name	# imgs
Applauding	279	Reading book	234
Blowing bubbles	292	Repairing a car	122
Brushing teeth	211	Riding a bike	288
Calling	272	Riding a horse	260
Cooking	295	Rowing a boat	149
Cutting trees	175	Running	254
Cutting vegetables	131	Shooting an arrow	211
Drinking	194	Smoking cigarette	175
Feeding a horse	319	Taking photos	154
Fishing	269	Throwing a frisby	195
Fixing a bike	131	Using a computer	230
Filling up gas	123	Using a microscope	127
Hanging clothes	121	Using a telescope	151
Holding an umbrella	289	Using an ATM	144
Jumping	299	Walking a dog	294
Mopping the floor	159	Washing dishes	183
Playing guitar	295	Watching TV	146
Playing violin	268	Waving hands	209
Poling a boat	118	Writing on a board	133
Pushing a cart	172	Writing on a book	127

Table 4.1: The Stanford 40 Actions dataset: the list of actions and number of images in each action.

Dataset	# actions	# images	Clutter?	Pose vary?	Visib. vary?
Ikizler [59]	5	1,727	**Yes**	**Yes**	**Yes**
Gupta [49]	6	300	Small	Small	No
PPMI [126]	**24**	4,800	**Yes**	**Yes**	No
PASCAL [29]	10	10,595	**Yes**	**Yes**	**Yes**
Stanford 40	**40**	**9,532**	**Yes**	**Yes**	**Yes**

Table 4.2: Comparison of our Stanford 40 Action dataset and other existing human action datasets on still images. Bold font indicate relatively larger scale datasets or larger image variations.

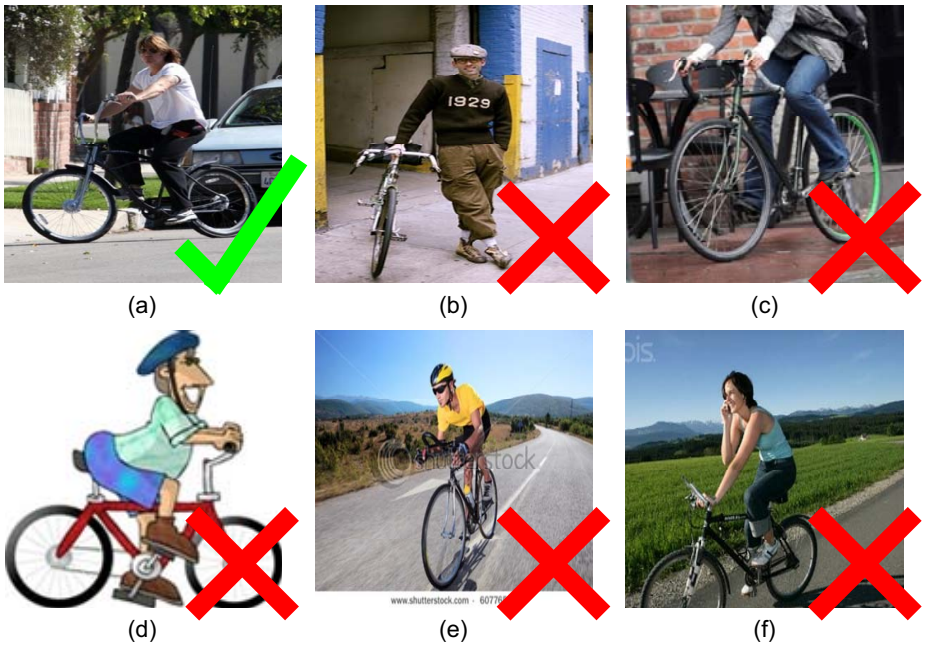


Figure 4.2: The criteria of collecting images for the Stanford 40 Actions dataset. In the case of "riding a bike", we want to collect images such as (a). The other images are not satisfying because: (b) the human is not riding the bike; (c) the human's head is totally outside of the image; (d) it is a cartoon image; (e) it is an advertisement image and texts are placed on the image; (f) the human is riding a bike while making a phone call.

images are obtained from Google, Bing, and Flickr. The images within each class have large variations in human pose, appearance, and background clutter. The comparison between our dataset and the existing still image action datasets are summarized in Table 4.2, where visibility variation refers to the variation of visible human body parts (e.g. in some images the full human body is visible, while in some other images only the head and shoulder are visible). As there might be multiple people in a single image, we provide bounding boxes for the humans who are doing one of the 40 actions in each image, similar to [27].

The images are collected in the following procedure. For each action, we first use some keywords to crawl as many images as we can from Google, Bing, and Flickr. Instead of only using the action name as the keyword, we also consider some other

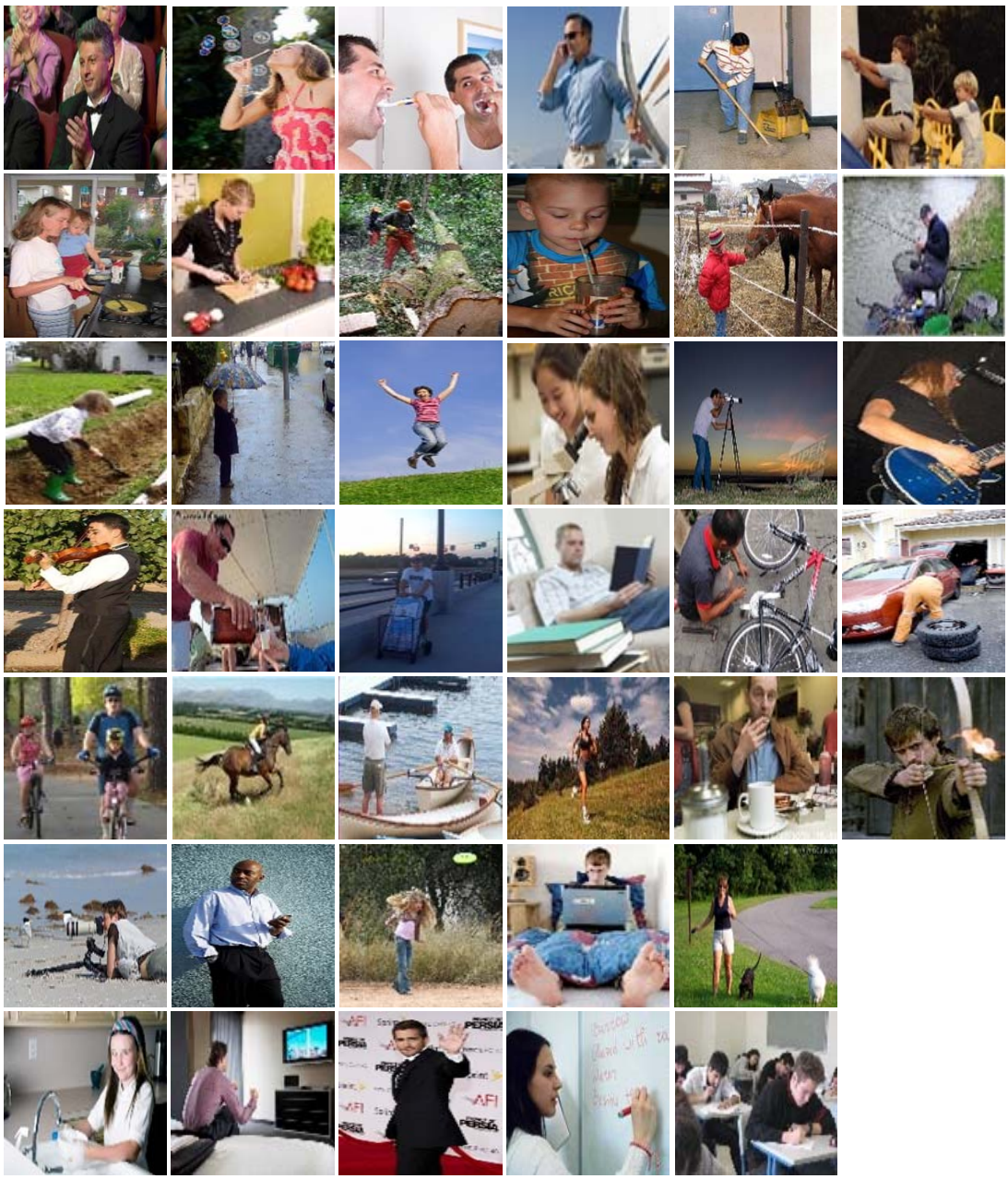


Figure 4.3: Example images of the Stanford 40 Actions Dataset.

keywords which we believe can collect more images of the corresponding action. For example, the query keywords we use for "watching TV" is: "watching television", "man watching TV", "woman watching TV", "family watching TV", and "children

watching TV". We can crawl 10,000+ images for each class. Then, we select the desired images from the crawling results in each class. As demonstrated in Figure 4.2, the selection criteria are: (1) the human should be doing the corresponding action; (2) the human's head needs to be visible; (3) the image is not a cartoon image; (4) the image should not be significantly edited (e.g. with many texts on it); (5) the human should not be doing more than one of our 40 actions (e.g. pushing a cart while calling). The next step is to de-duplicate the selected images by a simple color histogram matching method. Finally, we check the remaining images and further manually remove some images to guarantee the image diversity within each class. Examples of the images in our dataset are shown in Figure 4.3.

4.6 Experiments and results

4.6.1 Experiment setup

We test the performance of our proposed method on the PASCAL action dataset [28] and the Stanford 40 Actions dataset.

On the PASCAL dataset, we use the training and validation set specified in [28] for training, and use the same testing set. On the Stanford 40 Action dataset, we randomly select 100 images in each class for training, and the remaining images for testing. For each dataset, we annotate the attributes that can be used to describe the action in each image, and then train a binary classifier for each attribute. We take a global representation of the attributes as in [31], and use the Locality-constrained Linear Coding (LLC) method [120] on dense SIFT [78] features to train the classifier for each attribute. As in [18], the classifiers are trained by concatenating the features from both the foreground bounding box of the action and the whole image. We extend and normalize the bounding boxes in the same way as in [18]. For objects, we use the ImageNet [20] dataset with provided bounding boxes to train the object detectors by using the Deformable Parts Model [37], instead of annotating the positions of objects in the action data. For poselets, we use the pre-trained poselet detectors in [8]. For each object or poselet detector, we use the highest detection score in the response

map of each image to measure the confidence of the object or poselet in the given image. We linearly normalize the confidence scores of all the attribute classifiers and part detectors so that all the feature values are between 0 and 1.

We use 15 attributes and 27 objects for the PASCAL data, 45 attributes and 81 objects for the Stanford 40 Action data. We only use the attributes and objects that we believe are closely related to the actions in each dataset. A full list of attributes and objects that we use are:

- attributes for PASCAL: calling, playing, reading, riding, running, taking, using, walking, cycling, jumping, standing, sitting, squatting, lying, and moving.

- objects for PASCAL: beach, bicycle, bicycle built for two, camcorder, camera, cello, cellular telephone, computer, computer keyboard, desktop computer, dial telephone, flute, grass, guitar, keyboard, laptop, monitor, motorcycle, musical instrument, newspaper, notebook, pay phone, piano, skyscraper, telephone, and violin.

- attributes for 40-class: applauding, bending, blowing, brushing, calling, cooking, cutting, cycling, drinking, feeding, fishing, fixing, filling, hanging, holding, jumping, looking through, lying, mopping, playing, poling, pulling, pushing, reading, repairing, riding, rowing, running, shooting, singing, sitting, smoking, speaking, standing, squatting, taking, throwing, typing, using, walking, watching, waving, wearing, withdrawing, and writing.

- objects for 40-class: African hunting dog, Eskimo dog, Polaroid camera, beach, beer, beer bottle, beer glass, bicycle, bicycle built for two, blackboard, boat, boathouse, bow, bowl, broom, bulldog, camcorder, camera, car-12982, car-1527, car-1634, car tire, coat, computer, computer keyboard, cup, cuppa, desktop computer, dog, fish, fishing rod, gas pump, glass, golden retriever, grass, guitar, hand-held computer, handcart, laptop, laundry cart, male horse, motorcycle, mountain bike, mug, newspaper, notebook, optical telescope, passenger car, point-and-shoot camera, radio telescope, sheet, shopping cart, sky, streetcar, television, violin, washbasin, washer, and wheel.

The attributes for all the training images are annotated by us. The objects we consider are limited to the classes that have annotated bounding boxes in ImageNet [20]. For instance, "car-12982", "car-1527", and "car-1634" are three different cars in ImageNet. Cigarette is helpful for recognizing the action of "smoking cigarette", but cigarette is not included because there is no cigarette bounding boxes in ImageNet. We use 150 poselets as provided in [8] on both datasets. The number of action bases are set to 400 and 600 respectively. The λ and γ values in Eqn.4.2, 4.4, and 4.3 are set to 0.1 and 0.15.

In the following experiment, we consider two approaches of using attributes and parts for action recognition. One is to simply concatenate the normalized confidence scores of attributes classification and parts detection as feature representation (denoted as "Conf_Score"), the other is to use the reconstruction coefficients on the learned sparse bases as feature representation (denoted as "Sparse_Bases"). We use linear SVM classifiers for both feature representations. As in [28], we use mean Average Precision (mAP) to evaluate the performance on both datasets.

4.6.2 Results on the PASCAL action dataset

The action classification competition of PASCAL VOC has two tasks. The results reported in the previous Chapter is comp9 where the classifiers must be trained by using only the images provided by the organizers. In comp10, participants are free to train their classifiers on any dataset or use additional annotations.

We participated the comp10 of PASCAL VOC 2011, and the average precision of different approaches is shown in Table 4.3. We can see that by simply concatenating the confidence scores of attributes classification and parts detection, our method outperforms the best result in the PASCAL challenge in terms of the mean Average Precision (mAP). The performance can be further improved by learning high-order interactions of attributes and parts, from which the feature noise can be reduced. A visualization of the learned bases of our method is shown in Figure 4.5. We observe that almost all the bases are very sparse, and many of them carry useful information for describing specific human actions. However due to the large degree of noise in

Action	ACTION_ POSELETS	MAPSVM_ POSELET	Ours DIS_RF	Ours ATTR_PRT
Jumping	59.5%	27.0%	66.0%	**66.7%**
Phoning	31.3%	29.3%	41.0%	**41.1%**
Playing instrument	45.6%	28.3%	60.0%	**60.8%**
Reading	27.8%	23.8%	41.5%	**42.2%**
Riding bike	84.4%	71.9%	90.0%	**90.5%**
Riding horse	88.3%	82.4%	92.1%	**92.2%**
Running	77.6%	67.3%	**86.6%**	86.2%
Taking photo	**31.0%**	20.1%	28.8%	28.8%
Using computer	47.4%	26.0%	62.0%	**63.5%**
Walking	57.6%	46.4%	**65.9%**	64.2%

Table 4.3: Comparison of our method and the approaches in comp10 of PASCAL VOC 2012. We also compare with our winning method in comp9 (described in the previous chapter). Performance is evaluated in mean average precision. Each column represents an approach. The best results are highlighted with bold fonts.

both object detectors and poselet detectors, some bases contain noise, e.g. "guitar" in the basis of "calling - cell phone - guitar". In Figure 4.6 we show some action images with the annotations of attributes and objects that have high confidence score in the feature representation reconstructed from the bases.

Our approach considers three concepts: attributes, parts as objects, and parts as poselets. To analyze the contribution of each concept, we remove the confidence scores of attribute classifiers, part detectors, and poselet detectors from our feature set, one at a time. The classification results are shown in Figure 4.4. We observe that using the reconstruction coefficients consistently outperform the methods that simply concatenating the confidence scores of classifiers and detectors. We can also see that attributes make the biggest contribution to the performance, because removing the attribute features makes the performance much worse. This is due to the large amount of noise produced from objects and poselets detectors which are pre-trained from the other datasets. However, objects and poselets do contain complementary information with the attributes, and the effect of the noise can be alleviated by the bases learned from our approach. We observe that in the case of only considering objects and poselets, learning the sparse bases significantly improves the performance.

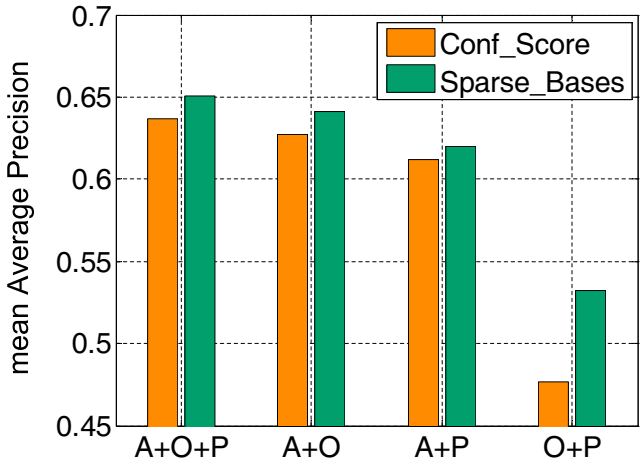


Figure 4.4: Comparison of the methods by removing the confidence scores obtained from attributes (A), objects (O), and poselets (P) from the feature vector, one at a time. The performance are evaluated using mean Average Precision on the PASCAL dataset.

By combining attributes, objects and poselets and learning the action bases, our method achieves state-of-the-art classification performance.

Our learning method (Eqn.4.2) has the dual sparsity on both action bases $\mathbf{\Phi}$ and reconstruction coefficients $\mathbf{W}$. Here we compare our method with a simple l_1-norm method - l_1 logistic regression based on the concatenation of the confidence scores of attributes and parts. The mAP result of l_1 logistic regression is 47.9%, which is lower than our results. This shows that a simple l_1-norm logistic regression cannot effectively learn the information from the noisy attributes classification and parts detection features. Furthermore, in order to demonstrate the effectiveness of the two sparsity constraints, we remove the constraints one at a time. To remove the sparsity constraint on the reconstruction weight $\mathbf{W}$, we simply change $\|\mathbf{w}_i\|_1$ in Eqn.4.2 and Eqn.4.4 to $\|\mathbf{w}_i\|_2$. To remove the sparsity constraint on the bases $\mathbf{\Phi}$, we change the convex set $\mathcal{C}$ in Eqn.4.3 to be:

$$\mathcal{C} = \{\mathbf{\Phi} \in \mathbb{R}^{P\times M}, \text{s.t. } \forall j,\ \|\mathbf{\Phi}_j\|_2^2 \leq 1\}. \tag{4.7}$$

In the first case, where we do not have sparsity constraint on $\mathbf{W}$, the mAP result drops

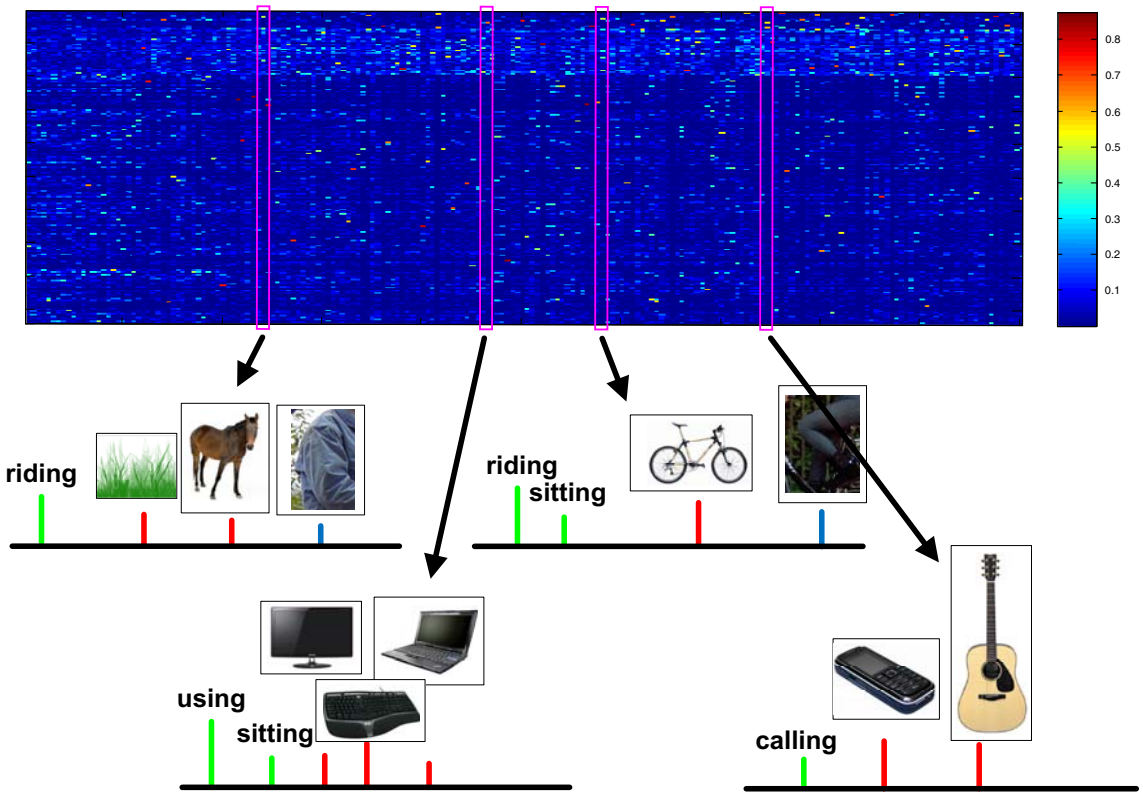


Figure 4.5: Visualization of the 400 learned bases from the PASCAL action dataset. Each column in the top-most matrix corresponds to one basis. Red color indicates large magnitude in the action bases while blue color indicates low magnitude. We observe that the bases are indeed very sparse. We also show some semantically meaningful action bases learned by our results, e.g. "riding - grass - horse". By using the learned action bases to reconstruct the attributes and parts representation, we show the attributes and objects that have high confidence scores on some images. Magenta color indicates wrong tags.

to 64.0%, which is comparable to directly concatenating all attributes classification and parts detection confidence scores. This shows that the sparsity on $\mathbf{W}$ helps to remove noise from the original data. In the second case where we do not have sparsity constraint on $\mathbf{\Phi}$, the performance becomes 64.7% which is very close to that of having sparsity constraint on $\mathbf{\Phi}$. The reason might be that although there is much noise in the parts detections and attribute classifications, the original vector of confidence scores

Figure 4.6: Some semantically meaningful action bases learned by our results, e.g. "riding - grass - horse". By using the learned action bases to reconstruct the attributes and parts representation, we show the attributes and objects that have high confidence scores on some images. Magenta color indicates wrong tags.

already has some level of sparsity. However, by explicitly imposing the sparsity on $\mathbf{\Phi}$,

Method	Object Bank [74]	LLC [120]	Ours Conf_Score	Ours Sparse_Bases
mAP	32.5%	35.2%	44.6%	**45.7%**

Table 4.4: Comparison of our attributes and parts based action recognition methods with two baselines: object bank [74] and LLC [120]. The performance is evaluated with mean average precision (mAP). The bold font indicates the best performance on this dataset.

we can guarantee the sparsity of the bases, so that our method can explicitly extract more semantic information and its performance is also theoretically guaranteed.

4.6.3 Results on the Stanford 40 Actions dataset

We next show the performance of our proposed method on the new Stanford 40 Actions dataset (details of the dataset in Section 4.5). We setup two baselines on this dataset: LLC [120] method with densely sampled SIFT [18] features, and object bank [74]. Comparing these two algorithms with our approach, the mAP is shown in Table 4.4. The results show that compared to the baselines which uses image classifiers or object detectors only, combining attributes and parts (objects and poselets) significantly improved the recognition performance by more than 10%. The reason might be that, on this relatively large dataset, more attributes are used to describe the actions and more objects are related to the actions, which contains a lot of complementary information.

As done in Section 4.6.2, we also remove the features that are related to attributes, objects, and poselets from our feature set, one at a time. The results are shown in Figure 4.7. On this dataset, the contribution of objects is larger than that on the PASCAL dataset. This is because more objects are related to the actions on this larger scale dataset, and therefore we can extract more useful information for recognition from the object detectors.

The average precision obtained from LLC and our method by using reconstruction coefficients as feature representation for each of the 40 classes is shown in Figure 4.8. Using a sparse representation on the action bases of attributes and parts, our method

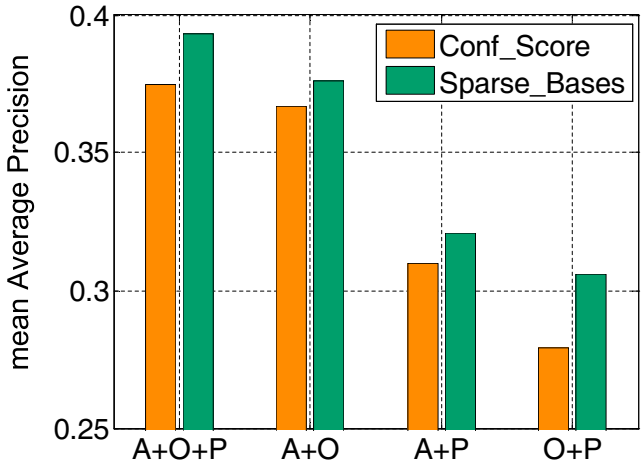


Figure 4.7: Comparison of the methods by removing the confidence scores obtained from attributes (A), objects (O), and poselets (P) from the feature vector, one at a time. The performance is evaluated using mean Average Precision on the Stanford 40 Actions dataset.

outperforms LLC on all the 40 classes. Furthermore, the classification performance on different actions varies a lot, ranging from 89.2% on "riding a horse" to only 6.2% on "texting message". It is interesting to observe that the result shown in Figure 4.8 is somewhat similar to that on the PASCAL dataset in Table 4.3. The classes "riding a horse" and "riding a bike" have high classification performance on both datasets while the classes "calling", "reading a book" and "taking photos" have low classification performance, showing that the two datasets capture similar image statistics of human actions. The classes "riding a horse" and "riding a bike" can be easily recognized in part because the human poses do not vary much within each action, and the objects (horse and bike) are easy to detect. However, the performance on "feeding a horse" and "repairing a bike" is not as good as that on "riding a horse" and "riding a bike". One reason is that the body parts of horses in most of the images of "feeding a horse" are highly occluded, and therefore the horse detector is difficult to detect them. From the images of "repairing a bike", we can see that the human pose changes a lot and the bikes are also occluded or disassembled, making them difficult to be recognized by bike detectors. There are some classes on which the recognition performance is

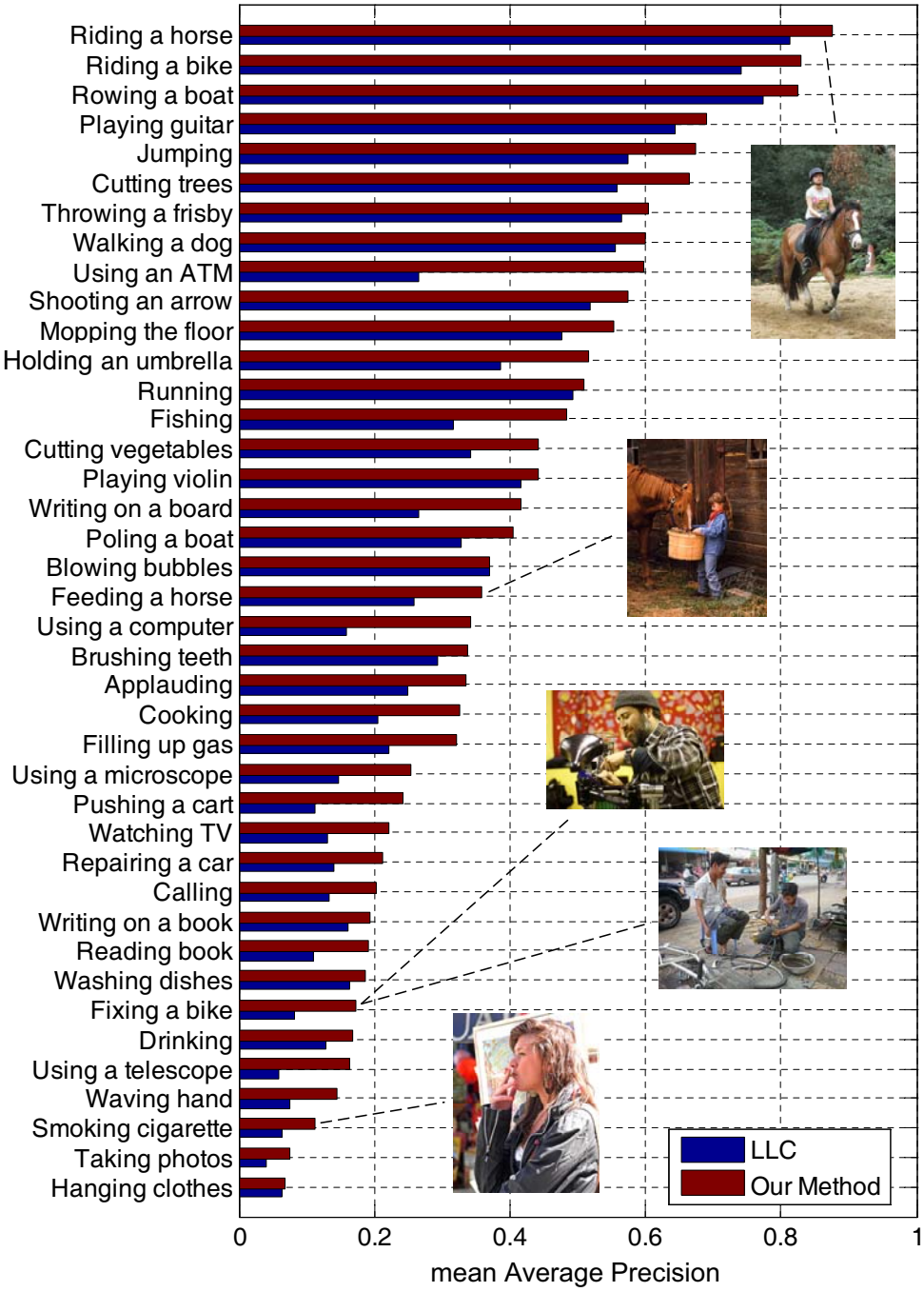


Figure 4.8: Average precision of our method (Sparse_Bases) on each of the 40 classes of the Stanford 40 Actions dataset. We compare our method with the LLC algorithm.

very low, e.g. "taking photos". The reason is that the cameras are very small, which makes it difficult to distinguish "taking photos" and the other actions.

4.7 Summary

In this chapter, we use attributes and parts for action recognition. The attributes are verbs related description of human actions, while the parts are composed of objects and poselets. We learn a set of sparse bases of the attributes and parts based image representation, allowing an action image to be reconstructed by a set of sparse coefficients with respect to the bases. Experimental results show that our method achieves state-of-the-art performance on two datasets.

Experimental results on Section 4.6.2 show that using attributes and parts can improve action recognition performance, even compared with the discriminative random forest method discussed in Chapter 3. One disadvantage of the method discussed in this chapter is that it requires additional annotations of action attributes, object bounding boxes, and human body parts, while the random forest method only needs annotations of class labels. But the method still generalizes well on large scale datasets. This is because all the classifiers and detectors can be trained offline, and theus it is not necessary to re-train them on new datasets.

Chapter 5

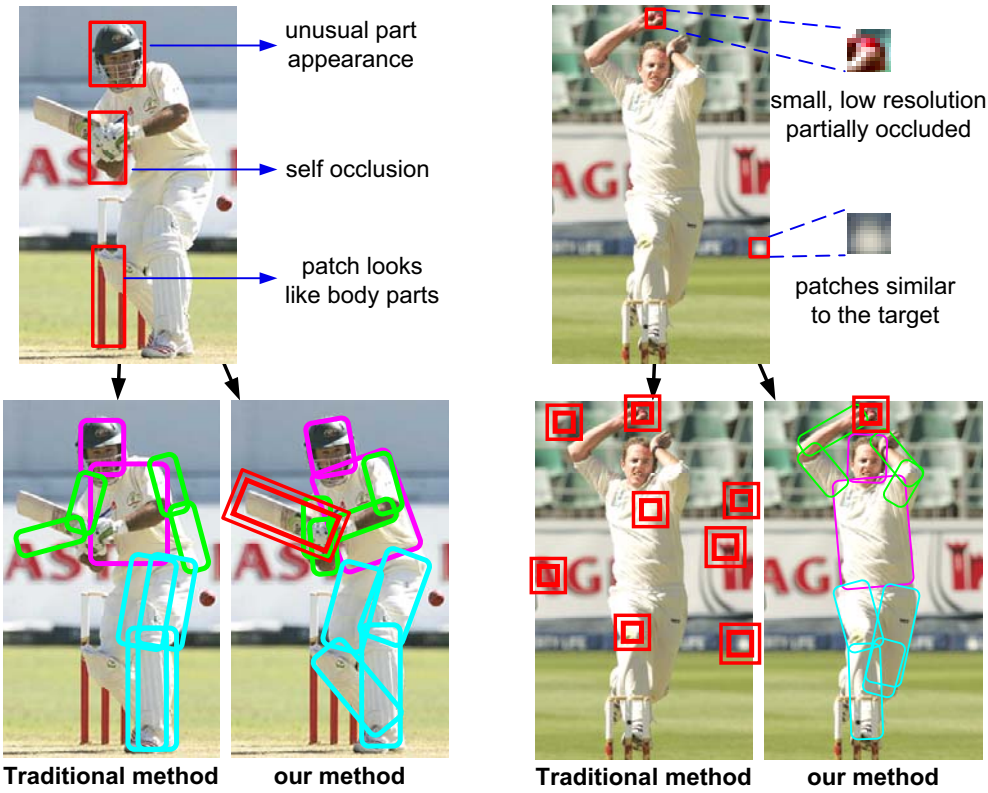
Mutual Context Model I: Single Object

In the preceding three chapters, we treat action recognition as an image classification task. In Chapter 4, we show that using high-level cues such as human pose and object helps improving action recognition performance. Indeed, estimating human pose and detecting objects provide more detailed understanding of human actions.[1]

5.1 Introduction

Using context to aid visual recognition is recently receiving more and more attention. Psychology experiments show that context plays an important role in recognition in the human visual system [6, 91]. In computer vision, context has been used in problems such as object detection and recognition [96, 52, 24], scene recognition [84], action classification [81], and segmentation [108]. While the idea of using context is clearly a good one, a curious observation shows that most of the context information has contributed relatively little to boost performances in recognition tasks. In the recent Pascal VOC challenge dataset [27], the difference between context based methods and sliding window based methods for object detection (e.g. detecting bicycles) is only within a small margin of $3-4\%$ [21, 51].

[1] An early version of this chapter has been presented in [127].



(a) human pose estimation (b) object (ball) detection

Figure 5.1: Objects and human poses can serve as mutual context to facilitate the recognition of each other. In **(a)**, the human pose is better estimated by seeing the cricket bat, from which we can have a strong prior of the pose of the human. In **(b)**, the cricket ball is detected by understanding the human pose of throwing the ball.

One reason to account for the relatively small margin is, in our opinion, the lack of strong context. While it is nice to detect cars in the context of roads, powerful car detectors [73] can nevertheless detect cars with high accuracy whether they are on the road or not. Indeed, for the human visual system, detecting visual abnormality out of context is crucial for survival and social activities (e.g. detecting a cat in the fridge, or an unattended bag in the airport) [53].

So is context oversold? Our answer is 'no'. Many important visual recognition tasks rely critically on context. One such scenario is the problem of human pose estimation and object detection in human-object interaction (HOI) activities [49, 126]. As shown in Figure 5.1, without knowing that the human is making a defensive

shot with the cricket bat, it is not easy to accurately estimate the player's pose (Figure 5.1(a)); similarly, without seeing the player's pose, it is difficult to detect the small ball in the player's hand, which is nearly invisible even to the human eye (Figure 5.1(b)).

However, the two difficult tasks can benefit greatly from serving as context for each other, as shown in Figure 5.1. The goal of this chapter is to model the *mutual context* of objects and human poses in HOI activities so that each can facilitate the recognition of the other. Given a set of training images, our model automatically discovers the relevant poses for each type of HOI activity, and furthermore the connectivity and spatial relationships between the objects and body parts. We formulate this task as a structure learning problem, of which the connectivity is learned by a structure search approach, and the model parameters are discriminatively estimated by a novel max-margin approach. By modeling the mutual co-occurrence and spatial relations of objects and human poses, we show that our algorithm significantly improves the performance of both object detection and pose estimation on a dataset of sports images [49].

The rest of this chapter is organized as follows. Section 5.2 describes related work. Details of our model, as well as model learning and inference are elaborated in Section 5.3, 5.4, and 5.5 respectively. Experimental results are given in Section 5.6.

5.2 Related work

The two central tasks, human pose estimation and object detection, have been studied in computer vision for many years. Most of the pose estimation work uses a tree structure of the human body [38, 98, 2] which allows fast inference. In order to capture more complex body articulations, some non-tree models have also been proposed [100, 121]. Although those methods have been demonstrated to work well on the images with clean backgrounds, human pose estimation in cluttered scenes remains a challenging problem. Furthermore, to our knowledge, no existing method has explored context information for human pose estimation.

Sliding window is one of the most successful strategies for object detection. Some

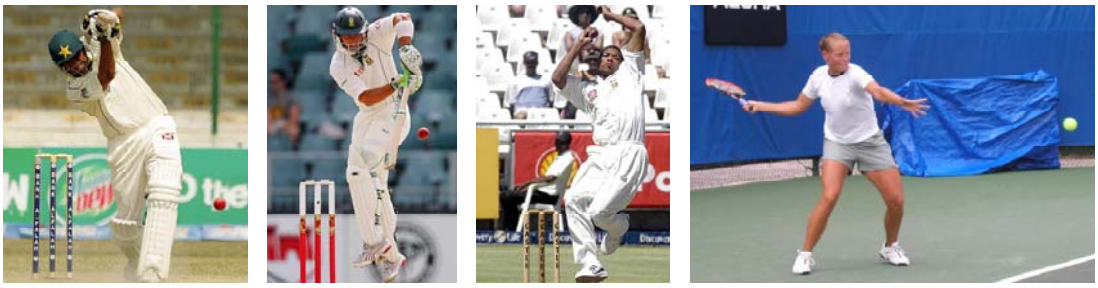
techniques have been proposed to avoid exhaustively searching the image [119, 67], which makes the algorithm more efficient. While the most popular detectors are still based on sliding windows, more recent work has tried to integrate context to obtain better performance [96, 52, 24]. However, in most of the works the performance is improved by a relatively small margin.

It is out of the scope of this chapter to develop an object detection or pose estimation method that generally applies to all situations. Instead, we focus on the role of context in these problems. Our work is inspired by a number of previous works that have used context in vision tasks [84, 56, 108, 96, 52, 24, 81]. In most of these works, one type of scene information serves as contextual facilitation to a main recognition problem. For example, ground planes and horizons can help to refine pedestrian detections. In this chapter, we try to bridge the gap between two seemingly unrelated problems - object detection and human pose estimation, in which the *mutual contexts* play key roles for understanding their interactions. The problem of classifying HOI activities has been studied in [49] and [126], but no detailed understanding of the human pose (e.g. parsing the body parts) is offered in these works. To our knowledge, our work is the first one that explicitly models the mutual contexts of human poses and objects and allows them to facilitate the recognition of each other.

5.3 Modeling mutual context of object and pose

Given an HOI activity, our goal is to estimate the human pose and to detect the object that the human interacts with. Figure 5.2 illustrates that both tasks are challenging. The relevant objects are often small, partially occluded, or tilted to an unusual angle by the human. The human poses, on the other hand, are usually highly articulated and many body parts are self-occluded. Furthermore, even in the same activity, the configurations of body parts might differ in different images due to different shooting angles or human poses.

Here we propose a novel model to exploit the mutual context of human poses and objects in one coherent framework, where object detection and human pose estimation can benefit from each other. For simplicity, we assume that only one object is involved



(a) The relevant objects that interact with the human may be very small, partially occluded, or tilted to an unusual angle.

(b) Human poses of the same activity might be inconsistent in different images due to different camera angles (the left two images), or the way that the human interacts with the object (the right two images).

Figure 5.2: Challenges of both object detection and human pose estimation in HOI activities.

in each activity.

5.3.1 The model

A graphical illustration of our model is shown in Figure 5.3(a). Our model can be thought of as a hierarchical random field, where the overall activity classes, objects, and human poses all contribute to the recognition and detection of each other. We use categorical variables A, O, and H to denote the class labels of activities, objects, and human poses, respectively. The human pose is further decomposed into some body parts, denoted by $\{P_n\}_{n=1}^{N}$. For each body part P_n and the object O, f_{P_n} and f_O denote the vectors of visual features that describe the corresponding image regions respectively. Note that because of the difference between the human poses in each HOI activity (Figure 5.2(b)), we allow each activity class (A) to have more than one types of human pose (H), which are latent (unobserved) variables to be learned in

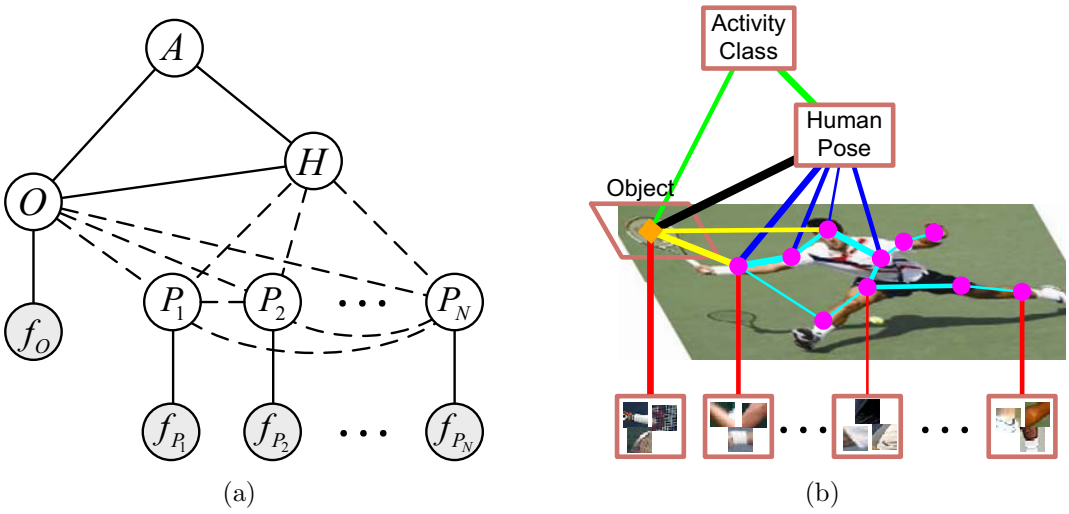


Figure 5.3: **(a)** A graphical illustration of the mutual context model. The edges represented by dashed lines indicate that their connectivity will be obtained by structure learning. A denotes an HOI activity class, H the human pose class, P a body part, and O the object. f_O and f_P's are image appearance information of O and P respectively. **(b)** Illustration of our model on an image of a human playing tennis. Different types of potentials are denoted by lines with different colors. Line widths represent the importance of the potentials for the human-object interaction of playing tennis.

training.

Our model encodes the mutual connections between the object, the human pose and the body parts. Intuitively speaking, this allows the model to capture important connections between, say, the tennis racket and the right arm that is serving the tennis ball (Figure 5.3(b)). We observe, however, that the left leg in tennis serving is often less relevant to the detection of the ball. The model should therefore have the flexibility in deciding what parts of the body should be connected to the object O and the overall pose H. Dashed lines in Figure 5.3(a) indicate that these connections will be decided through structure learning. Depending on A, O and H, these connections might differ in different situations. Putting everything together, the overall model can be computed as $\Psi = \sum_e w_e \psi_e$, where e is an edge of the model, ψ_e and w_e are its potential function and weight respectively. We use the same w_e for each ψ_e even when it is a vector. We now enumerate the potentials of this model:

- $\psi_e(A, O)$, $\psi_e(A, H)$, and $\psi_e(O, H)$ model the agreement between the class labels of A, O, and H, each estimated by counting the co-occurrence frequencies of the pair of variables on training images.

- $\psi_e(O, P_n)$ models the spatial relationship between the object O and the body part P_n, which is computed by

$$\text{bin}(\mathbf{l}_O - \mathbf{l}_{P_n}) \cdot \text{bin}(\theta_O - \theta_{P_n}) \cdot \mathcal{N}(s_O/s_{P_n}) \tag{5.1}$$

  where $(\mathbf{l}, \theta, s)$ is the position, orientation, and scale of an image part. $\text{bin}(\cdot)$ is a binning function as in [98] and $\mathcal{N}(\cdot)$ is a Gaussian distribution.

- $\psi_e(P_m, P_n)$ models the spatial relationship between different body parts, computed similarly to Eq.5.1.

- $\psi_e(H, P_n)$ models the compatibility between the pose class H and a body part P_n. It is computed by considering the spatial layout of P_n given a reference point in the image, in this case the center of the human face (P_1).

$$\psi_e(H, P_n) = \text{bin}(\mathbf{l}_{P_n} - \mathbf{l}_{P_1}) \cdot \text{bin}(\theta_{P_n}) \cdot \mathcal{N}(s_{P_n}) \tag{5.2}$$

- $\psi_e(O, f_O)$ and $\psi_e(P_n, f_{P_n})$ model the dependence of the object and a body part with their corresponding image evidence. We use the shape context [3] feature for image representation, and train a detector [119] for each body part and each object in each activity. Detection outputs are normalized as in [2].

In our algorithm, all the above potential functions are dependent on O and H except those between A, O, and H (the first bullet). We omit writing this point every time for space consideration. For example, for different human pose H, $\psi_e(O, P_n)$ is estimated with different parameters, which represents a specific spatial configuration between P_n and the object O, conditioned on the particular human pose H.

Figure 5.4: Visualization of the learned HOI models. Each row shows two models and their corresponding image examples for one activity. The illustrative figure for each model represents the average spatial layout of the object and body parts of all the images that are assigned to the model. The different color codes are: object = double red box, head and torso = magenta, arms = green, legs = cyan.)

5.3.2 Properties of the model

Central to our model formulation is the hypothesis that both human pose estimation and object detection can benefit from each other in HOI activities. Without knowing the location of the arm, it is difficult to spot the location of the tennis racket in tennis serving. Without seeing the croquet mallet, the heavily occluded arms and legs can become too obscured for robust pose estimation. We highlight here some important properties of our model.

Co-occurrence context for the activity class, object, and human pose. Given the presence of a tennis racket, the human pose is more likely to be playing tennis instead of playing croquet. That is to say, co-occurrence information can be

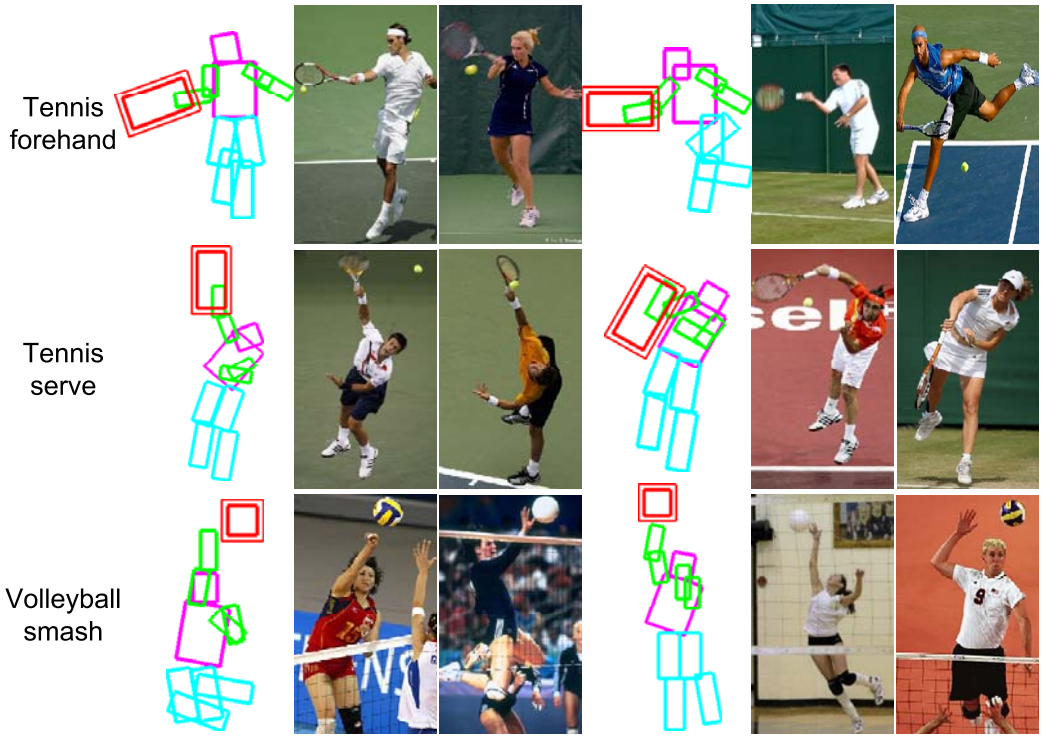


Figure 5.5: Continuation of Figure 5.4.

beneficial for coherently modeling the object, the human pose, and the activity class.

Multiple types of human poses for each activity. Our model allows each activity (A) to consist of more than one human pose (H). Treating H as a hidden variable, our model automatically discovers the possible poses from training images. This gives us more flexibility to deal with the situations where the human poses in the same activity are inconsistent, as shown in Figure 5.2(b). We show in Figure 5.4 and Figure 5.5 the pose variability for each HOI activity.

Spatial context between object and body parts. Different poses imply that the object is handled by the human in different manners, which are modeled by $\{\psi_e(O, P_n)\}_{n=1}^{N}$. Furthermore, not all these relationships are critical for understanding an HOI activity. Therefore for each combination of O and H, our algorithm automatically discovers the connectivity between O and each P_n, as well as the connectivity

among H and $\{P_n\}_{n=1}^{N}$.

Relations with the other models. Our model has drawn inspirations from a number of previous works, such as modeling spatial layout of different image parts [38, 98, 2], using agreement of different image components [96], using multiple models to describe the same concept (human pose in our problem) [75], non-tree models for better pose estimation [121, 13], and discriminative training [2]. Our model integrates all the properties in one coherent framework to perform two seemingly different tasks, human pose estimation and object detection, to the benefit of each other.

5.4 Model learning

Given the training images of HOI activities with labeled objects and body parts, the learning step needs to achieve two goals: *structure learning* to discover the hidden human poses and the connectivity among the object, human pose, and body parts; and *parameter estimation* for the potential weights to maximize the discrimination between different activities. The output of our learning method is a set of models, each representing one connectivity pattern and potential weights for one type of human pose in one activity class. Algorithm 3 is a sketch of the overall framework. We discover new human poses by clustering the samples in the model that has the weakest discriminative ability in each iteration, which results to some sub-classes. Structure learning is applied to each sub-class respectively. The learning process terminates when the number of mis-classified samples in each sub-class is small (less than three in this chapter).

5.4.1 Hill-climbing structure learning

Our algorithm performs structure learning for each sub-class, i.e. each pose in each activity, respectively. Given a set of images where humans interact with the same class of object (O) with the same type of pose (H), our objective is to learn a connectivity pattern ($C = \{C_{OP}, C_{HP}, C_{PP}\}$) which best models the interaction between the human and the object. As shown in Figure 5.3(a), C_{OP} describes the connection

foreach *activity class* **do**
  - Hill climbing structure learning.;
end
foreach *iteration* **do**
  - Model parameter estimation by max-margin learning;
  - Choose the model with the largest number of mis-classified images;
  - Cluster the images in the selected model into two sub-classes;
  - Structure learning for the two new sub-classes;
end

Algorithm 3: Learning framework of the mutual context model. Each sub-class corresponds to a type of human pose in an HOI activity. Initially there are one sub-class for each activity.

between the object and different body parts, C_{HP} the connection between the human pose and body parts, and CPP the connection among different body parts. Note that we learn a connectivity for each pair of human pose and object respectively.

In the learning step, given the locations and size of the object and human body parts, our objective function is

$$\arg\max_{\mathcal{C}} \sum_{i} \left\{ \sum_{\mathcal{C}_{OP}} \psi_e^i(O, P_n) + \sum_{\mathcal{C}_{HP}} \psi_e^i(H, P_n) + \sum_{\mathcal{C}_{PP}} \psi_e^i(P_m, P_n) + \log \mathcal{N}(|\mathcal{C}|) \right\} \quad (5.3)$$

where $|\mathcal{C}|$ is the number of edges in $\mathcal{C}$, and $\log(|\mathcal{C}|)$ is a Gaussian prior over the number of edges. The other structure learning $\psi_e^i(\cdot)$ is the potential value computed from the i-th image. Note that in the structure learning stage, we omit the weights of different potential terms. The potential weights will be estimated in the parameter estimation step (Section 5.4.2).

For each sample i, the value of all the potential terms $\psi_e^i(O, P_n)$, $\psi_e^i(H, P_n)$, and $\psi_e^i(P_m, P_n)$ can be computed by using the Maximum-Likelihood approach. Therefore, given the values of all the potential terms, we use a hill-climbing search [66] method to optimize Eq.5.3. Our hill climbing method can be used to solve other structural learning constraints, such as a L1 penalty on the number of edges.

In the hill-climbing method, we first randomly initialize the connectivity. Then we execute the following steps repeatedly: We consider all of the solutions that are

neighbors of the current one by adding or removing an edge. We compute the score of each solution using Eq.5.3, from which the one that leads to the best improvement in the score is selected. We continue this process until no improvement can be achieved. Because all the potential terms in Eq.5.3 can be pre-computed, hill-climbing method converges fast in our problem. The method, however, can only reach a local maximum. There is no guarantee that the local maximum is actually the global optimum. To improve the search result, we adopt the following two approaches.

The first approach is to keep a *tabu list* of operators (adding or deleting a specific edge) that we have recently applied. Then in each search step, we do not consider the operators that reverse the effect of operators applied in the last five steps. Thus, if we add an edge between two nodes, we cannot delete this edge in the next five steps. The tabu list forces the search procedure to explore new directions in the search space so that the performance can be improved [66].

The other approach to reduce the impact of local optimum is *randomization*. We can initialize the connectivity at different starting points, and then use a hill-climbing algorithm for each one, from which the best result is selected. In our method, we use one manually designed starting point and two other random ones. In the manually designed starting point, we connect the human pose node with all the body parts, connect the object with the right-lower-arm, and use a kinematic structure among different body parts.

5.4.2 Max-margin parameter estimation

Given the model outputs by the structure learning step, the parameter estimation step aims to obtain a set of potential weights that maximize the discrimination between different classes of activities (A in Figure 5.3(a)). But unlike the traditional random field parameter estimation setting [112], in our model each class can contain more than one pose (H), which can be thought of as multiple sub-classes. Our learning algorithm needs to, therefore, estimate parameters for each pose (i.e. sub-class, where the sub-class labels are obtained from Section 5.4.1) while optimizing for maximum discrimination among the global activity classes.

We propose a novel max-margin learning approach to tackle this problem. Let $(\mathbf{x}_i, c_i, y(c_i))$ be a training sample, where $\mathbf{x}_i$ is a data point, c_i is the sub-class label of $\mathbf{x}_i$, and $y(c_i)$ maps c_i to a class label. We want to find a function $\mathcal{F}$ that assigns an instance $\mathbf{x}_i$ to a sub-class. We say that $\mathbf{x}_i$ is correctly classified if and only if $y(\mathcal{F}(\mathbf{x}_i)) = y(c_i)$. Our classifier is then formulated as $\mathcal{F}(\mathbf{x}_i) = \arg\max_r\{\mathbf{w}_r \cdot \mathbf{x}_i\}$, where $\mathbf{w}_r$ is a weight vector for the r-th sub-class. Inspired by the traditional max-margin learning problems [15], we introduce a slack variable ξ_i for each sample $\mathbf{x}_i$, and optimize the following objective function:

$$\min_{\mathbf{w},\xi} \frac{1}{2}\sum_r \|\mathbf{w}_r\|_2^2 + \beta\sum_i \xi_i \tag{5.4}$$

$$\text{subject to:} \quad \forall i,\ \xi_i \geq 0$$

$$\forall i, r \text{ where } y(r) \neq y(c_i),\ \mathbf{w}_{c_i} \cdot \mathbf{x}_i - \mathbf{w}_r \cdot \mathbf{x}_i \geq 1 - \xi_i$$

where $\|\mathbf{w}_r\|_2$ is the L2 norm of $\mathbf{w}_r$, β is a normalization constant. Again, note that the weights are defined with respect to sub-classes while the classification results are measured with respect to classes. We optimize Eq.5.4 by using the multiplier method [54]. Mapping the above symbols to our model, $\mathbf{x}_i$ are the potential function values computed on an image. Potential values for the disconnected edges are set to 0. In order to obtain better discrimination among different classes, we compute the potential values of an image on the models of all the sub-classes, and concatenate these values to form the feature vector. Sub-class variable c_i indicates human pose H, and $y(c_i)$ is the class label A. Please refer to the full version of [127] for more detail about the method.

5.4.3 Analysis of our learning algorithm

Figure 5.4 illustrates the two models (correspond to two types of human poses) learned by our algorithm for each HOI class. We can see the big difference of human poses in some activities (e.g. croquet-shot and tennis-serve), and such wide intra-class variability can be effectively captured by our algorithm. In these cases, using only one human pose for each HOI class is not enough to characterize well all the images in

this class. Furthermore, we observe that by using structure learning, our model can learn meaningful connectivity between the object and the body parts, e.g. croquet mallet and legs, right forehand and tennis racket.

5.5 Model inference

Given a new testing image $\mathcal{I}$, our objective is to estimate the pose of the human in the image, and to detect the object that is interacting with the human. An illustration of the inference procedure is shown in Figure 5.6. In order to detect the tennis racket in this image, we maximize the likelihood of this image given the models that are learned for tennis-forehand. This is achieved by finding a best configuration of human body parts and the object (tennis racket) in the image, which is denoted as $\max_{O,H} \Psi(A_k, O, H, \mathcal{I})$ in Figure 5.6. In order to estimate the human pose, we compute $\max_{O,H} \Psi(A_k, O, H, \mathcal{I})$ for each activity class and find the class A^* that corresponds to the maximum likelihood score. This score can be used to measure the confidence of activity classification as well as human pose estimation.

For each model, the above inference procedure involves a step to find the best spatial configuration of the object and different body parts for an image. We solve this problem by using the compositional inference method [13]. The algorithm has a bottom-up stage which makes proposals of different parts. The bottom-up stage starts from the object detection and human body parts detection scores, from which we obtain the image parts with large detection scores for further processing. Then in the first level of the bottom-up stage, if two nodes of the body parts or the object are connected, we enumerate all combinations of the strong detection responses of the two nodes. The combinations with low fitness scores are removed. We compute the fitness score by adding the detection scores of the two parts, as well as the potential value of the edge between them. A clustering method is applied to the remaining combinations to obtain a small set of max-proposals. Then the remaining proposals are merged according to the connectivity structure among different image parts. In the compositional inference stage, we omit the weights of different potentials and set all of them to 1. Please refer to [13] for more details about this inference method. In

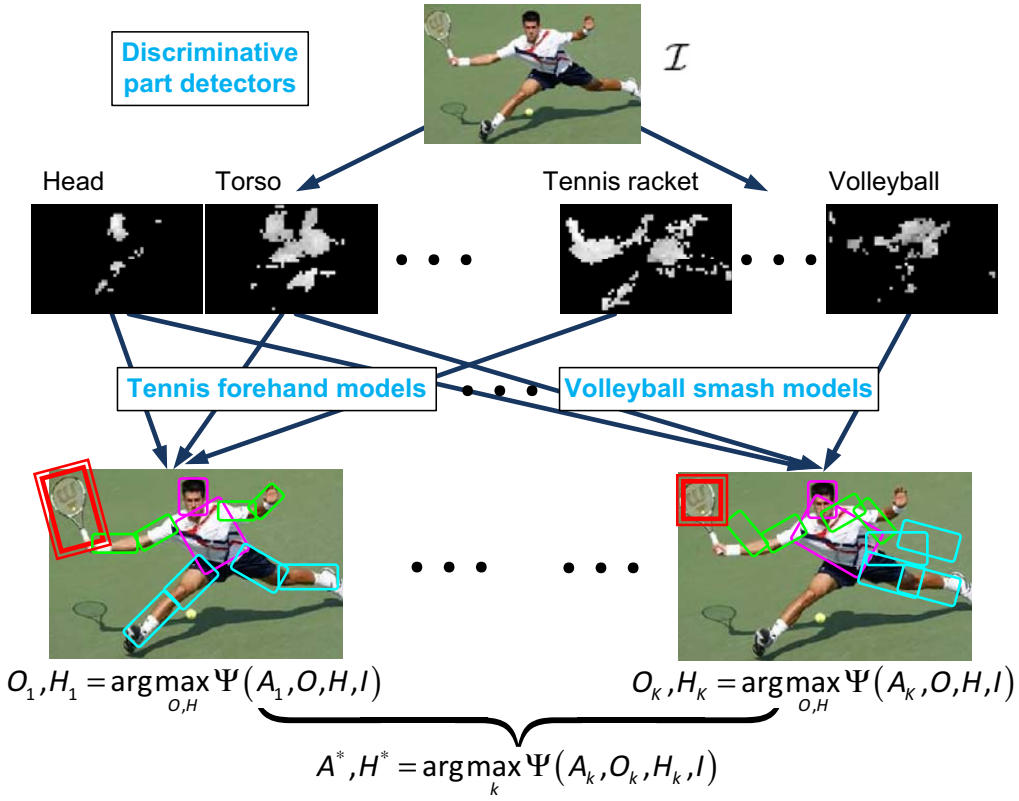


Figure 5.6: The framework of our inference method for the mutual context model. Given an input image $\mathcal{I}$, the inference results are: (1) object detection results O_k (e.g. O_1 is the tennis racket detection result); (2) human pose estimation result H^*; (3) activity classification result A^*.

the first level of the bottom-up stage, we propose 5000 node combinations for each edge. After clustering only $30 \sim 100$ combinations are remained.

5.6 Experiments

5.6.1 The sports dataset

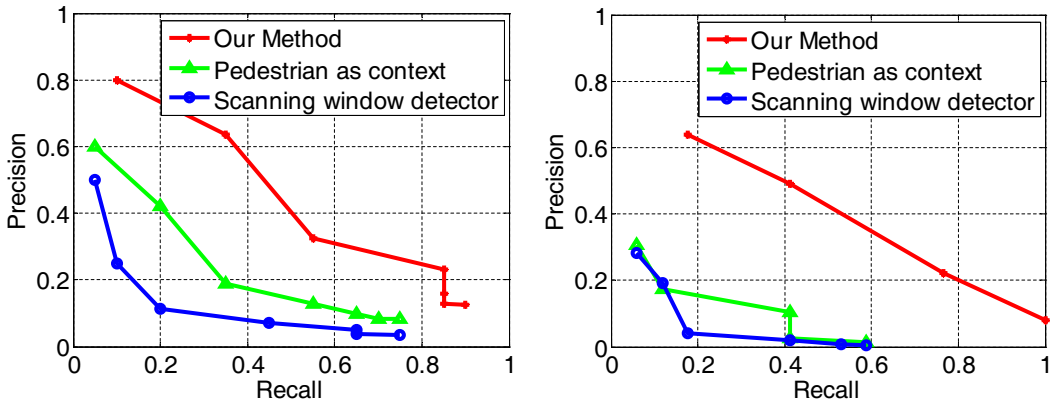
We evaluate our approach on a known HOI dataset of six activity classes [49]: cricket-defensive shot (player and cricket bat), cricket-bowling (player and cricket ball), croquet-shot (player and croquet mallet), tennis-forehand (player and tennis racket),

tennis-serve (player and tennis racket), and volleyball-smash (player and volleyball). There are 50 images in each activity class. We use the same setting as that in [49]: 30 images for training and 20 for testing. In [49] only activity classification results were reported. In this work we also evaluate our method on the tasks of object detection and human pose estimation.

5.6.2 Better object detection by pose context

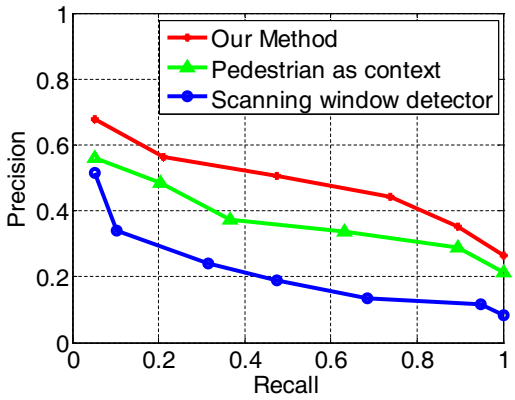
In this experiment, our goal is to detect the presence and location of the object given an HOI activity. To evaluate the effectiveness of our model, we compare our results with two control experiments: a scanning window detector as a baseline measure of object detection without any context, and a second experiment in which the approximate location of the person is provided by a pedestrian detector [16], hence providing a co-occurrence context and a very weak location context. Results of these three experiments, measured by precision-recall curves, are shown in Figure 5.7. The curves of our algorithm are obtained by considering the scores $\Psi(A, O, H, \mathcal{I})$ of all the results that are proposed by the compositional inference method. To ensure fair comparison, all experiments use the same input features and object detectors described in Section 5.3, and non-max suppression is applied equally to all methods.

The results in Figure 5.7 show that our detection method achieves the best performance. By using human pose as context, more detailed spatial relationship between different image parts can be discovered, which greatly helps to detect objects that are traditionally very difficult. For example, in the case of the cricket ball (Figure 5.7(b)), a sliding window method yields an average precision of 17%, whereas our model with pose-context measure is 46%. In almost all the cases of the five objects, the average precision score of our method is more than three times as the sliding window method. Figure 5.8 shows an example of using the three methods for object detection. Figure 5.9 shows more object detection results on a variety of testing images.

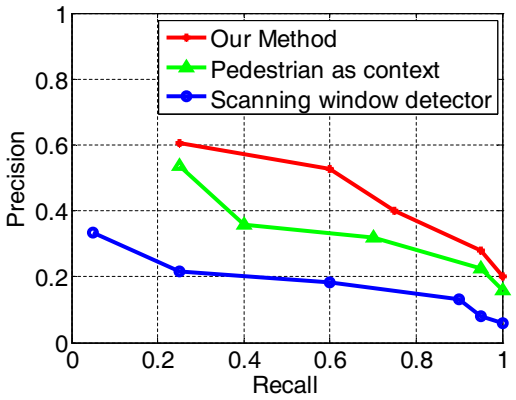


(a) cricket bat

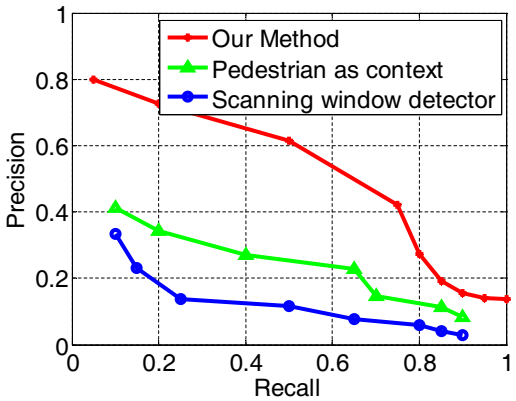
(b) cricket ball



(c) croquet mallet



(d) tennis racket



(e) volleyball

Figure 5.7: Object detection results measured by precision-recall curves. We compare our algorithm to a scanning window detector and a detector that uses pedestrian detection as the human context for object detection.

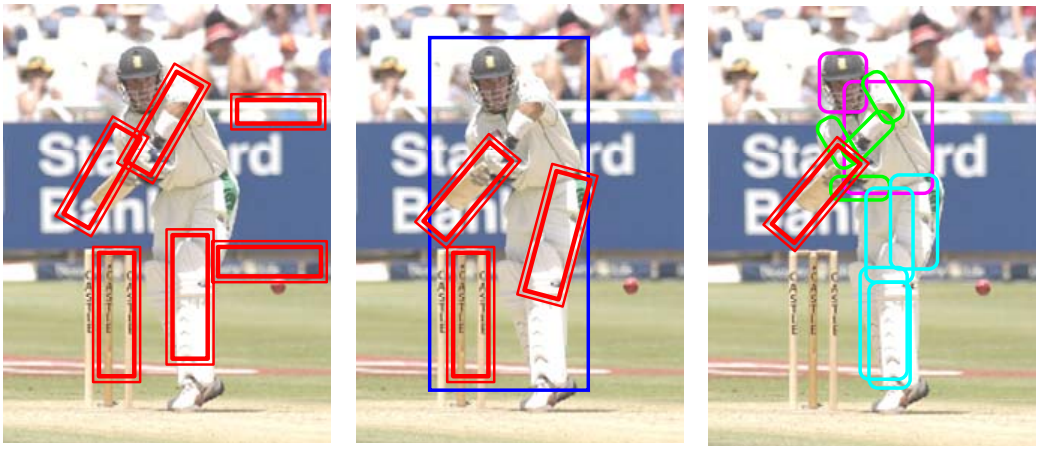


Figure 5.8: Object (cricket bat) detection results (red double-line bounding boxes) obtained by: a sliding window detector (**left**), the same detector using pedestrian detection as context (**middle**), and our method (**right**). Pedestrian detection is shown in a blue bounding box. The human pose estimation results are shown in colored rectangles in the right image.

Method	Iterative parsing [98]	Pictorial structure [38]	Class based PS	Our model one pose	Our full model
Torso	52 ± 19	50 ± 14	59 ± 9	63 ± 5	$\mathbf{66 \pm 6}$
Upper leg	22 ± 14 22 ± 10	31 ± 12 30 ± 9	36 ± 11 26 ± 17	40 ± 8 36 ± 15	$\mathbf{43 \pm 8}$ $\mathbf{39 \pm 14}$
Lower leg	21 ± 9 28 ± 16	31 ± 15 27 ± 18	39 ± 9 27 ± 9	41 ± 10 31 ± 9	$\mathbf{44 \pm 10}$ $\mathbf{34 \pm 10}$
Upper arm	24 ± 16 28 ± 17	18 ± 6 19 ± 9	30 ± 12 31 ± 12	38 ± 13 35 ± 10	$\mathbf{44 \pm 9}$ $\mathbf{40 \pm 13}$
Fore arm	17 ± 11 14 ± 10	11 ± 8 11 ± 7	13 ± 6 18 ± 14	21 ± 12 23 ± 14	$\mathbf{27 \pm 16}$ $\mathbf{29 \pm 13}$
Head	42 ± 18	45 ± 8	46 ± 11	52 ± 8	$\mathbf{58 \pm 11}$

Table 5.1: Pose estimation results by our full model and four comparison methods for all testing images. The average part detection percent correctness and standard deviation over 6 HOI classes are presented for each body part. If two numbers are reported in one cell, the left one indicates the left body part and right one indicates the right body part. The best result for each body part is marked in bold font.

5.6.3 Better pose estimation by object context

Similarly to object detection, we show in this experiment that human pose estimation is significantly improved by object context. Here we compare our full model with four

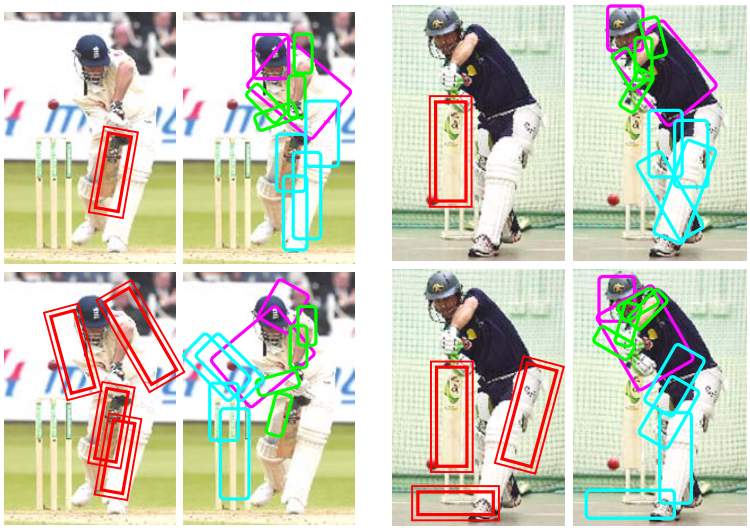
different control experiments.

- An *iterative parsing* method by Ramanan et al [98];
- A state-of-the-art *pictorial structure* model [2];
- We re-train the model in [2] with a *pictorial structure model per class* for better modeling of each class;
- Our proposed model by imposing only *one sub-class (human pose, H) per HOI activity*, examining the importance of allowing a flexible number of pose models to account for the intra-activity variability.

All of the models are trained using the same training data described in Section 5.6.1. Following the convention proposed in [40], a body part is considered correctly localized if the endpoints of its segment lie within 50% of the ground-truth segment length from their true positions. Experimental results are shown in Table 5.1. The percentage correctness tells us that pose estimation still remains a difficult problem. No method offers a solution near 100%. Our full model significantly outperforms the other approaches, even showing a 10% average improvement over a class-based, discriminatively trained pictorial structure model. Furthermore, we can see that allowing multiple poses for each activity class proves to be useful for improving pose estimation accuracy. More sample results are shown in Figure 5.9 and Figure 5.10, where we visualize the pose estimation results by comparing our model with the state-of-the-art pictorial structure model by [2]. We show that given the object context, poses estimated by our model are less prone to errors that result in strange looking body gestures (e.g. Figure 5.9(d)), or a completely wrong location (e.g. Figure 5.10(c)).

5.6.4 Combining object and pose for action classification

As shown in Figure 5.6, by inferring the human pose and object in the image, our model gives a prediction of the class label of the human-object interaction. We compare



(a) cricket defensive shot (b) cricket defensive shot

(c) tennis forehand (d) tennis serve (e) croquet shot

Figure 5.9: Example testing results of object detection and pose estimation. Each sub-figure contains one testing image, tested on the following four conditions: upper-left→object detection by our model, lower-left→object detection by a scanning window, upper-right→pose estimation by our model, and lower-right→pose estimation by the state-of-the-art pictorial structure method in [2]. Detected objects are shown in double-line red bounding boxes. The color codes for different body parts are: head and torso - magenta, arms - green, legs - cyan.

(a) cricket bowling (b) volleyball smash (c) volleyball smash

Figure 5.10: Continuation of Figure 5.9.

our method with the results reported in [49], and use a bag-of-words representation with a linear SVM classifier as the baseline. The results are shown in Figure 5.11.

Figure 5.11 shows that our model significantly outperforms the bag-of-words method and performs slightly better than [49]. Note that the method in [49] uses predominantly the background scene context (e.g. appearance differences in sport courts), which turns out to be highly discriminative among most of these classes of activities. Our method, on the other hand, focuses on the core problem of human-object interactions. It is therefore less data set dependent.

5.7 Summary

In this chapter, we treat object and human pose as the context of each other in different HOI activity classes. We develop a random field model that uses a structure learning method to learn important connectivity patterns between objects and human body parts. Experiments show that our model significantly outperforms other state-of-the-art methods in both problems. Our model can be further improved in a number

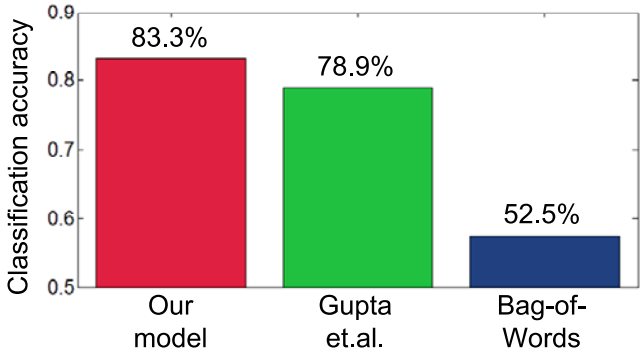


Figure 5.11: Activity recognition accuracy of different methods: our model, Gupta et al [49], and bag-of-words.

of directions. For example, inspired by [84, 49], we can incorporate useful background scene context to facilitate the recognition of foreground objects and activities. In the next chapter, we will show how to deal with more than one object in each action.

Chapter 6

Mutual Context Model II: Multiple Objects

The mutual context model in the previous chapter assumes one human and one object interaction in each action, and can be used for action recognition, object detection, and human pose estimation. In this chapter,[1] we extend the model so that it can deal with the cases where a human interacting with any number of objects. We also show the application of mutual context model in some higher level, such as action retrieval.

6.1 Introduction

Many recent works on human action recognition use contextual information [49, 127] to help improve the recognition performance. Compared to the methods that directly associate low-level image descriptors with class labels [126, 18], context (e.g. estimating human pose, detecting objects) provides deeper understanding of human actions.

Following the method of Yao & Fei-Fei [127], in this chapter we consider human actions as interactions between humans and objects, and jointly model the relationship between them using the *mutual context model*. As shown in Figure 6.1, our

[1] An early version of this chapter has been presented in [131].

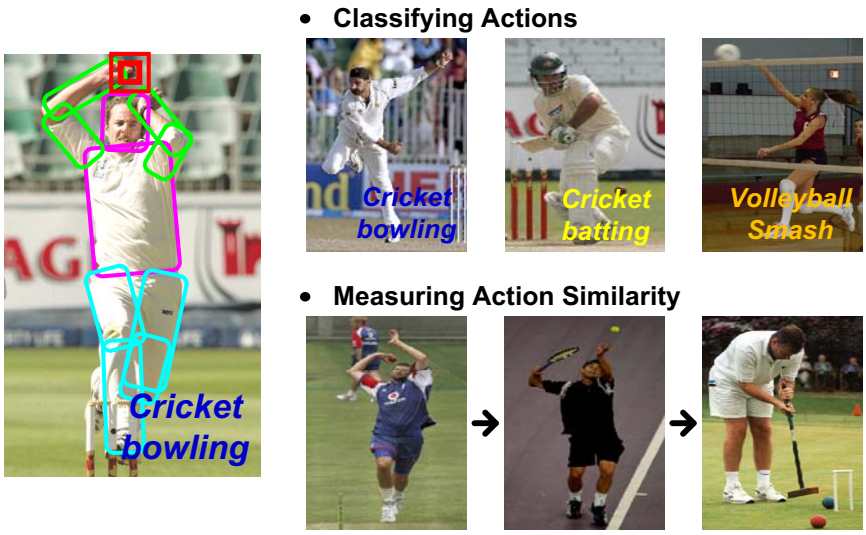
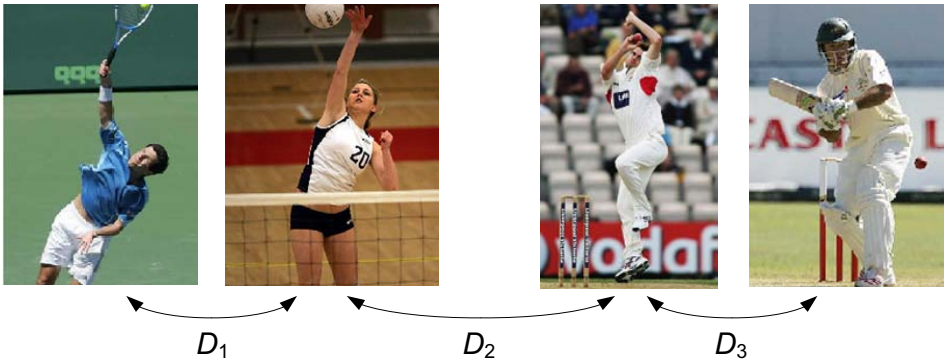


Figure 6.1: Objects and human poses can facilitate the recognition of each other in the actions of human-object interactions, as shown in the cricket bowling image. Based on the recognition of objects and human poses, we consider two tasks: action classification and measuring action similarity. "→" indicates that the left image is more similar to the left-most cricket bowling image than the right one.

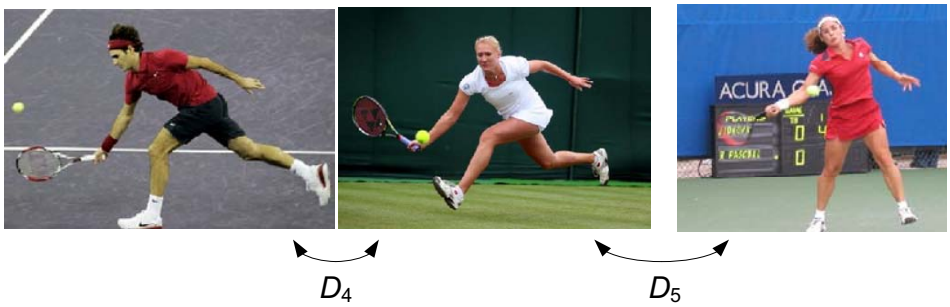
method allows objects and human poses to serve as mutual context to facilitate the recognition of each other, based on which we address two action recognition tasks:

- Conventional *action classification* where we assign a class label to each action image.
- *Measuring the similarity* between different action images. The goal is to make the similarity measure consistent with human perception.

The second task, measuring action similarity, is very different from conventional action classification problems. As shown in Figure 6.2, human actions lie in a relatively continuous space and different actions can be correlated. We humans are able to distinguish small changes in human poses as well as capture the relationship of different actions from the objects or scene background. However it is difficult to capture all these subtleties by simply assigning action images into several independent classes as in the conventional action classification problem. In this work, by explicitly



(a) A human action can be more related to some actions than others. $D_1 < D_2$ because the left-most two images have similar human poses. $D_3 < D_2$ because the right-most two images are from the same sport and the objects "cricket ball" and "cricket stump" are present in both images.



(b) Human actions lie in a continuous space. Humans are able to capture the difference between different images even if they belong to the same action class. $D_4 < D_5$ because the left two images have very similar human poses.

Figure 6.2: Examples of the distance between different images of human actions denoted by D_i.

considering objects and human poses, we obtain a distance[2] measure of action images which is largely consistent with human annotation.

In the rest of this chapter, we first introduce related work in Section 6.2, and then elaborate on the mutual context model and distance measure method in Section 6.3. Finally, experimental results are presented in Section 6.4.

[2]Small distance indicates large image similarity.

6.2 Related work

Our method builds upon the mutual context model [127] that explores the relationships between objects and human poses in human actions. The model presented in this chapter is more flexible and discriminative in that: (1) it learns an overall relationship between different actions, objects, and human poses, rather than modeling each action class separately; (2) it can deal with any number of objects, instead of being limited to the interactions between one human and one object; (3) it incorporates a discriminative action classification component which takes global image information into consideration.

While different objects and annotations of action classes can be represented by discrete indexes, human poses lie in a space where the location of body parts changes continuously. To make the joint modeling of actions, objects, and human poses easier, we discretise possible layouts of human body parts into a set of representative poses, termed as *atomic poses* (as shown in Figure 6.3). Our atomic poses are discovered in a similar manner as poselets [8]. While poselets are local detectors for specific body parts, the atomic poses consider the whole human body and can be thought of as a dictionary of human poses.

6.3 Algorithm

In this section, we describe the mutual context model that jointly models a set of actions $\mathcal{A}$, objects $\mathcal{O}$, and atomic poses $\mathcal{H}$. We first introduce the model (Section 6.3.1), then describe how to obtain the atomic poses (Section 6.3.2) and the model learning approach (Section 6.3.3). Finally we show our approach to classify action images (Section 6.3.4) and measure action distance (Section 6.3.5).

6.3.1 Mutual context model representation

Given an image I with annotations of action class $A \in \mathcal{A}$, bounding boxes of objects $O \in \mathcal{O}$ and body parts in the human pose $H \in \mathcal{H}$, our model learns the strength of the interactions between them. We further make the interaction conditioned on image

evidence, so that the components that are harder to recognize play less important roles in the interaction. Our model is represented as

$$\Psi(A, O, H, I) = \phi_1(A, O, H) + \phi_2(A, I) + \phi_3(O, I) + \phi_4(H, I) + \phi_5(O, H) \quad (6.1)$$

where ϕ_1 models the compatibility between A, O, and H; ϕ_{2-4} models the image evidence using state-of-the-art action classification, object detection, and pose estimation approaches; ϕ_5 considers the spatial relationship between objects and body parts. We now elaborate on each term in Eqn.6.1.

Compatibility between actions, objects, and human poses. $\phi_1(A, O, H)$ is parameterized as

$$\phi_1(A, O, H) = \sum_{i=1}^{N_h} \sum_{m=1}^{M} \sum_{j=1}^{N_o} \sum_{k=1}^{N_a} \mathbf{1}_{(H=h_i)} \cdot \mathbf{1}_{(O^m=o_j)} \cdot \mathbf{1}_{(A=a_k)} \cdot \zeta_{i,j,k} \quad (6.2)$$

where N_h is the the total number of atomic poses (see Section 6.3.2) and h_i is the i-th atomic pose in $\mathcal{H}$ (similarly for N_o, o_j, N_a, and a_k). $\zeta_{i,j,k}$ represents the strength of the interaction between h_i, o_j, and a_k. M is the number of object bounding boxes within the image, and O^m is the object class label of the m-th box.

Modeling Actions. $\phi_2(A, I)$ is parameterized by training an action classifier based on the extended image regions of the humans. We have

$$\phi_2(A, I) = \sum_{k=1}^{N_a} \mathbf{1}_{(A=a_k)} \cdot \eta_k^T \cdot s(I) \quad (6.3)$$

where $s(I)$ is an N_a-dimensional output of a one-vs-all discriminative classifier. η_k is the feature weight corresponding to a_k.

Modeling objects. Inspired by [21], we model objects in the image using object detection scores in each detection bounding box and the spatial relationships between these boxes. Denoting the detection scores of all the objects for the m-th box as

$g(O^m)$, $\phi_3(O, I)$ is parameterized as

$$\phi_3(O, I) = \sum_{m=1}^{M}\sum_{j=1}^{N_o} \mathbf{1}_{(O^m=o_j)} \cdot \gamma_j^T \cdot g(O^m) + \sum_{m=1}^{M}\sum_{m'=1}^{M}\sum_{j=1}^{N_o}\sum_{j'=1}^{N_o} \mathbf{1}_{(O^m=o_j)} \cdot \mathbf{1}_{(O^{m'}=o_{j'})} \cdot \gamma_{j,j'}^T \cdot b(O^m, O^{m'}) \tag{6.4}$$

where γ_j is the weights for object o_j. $\gamma_{j,j'}$ encodes the weights for geometric configurations between o_j and $o_{j'}$. $b(O^m, O^{m'})$ is a bin function with a grid representation as in [21] that models the relationship between the m-th and m'-th bounding boxes.

Modeling human poses. $\phi_4(H, I)$ models the atomic pose that H belongs to and the likelihood of observing image I given that atomic pose. We have

$$\phi_4(H, I) = \sum_{i=1}^{N_h}\sum_{l=1}^{L} \mathbf{1}_{(H=h_i)} \cdot \left(\alpha_{i,l}^T \cdot p(\mathbf{x}_I^l|\mathbf{x}_{h_i}^l) + \beta_{i,l}^T \cdot f^l(I)\right) \tag{6.5}$$

where $\alpha_{i,l}$ and $\beta_{i,l}$ are the weights for the location and appearance of the l-th body part in atomic pose h_i. $p(\mathbf{x}_I^l|\mathbf{x}_{h_i}^l)$ is the Gaussian likelihood of observing $\mathbf{x}_I^l$, the joint of the l-th body part in image I, given the standard joint location of the l-th body part in atomic pose h_i. $f^l(I)$ is the output of a detector for the l-th body part in this image.

Spatial relationship between objects and body parts. We achieve a better modeling of objects and human body parts by considering their spatial relationships. $\phi_5(H, O)$ is parameterized as

$$\phi_5(H, O) = \sum_{m=1}^{M}\sum_{i=1}^{N_h}\sum_{j=1}^{N_o}\sum_{l=1}^{L} \mathbf{1}_{(H=h_i)} \cdot \mathbf{1}_{(O^m=o_j)} \cdot \lambda_{i,j,l}^T \cdot b(\mathbf{x}_I^l, O^m) \tag{6.6}$$

where $b(\mathbf{x}_I^l, O^m)$ denotes the spatial relationship between the l-th body part in I and the m-th object bounding box. We again use the bin function as in [21]. $\lambda_{i,j,l}$ encodes the weights for this relationship when the object class of O^m is o_j.

6.3.2 Obtaining atomic poses

In this section, we discuss a clustering method to obtain atomic poses. Given the training images, we first align the annotations of each image so that the torsos of all the humans have the same position and size, and normalize the range of variations of both position and orientation to $[-1, 1]$. If there is a missing body part due to occlusion, we fill in the annotation with the average annotation values for that particular part. We then use hierarchical clustering with the max linkage measure to obtain a set of clusters, where each cluster represents an atomic pose. Given two images i and j, their distance is measured by $\sum_{l=1}^{L} \mathbf{w}^T \cdot |\mathbf{x}_i^l - \mathbf{x}_j^l|$, where $\mathbf{x}_i^l$ denotes the position and orientation of the l-th body part in image i, $\mathbf{w}$ is a weight vector (0.15 and 0.1 for location and orientation components respectively), L is the number of body parts.

The atomic poses can be thought of as a dictionary of human poses, where the layouts of body parts described by the same atomic pose are similar. Intuitively, human pose estimation performance can be improved by using a prior which is learned from the images of the same atomic pose, as compared to relying on a single model for all the images. Therefore, we estimate the spatial relationship between body parts in the pictorial structure [38] model for each atomic pose respectively, which will be used in our model inference stage (Section 6.3.4).

6.3.3 Model learning

Our model (Eqn.6.1) is a standard Conditional Random Field (CRF) with no hidden variables. We use a maximum likelihood method with Gaussian priors to learn the model parameters $\{\zeta, \eta, \gamma, \alpha, \beta, \lambda\}$. All object detectors and body part detectors are trained using the deformable parts model [37], while the action classifier is trained using the spatial pyramid method [71]. A constant 1 is appended to each feature vector so that the model can learn biases between different classes.

Conditioned on the image appearance information in $\phi_2 \sim \phi_5$, our model learns the strength of the compatibility between a set of actions, objects, and human poses in ϕ_1. Figure 6.3 visualizes the connectivity structure learned from the sports

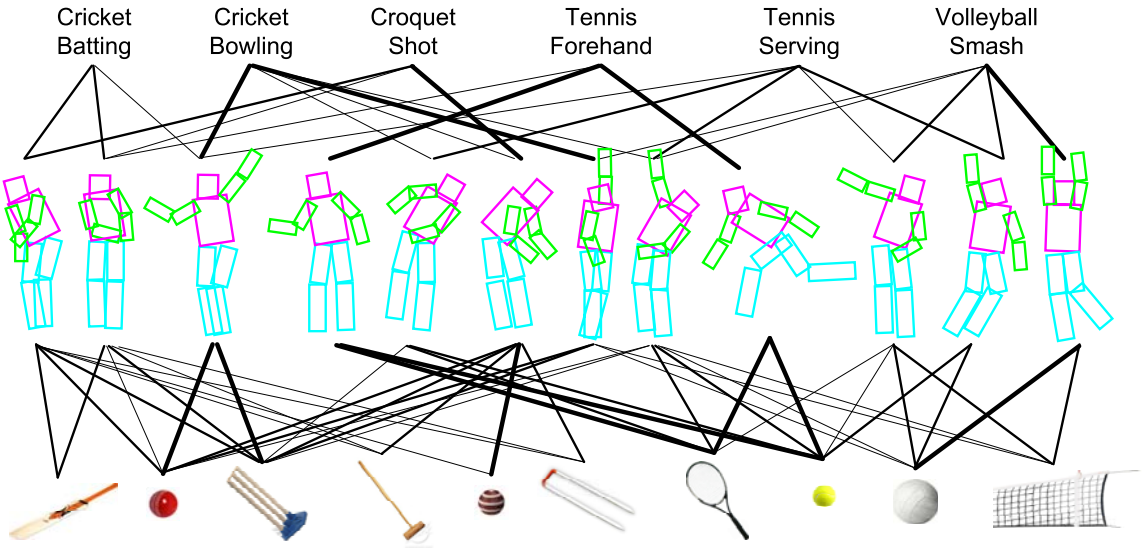


Figure 6.3: The learned connectivity map of actions, poses, and objects using the sports dataset. Thicker lines indicate stronger connections while thinner connections indicate weaker connections. We did not show the connections between actions and objects because they are tricky (e.g. "tennis serving" connects with "tennis ball" and "tennis racket"). We also ignore connections that are very weak.

dataset (described in Section 6.4.1). Each connection is obtained by marginalizing ζ in Eqn.6.2 with respect to the other concept, e.g. the strength of the connection between pose h_i and object o_j is estimated by $\sum_{k=1}^{N_a} \exp(\zeta_{i,j,k})$.

Figure 6.3 shows that our method learns meaningful action-object-pose interactions, such as the connection between "tennis forehand" and the fourth atomic pose which is a reasonable gesture for the action, the object "volleyball" and the last atomic pose, etc.

6.3.4 Model inference

Given a new image, inference on Eqn.6.1 gives us the results of action classification, object detection, and human pose estimation. We initialize the model inference with the SVM action classification results using the spatial pyramid representation [71], object bounding boxes obtained from independent object detectors [37], as well as initial pose estimation results from a pictorial structure model [102] estimated from

all training images, regardless of the belongingness of different atomic poses. We then iteratively perform the following three steps until a local maximum of $\Psi(V, O, H, I)$ is reached.

Updating the layout of human body parts. From the current inference result, we compute the marginal distribution of the human pose over all atomic poses: $\{p_{(H=h_i)}\}_{i=1}^{N_h}$. From this distribution, we refine the prior of the joint location of each body part l in this image using a mixture of Gaussians $\sum_{i=1}^{N_h}[p_{(H=h_i)} \cdot \mathcal{N}(\mathbf{x}_{h_i}^l)]$, where $\mathcal{N}(\mathbf{x}_{h_i}^l)$ is the prior distribution for body part l in the i-th atomic pose estimated in Section 6.3.2. Furthermore because the pictorial structure inference can be very efficient if the part dependencies are Gaussians, we use a Gaussian distribution to approximate each mixture of Gaussians. Then we use pictorial structure with these new Gaussian distributions to update the pose estimation results.

Updating the object detections. With the current pose estimation result as well as the marginal distribution of atomic poses and action classes, we use a greedy forward search method [21] to update the object detection results. We use (m, j) to denote the score of assigning the m-th object bounding box to object o^j, which is initialized as

$$\begin{aligned}(m, j) = &\sum_{i=1}^{N_h}\sum_{l=1}^{L} p_{(H=h_i)} \cdot \lambda_{i,j,l}^T \cdot b\left(\mathbf{x}_H^l, O^m\right) \\ &+ \sum_{i=1}^{N_h}\sum_{k=1}^{N_a} p_{(H=h_i)} \cdot p_{(A=a_k)} \cdot \zeta_{i,j,k} + \gamma_j^T \cdot g(O^m)\end{aligned} \tag{6.7}$$

Initializing the labels of all the windows to be background, the forward search repeats the following steps

1. Select $(m^*, j^*) = \arg\max\{(m, j)\}$.

2. Label the m^*-th object detection window as o_{j^*} and remove it from the set of detection windows.

3. Update $(m, j) = (m, j) + \gamma_{j,j^*}^T \cdot b(O^m, O^{m^*}) + \gamma_{j^*,j}^T \cdot b(O^{m^*}, O^m)$.

until $(m^*, j^*) < 0$. After this step, all object bounding boxes are assigned to either an object label or the background.

Updating the action and atomic pose labels. Based on the current pose estimation and object detection results, we optimize $\Psi(A, O, H, I)$ by enumerating all possible combinations of A and H labels.

6.3.5 Computing action distance

Based on our model inference results, we measure the distance between two action images considering not only action classes but also objects and human poses in the action. For an image I, we use our mutual context model to infer marginal distributions on the action classes $p(A|I)$ and atomic poses $p(H|I)$ respectively. We also obtain a N_o-dimensional vector whose j-th component is set to 1 if the object o_j is detected in image I, or 0 otherwise. We normalize this vector to obtain a distribution $p(O|I)$ for all the objects in this image. We then measure the distance between two images I and I' by

$$2 \cdot D(p(A|I), p(A|I')) + D(p(O|I), p(O|I')) + 2 \cdot D(p(H|I), p(H|I')) \tag{6.8}$$

where D (described below) indicates the distance between two probability distributions. We assign a lower weight to objects because the performance of object detection is not as good as action classification and pose estimation (Section 6.4.2). In this chapter we consider two distance measures (D) for probabilities:

- Total variance $T(\mathbf{p}, \mathbf{q}) = \sum_i |p_i - q_i|$.
- Chi square statistic $\chi^2(\mathbf{p}, \mathbf{q}) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i}$.

Note that our model (Section 6.3.1) jointly considers human actions, objects, and human poses, and therefore the probability distribution estimated from each of them considers image appearance as well as contextual information from the other two. Our distance measure further takes into account the three components together. In

Section 6.4.4 we show that our approach captures much semantic level differences between images of human actions and the results are largely consistent with human perceptions as shown in Figure 6.2.

6.4 Experiment

6.4.1 The six-class sports dataset

We carry out experiments on the six-class sports dataset [49]. For each action there are 30 training images and 20 testing images. The objects that we consider are: cricket bat, ball, and stump in "cricket batting" and "cricket bowling"; croquet mallet, ball, and hoop in "croquet shot"; tennis racket and ball in "tennis forehand" and "tennis serving"; volleyball and net in "volleyball smash".

We train an upper-body detector on this dataset using citeFelzenszwalb10. The detector works almost perfectly because of the relatively clean image background. We normalize the images based on the size of the detection boxes such that we do not need to search over scales in human pose estimation. We obtain 12 atomic poses on this dataset (shown in Figure 6.3).

6.4.2 Action classification, object detection, and pose estimation

The action classification results are shown in Figure 6.4. We also compare our method with other approaches for object detection and human pose estimation in Table 6.1 and Table 6.2. Following the convention in citeFerrari08, a body part is considered correctly localized if the endpoints of its segment lie within 50% of the ground-truth segment length from their true positions. As in PASCAL VOC [27], an object detection bounding box is considered correct if the ratio between its intersection with the ground truth and its union with the ground truth is greater than 50%.

We observe that our method achieves better performance than the baselines in almost all experiments. We obtain better action classification and pose estimation

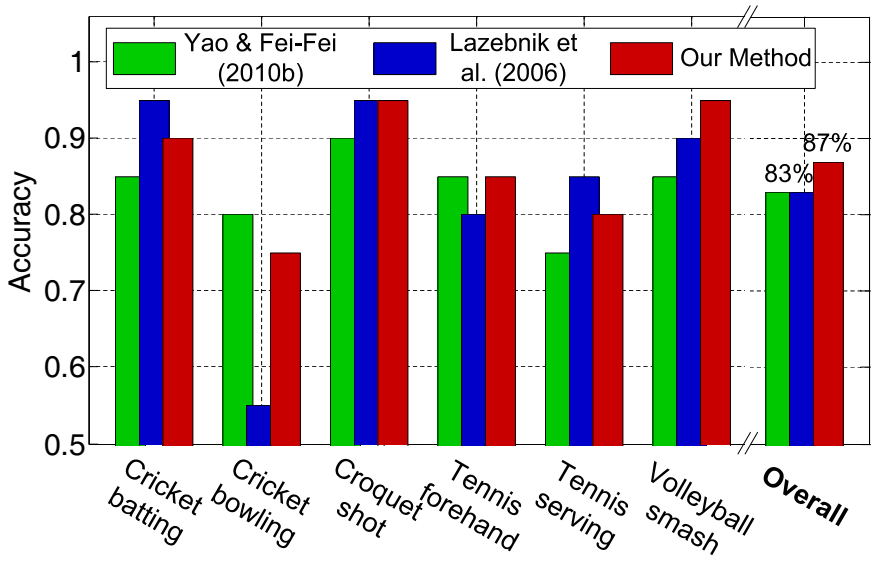


Figure 6.4: Action classification performance of different methods on the sports dataset.

Method	Felzenszwalb et al. [37]	Desai et al. [21]	Our Method
Cricket bat	17%	18%	**20%**
Cricket ball	24%	27%	**32%**
Cricket stump	77%	**78%**	77%
Croquet mallet	29%	32%	**34%**
Croquet ball	50%	52%	**58%**
Croquet hoop	15%	17%	**22%**
Tennis racket	33%	31%	**37%**
Tennis ball	42%	46%	**49%**
Volleyball	64%	65%	**67%**
Volleyball net	4%	6%	**9%**
Overall	36%	37%	**41%**

Table 6.1: Object detection results on the sports data measured by detection accuracy. We bold the best performance in each experiment. The object detection result is not directly comparable to that of the last chapter, because in this chapter we detect each object in all testing images, while in that chapter the object is only detected in images of the action classes that could contain the object (e.g. detecting "volleyball" in "volleyball smash" images).

Method	Yao & Fei-Fei [127]	Andriluka et al. [2]	Our Method
Head	58%	71%	**76%**
Torso	66%	69%	**77%**
Left/right upper arms	44%	44%	**52%**
	40%	40%	**45%**
Left/right lower arms	27%	35%	**39%**
	29%	36%	**37%**
Left/right upper legs	43%	58%	**63%**
	39%	**63%**	61%
Left/right lower legs	44%	59%	**60%**
	34%	71%	**77%**
Overall	42%	55%	**59%**

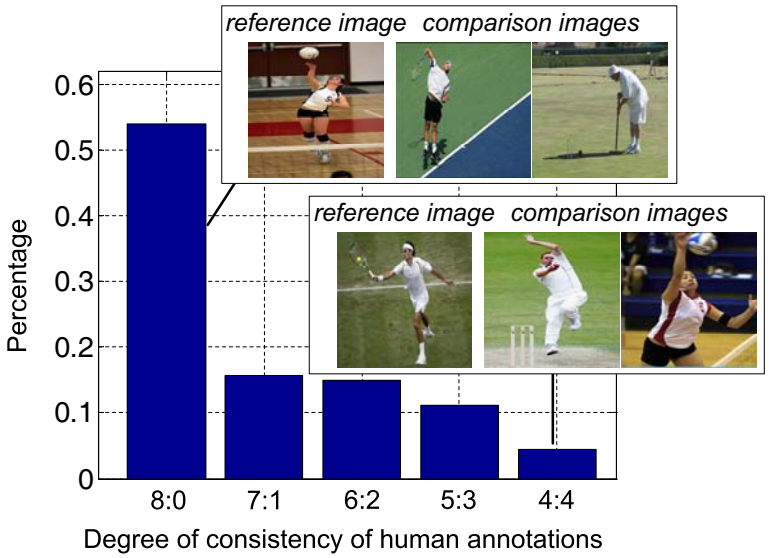
Table 6.2: Pose estimation results on the sports data measured by average precision. We bold the best performance in each experiment.

results compared to the previous chapter because we use stronger body part detectors and incorporate the discriminative action classification component in the model of this chapter. Please refer to the previous chapter for more analysis and comparison of the mutual context model and the other approaches.

6.4.3 Human perception of action distances

Before we evaluate our distance metric described in Section 6.3.5, we study how humans measure the similarity between action images. First, we are interested in whether humans agree with one another on this task. In every trial of our experimental study, we give a human subject one reference image and two comparison images (as shown in Figure 6.5(a)), and ask the subject to annotate which of the two comparison images is more similar to the reference image. We generate two trials of experiments for every possible combination of action classes from the sports dataset, and therefore our experiment consists of $2 \times (6 + C_{6,2}) = 252$ trials. We give the same 252 trials to eight subjects.

Figure 6.5(a) summarizes the consistency of the received responses. We observe that in most situations the eight subjects agree with each other (54% 8:0 as compared to 4% 4:4), even in many confusing trials. For example as shown in Figure 6.5(a), all



(a) X-axis represents the degree of consistency when humans measure the similarity between different action images, e.g. "7:1" means seven of the eight subjects have the same annotation in a given trial. Y-axis is the percentage of the corresponding trials in all the 252 trials.

(b) Examples of action similarities obtained from human annotation. In each row, the reference image is indicated by a yellow bounding box. The magenta numbers are the similarity with the corresponding reference image.

Figure 6.5: Human annotations of action distance.

eight subjects believe the "volleyball smash" image is closer to the "tennis forehand" image than the "croquet shot" image because the former two images have similar human poses.

Having shown that humans tend to give similar annotations in measuring the similarity between different action images, we obtain the ground truth similarity between action images by averaging annotations from different human subjects. We give each subject an annotation task where an image I^{ref} is treated as the reference image for 50 trials. In each trial we randomly select two different test images to compare with I^{ref}. Five of the eight subjects are assigned this task, resulting in 250 pairwise rankings of the 120 test images for I^{ref}. We then use the edge flow method [60] to convert these pairwise rankings to a similarity vector $\mathbf{s} = \{s(I^{ref}, I^i)\}_{i=1}^{120}$, where $s(I^{ref}, I^i)$ denotes the ground truth similarity between I^{ref} and I^i. We obtain $\mathbf{s}$ by solving an optimization problem

$$\begin{aligned} &\text{minimize } \mathbf{M} \cdot \mathbf{s} = \mathbf{1} \\ &s.t.\ \mathbf{s} \succeq 0,\ \|\mathbf{s}\|_2 \leq 1 \end{aligned} \tag{6.9}$$

where $\mathbf{M}$ is a 250×120 sparse matrix where $M_{j,k} = 1$ and $M_{j,l} = -1$ if the j-th pairwise ranking indicates that I^k is more similar to I^{ref} than I^l.

We repeat the above procedure to obtain a similarity vector for each test image. Figure 6.5(b) shows examples of action similarities. Note that $s(I^{ref}, I^i)$ is asymmetric because we obtain the similarity values by treating each test image as the reference image separately.

6.4.4 Evaluating the distance metric

In this section, we evaluate the approaches of computing the distance between different action images. With the ground truth similarities of each reference image against all the other images obtained from human annotation (Section 6.4.3), our goal is to automatically find the images that correspond to large similarity (small distance) values.

Our distance metric is evaluated in the following way. Denote the ground truth

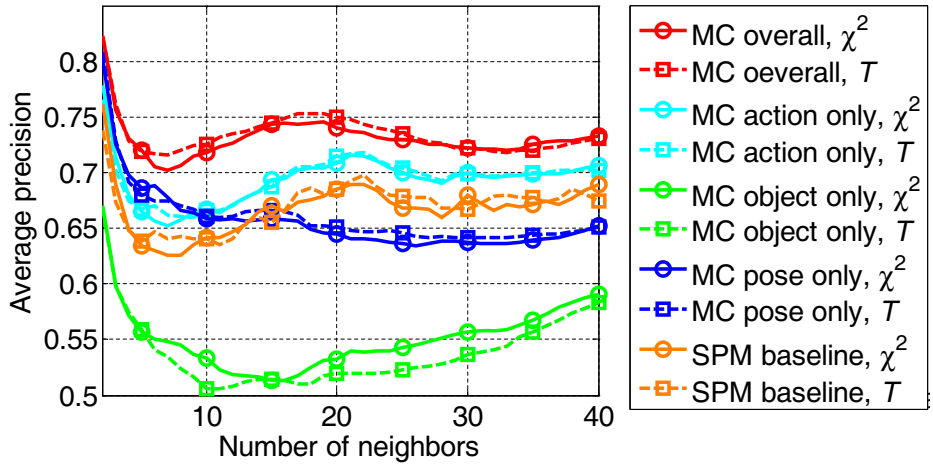


Figure 6.6: Comparison of different distance metrics evaluated by average precision with respect to the number of similar images in top of the ranking. "MC" denotes "mutual context" and "SPM" is "spatial pyramid matching".

similarity between an image I and the reference image I^{ref} as $s(I^{ref}, I)$. We have a ground truth ranking of all the images $\{I^{gt_1}, I^{gt_2}, \cdots\}$ such that $s(I^{ref}, I^{gt_i}) \geq s(I^{ref}, I^{gt_j})$ if $i \leq j$. Using our distance metric we obtain another ranking of all the images $\{I^{re_1}, I^{re_2}, \cdots\}$ by sorting their distance with the reference image in ascending order. The precision of using this distance metric to find n neighboring images for I^{ref} is evaluated by

$$\frac{\sum_{i=1}^{n} s(I^{ref}, I^{re_i})}{\sum_{i=1}^{n} s(I^{ref}, I^{gt_i})} \tag{6.10}$$

Average precision of using all the test images as reference images is adopted for performance evaluation.

We compare our distance metric (Eqn.6.8) with a baseline approach that is based on spatial pyramid image classification [71]. In that approach, an image is represented by the six-dimensional confidence scores obtained from one-vs-all SVM classification. The distance between the confidence scores is used to measure the distance between two images. We also compare our method with some control approaches that use each of the three components (action, object, and human pose) of Eqn.6.8 individually.

Figure 6.7: Comparison of our distance metric and the baseline on a "tennis serving" image. The top-left image surrounded by a yellow rectangle is the reference image, and all the other images are organized in a row major order with respect to ascending distance values to the corresponding reference image.

We observe from Figure 6.6 that our method outperforms the baseline and all the other control settings. The two probability distance measures, χ^2 and T, achieve very similar performance in all the methods. Among the three components, using actions only performs the best while using objects only performs the worst. The reason might

Figure 6.8: Comparison of our distance metric and the baseline on a "volleyball smash" image. The top-left image surrounded by a yellow rectangle is the reference image, and all the other images are organized in a row major order with respect to ascending distance values to the corresponding reference image.

be that, objects are usually small such that the human annotations put less weights to objects compared with that of actions or human poses. Also, Table 6.1 shows that object detection does not perform as well as pose estimation or action classification, making it less reliable when using objects only for distance computation.

Figure 6.7 and Figure 6.8 shows the top 20 images obtained using our method and the baseline spatial pyramid method. We observe that our results are significantly more consistent with human perception. Our method can not only find the images that have the same action as the reference image, but also capture the detailed similarity of semantic meaning such as human pose. For example, in Figure 6.8, the "volleyball smash" image returns 17 images of the same action, and the humans in the next 3 images have similar poses as the human in the reference image.

6.5 Summary

In this chapter, we show that the joint modeling of actions, objects, and human poses can not only improve the performance of action classification, object detection, and pose estimation, but also lead to an action distance measure approach whose output is largely consistent with human annotations.

Chapter 7

Discovering Object Functionality

In the previous two chapters, we have shown that we can do many tasks other than action classification in still images, such as object detection, human pose estimation, and action retrieval. In this chapter,[1] we aim for a higher level goal - discovering object functionality. We achieve this goal with a weakly supervised approach.

7.1 Introduction

What is an object? Psychologists have proposed two popular philosophies of how humans perceive objects. One view asserts that humans perceive objects by their physical qualities, such as color, shape, size, rigidity, etc. Another idea was proposed by Gibson [44], who suggested that humans perceive objects by looking at their affordances. According to Gibson and his colleagues [43, 12], affordance refers to the quality of an object or an environment which allows humans to perform some specific actions. Recent studies [88] have shown that affordance is at least as important as appearance in recognizing objects by humans. An example is shown in Figure 7.1.

In the field of computer vision, while most previous work has emphasized modeling the visual appearances of objects [39, 16, 37], research on object/scene affordance (also called functionality[2]) is attracting more and more researchers' attention recently [48,

[1] An early version of this chapter has been presented in [133].

[2] There are subtle differences between affordance and functionality in psychology. But in this

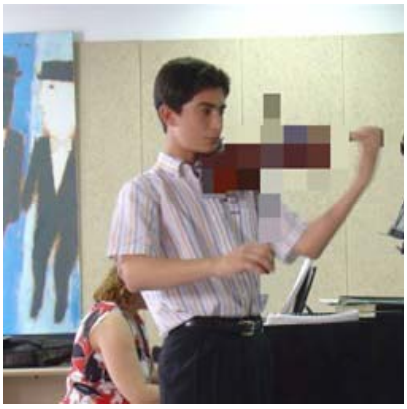



Figure 7.1: Humans can use affordance to perceive objects. In the left image, although the violin is almost invisible, most humans can easily conclude this is an image of a human playing a violin based on the way the human interacting with the object. However, it is difficult to recognize the object with mosaic in the image on the right.

65, 46, 129, 41]. On the one hand, observing the functionality of an object (e.g. how humans interact with it) provides a strong cue for us to recognize the category of the object. On the other hand, inferring object functionality itself is an interesting and useful task. For example, one of the end goals in robotic vision is not to simply tell a robot "this is a violin", but to teach the robot how to make use of the functionality of the violin - how to play it. Further, learning object functionality also potentially facilitates other tasks in computer vision (e.g. scene understanding [17, 41]) or even the other fields (e.g. exploring the relationship between different objects [36]).

In this chapter, our goal is to discover object functionality from weakly labeled images. Given an object, there might be many ways for a human to interact with it, as shown in Figure 7.2. As we will show in our experiments, these interactions provide us with some knowledge about the object and hence reveal the functionalities of those objects. Furthermore, while inferring these types of interactions, our method also builds a model tailored to object detection and pose estimation for each specific interaction.

We propose an iterative model to achieve our goals. Using violin as an example, given a set of images of human-violin interactions, we discover different types of

chapter, we use them interchangeably.

Figure 7.2: There are multiple possible modes of interactions between a human and a given object. Some interactions correspond to the typical functionality of the object while others do not.

human-violin interactions by first estimating human poses and detecting objects, and then clustering the images based on their pairwise distances in terms of human-object interactions. The clustering result can then be used to update the model of human pose estimation and object detection, and hence human-violin interaction. Compared with previous human-object interaction and affordance work, we highlight the following properties of our approach:

- **Same object, multiple interactions:** Our method takes into account the fact that humans might interact with the same object in different ways, with only some typical interactions corresponding to object affordance, as shown in Figure 7.2. This differs from most previous approaches that assume a single type of human-object interaction for each object [48, 65].

- **Weak supervision:** Comparing with [49, 129], our method does not require annotations of human poses and objects on every training image. We only need a general human pose estimator and a weak object detector trained from a small subset of training images, which will be updated by our iterative model.

- **Unconstrained human poses:** Rather than being limited to a small set of pre-defined poses such as sitting and reaching [46, 41], our method does not have any constraint on human poses. This allows us to learn a larger variety of human-object interactions.

- **Bridging the gap between 2D and 3D:** Considering that the same human-object interaction might lead to very different 2D appearances (Figure 7.3) because of different camera angles from which the images are taken, we convert 2D human poses to 3D and then measure the similarity between different images. This allows us to obtain more semantically meaningful clusters as compared to previous work [129, 94].

- **Aiming for details:** The functionality we learn refers to the details of human-object interactions, e.g. the pose of the human, the object, as well as how the object should be used by humans. This makes our work different from most previous functionality work which mainly focuses on object detection [65, 94].

The rest of the chapter is organized as follows. Section 7.2 introduces related work, then Section 7.3 elaborates on our approach of weakly supervised functionality discovery. Section 7.4 demonstrates experimental results.

7.2 Related work

Functionality (affordance) for object recognition. Recently, functionality has been used to detect objects [48, 65], where human gestures are recognized and treated as a cue to identify objects. In [46], 3D information is deployed such that one can recognize object affordance even when humans are not observed in test images. Such approaches assume that an object has the same functionality across all images, while our method attempts to infer object functionality given that humans might interact with the same object in different ways.

Human context. Context has been widely used in various computer vision tasks [96, 85]. Specifically, because of the advances in human detection [16, 37] and

Figure 7.3: The same human pose might lead to very different appearances and 2D spatial configurations of body parts because of variations in camera angle.

human pose estimation [2, 124], humans are frequently used as cues for other tasks, such as object detection (details below) and scene reconstruction [50, 17, 41]. Humans can also serve as context for each other to obtain performance improvement on all humans [26]. In this chapter, we use human poses as context to discover functionalities of objects.

Human-object interaction. Our method relies on modeling human-object interactions. Most such approaches first estimate human poses [2, 124] and detect objects [37], and then model human-object spatial relationships to improve action recognition performance [19, 49, 129]. There are also approaches that directly train components of human-object interactions [32]. While those approaches usually require detailed annotations on training data, a weakly supervised approach is adopted in [94] to infer the spatial relationship between humans and objects. While our method also uses weak supervision to learn the spatial relationship between humans and objects, it takes into account that humans can interact with the same object in different ways, which correspond to different semantic meanings.

Semantic image clustering. In this chapter, we use a clustering approach to discover different human-object interactions. Unsupervised learning of object classes from unlabeled data has been explored in object recognition [101]. Recently, unsupervised object clustering [101] has been used to improve the performance of object classification. In this work, we cluster human action images in 3D, where the clustering results are more consistent with human perception than those from 2D.

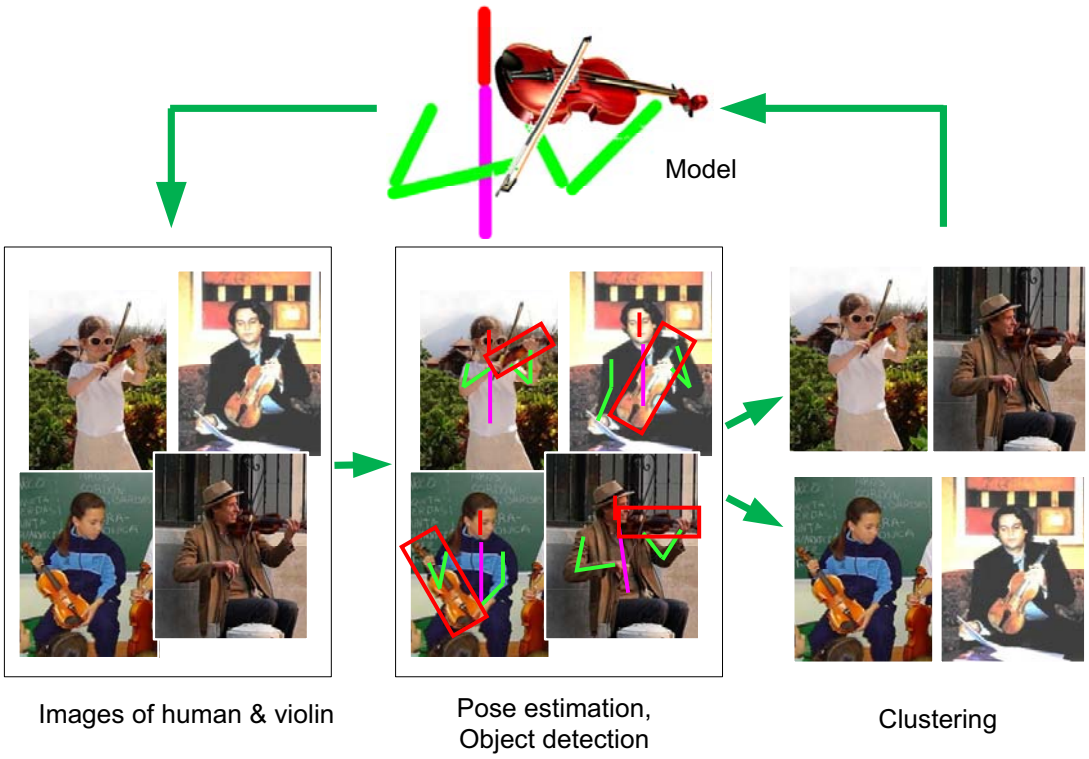


Figure 7.4: An overview of our approach ("violin" as an example). Given a set of images of human-violin interactions, our goal is to figure out what are the groups of interactions between a human and a violin, and output a model for this action.

7.3 Algorithm

7.3.1 Overview

As shown in Figure 7.2, there are many possible ways for a human to interact with an object. Different interactions, such as playing a violin or using a violin as a weapon, correspond to different object functionalities. Our goal is to discover those interactions from weakly supervised data.

An overview of our approach is shown in Figure 7.4. Given a set of images of humans interacting with a certain object and an initial model of object detection and pose estimation, we propose an iterative approach to discover different types of human-object interactions and obtain a model tailored to each interaction. On the

one hand, given a model of object functionality, we detect the object, estimate the human pose, convert 2D key points to 3D, and then measure the distance between each pair of images (Section 7.3.2). The pairwise distance can then be used to decide which interaction type does each image belong to (Section 7.3.3). On the other hand, given the clustering results, both the object detectors and human pose estimators can be updated so that the original model can be tailored to specific cases of object functionality (Section 7.3.4).

7.3.2 Pairwise distance of human-object interactions

To reduce the semantic gap between human poses and 2D image representation (shown in Figure 7.3), we evaluate the pairwise distance of human-object interactions in the three-dimensional space. The pipeline we use to compute similarity between two images is shown in Figure 7.5. First, the 2D locations and orientations of objects and human body parts are obtained using off-the-shelf object detectors [37] and human pose estimation [124] approaches. Coordinates of 2D key points are then converted to 3D [97], and we evaluate pairwise distance between images by aligning 3D perspectives and computing the sum of squared distances between the corresponding body parts [128] and objects.

Object detection. We use the deformable parts model [37] to detect objects. To get more detailed information about human-object interactions, our detector also takes object orientation into consideration, as shown in Figure 7.4. At training time, we provide rectified bounding boxes with upright objects as positive training examples, and treat all the other image windows without the object or with non-upright objects as negative examples. During detection, we rotate the image using 12 different orientations and apply the trained detector in each case. Non-maximum suppression is done by combining the detection results on all orientations.

2D pose estimation. We use the flexible mixture-of-parts [124] approach for 2D human pose estimation. This approach takes the foreshortening effect into consideration, which facilitates the generation of 3D poses. We consider six body parts for the upper body of humans: head, torso, left/right upper arms, and left/right

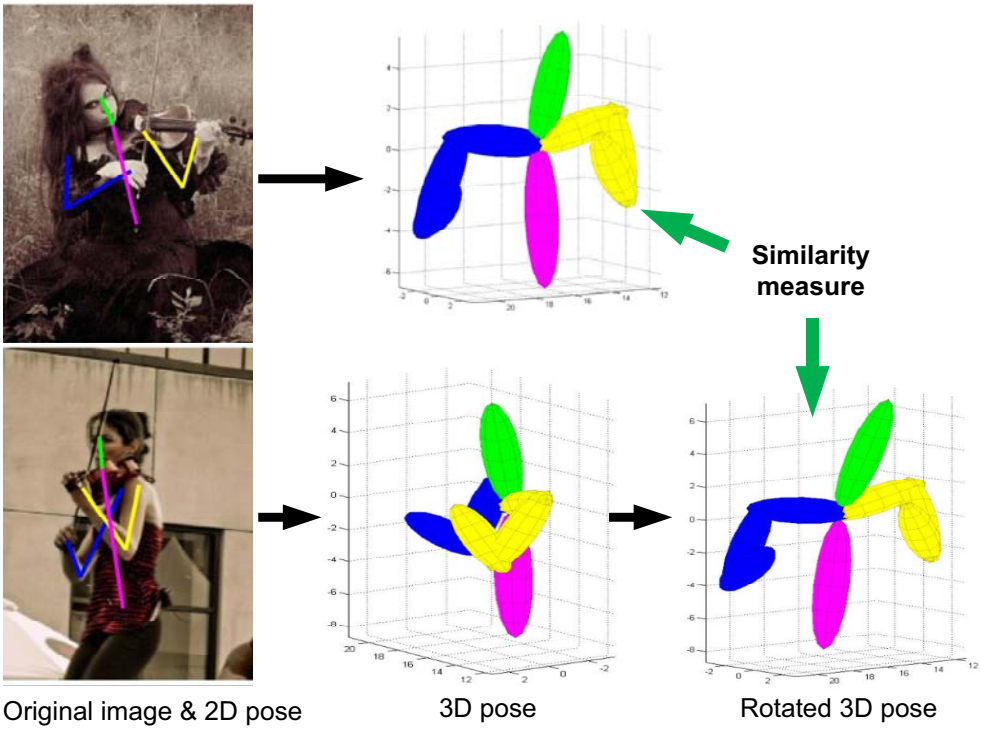


Figure 7.5: The pipeline we use to compute the similarity between two images of human-object interaction. We first detect objects and estimate human poses in each image, and then convert the key point coordinates in 3D and measure image similarity.

lower arms, as shown in Figure 7.5. For full-body humans, we also consider left/right upper legs and left/right lower legs. To improve performance, we replace the part filters with strong body-part detectors trained using the deformable parts model [37].

3D reconstruction of human pose. Because of camera angle changes, the same human pose might lead to very different 2D configurations of human body parts, as shown in Figure 7.3. Therefore, we use a data-driven approach to reconstruct 3D coordinates of human body parts from the result of 2D pose estimation [97]. By leveraging a corpus of 3D human body coordinates (e.g. CMU MOCAP), we recover 3D human body coordinates and camera matrices using a sparse linear combination of atomic poses. For the 3D locations of detected objects, we search the nearest body parts in 2D space, and average their 3D locations as the locations of objects in 3D

space.

Pairwise distance computation. It has been shown that pose features perform substantially better than low-level features in measuring human pose similarities [40]. Following this idea and inspired by [128], we measure the distance of two human poses by rotating one 3D pose to match the other, and then consider the point-wise distance of the rotated human poses. Mathematically, let $\mathbf{M}_1$ and $\mathbf{M}_2$ be the matrices of the 3D key-point locations of two images $\mathbf{x}_1$ and $\mathbf{x}_2$. We find a rotation matrix $\mathbf{R}^*$ such that

$$\mathbf{R}^* = \arg\min_R \|\mathbf{M}_1 - \mathbf{M}_2\mathbf{R}\|^2, \tag{7.1}$$

and the similarity between $\mathbf{M}_1$ and $\mathbf{M}_2$ can be computed by

$$\mathcal{D}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{M}_1 - \mathbf{M}_2\mathbf{R}^*\|^2. \tag{7.2}$$

We further incorporate the object in our similarity measure by adding the object as one more point in $\mathbf{M}$ and assuming that the depth of the object is the same as the hand that is closest to the object.

7.3.3 Clustering based on pairwise distance

The goal here is to cluster the given images so that images in the same cluster correspond to similar human-object interactions, as shown in Figure 7.4. However, the task is not straightforward, since we only have the pairwise distance between images, rather than having a feature representation for each image.

We use an approach similar to spectral clustering [82] to address this issue. First, we use kernel principal component analysis (kernel PCA) [104] to project each image $\mathbf{x}$ into a principal component space while keeping the pairwise image similarity computed from Section 7.3.2. Denote the $N \times N$ similarity matrix as $\mathbf{S}$, where

$$\mathbf{S}_{ij} = \frac{1}{\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) + \epsilon}, \epsilon > 0 \tag{7.3}$$

is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Assuming an unknown feature representation for $\mathbf{x}_i$ as $\Phi(\mathbf{x}_i)$, we have the covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T. \tag{7.4}$$

Performing PCA, we have $\lambda_k\mathbf{v}_k = \mathbf{C}\mathbf{v}_k$, where λ_k is the k-th largest eigenvalue of $\mathbf{C}$. There also exist coefficients $\alpha_{k,1}, \cdots, \alpha_{k,N}$ such that

$$\mathbf{v}_k = \sum_{i=1}^{N}\alpha_{k,i}\Phi(\mathbf{x}_i). \tag{7.5}$$

Since $\mathbf{S}_{ij} = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$, the projection of $\Phi(\mathbf{x}_l)$ on $\mathbf{v}_k$ can be written as

$$z_{l,k} = \mathbf{v}_k^T\Phi(\mathbf{x}_l) = \sum_{i=1}^{N}\alpha_{k,i}\mathbf{S}_{il}. \tag{7.6}$$

According to [104], $\boldsymbol{\alpha}_k = [\alpha_{k,1}, \cdots, \alpha_{k,N}]^T$ can be computed by solving

$$\begin{aligned} N\lambda_k\boldsymbol{\alpha}_k &= \mathbf{S}\boldsymbol{\alpha}_k, \\ \text{s.t.} \quad \boldsymbol{\alpha}_k^T\boldsymbol{\alpha}_k &= 1/\lambda_k. \end{aligned} \tag{7.7}$$

Given the projected vector $\mathbf{z}_i$ for each image $\mathbf{x}_i$, we perform k-means clustering on all $i = 1, \cdots, N$ to form clusters of human-object interactions. Our approach chooses an appropriate number of clusters for every step of the process by using the standard elbow method - a cluster number is chosen such that adding another cluster does not give much decrement of the k-means objective. Since the above computation requires $\mathbf{S}$ to be positive semidefinite, we use a matrix approximation to replace $\mathbf{S}$ with $\hat{\mathbf{S}}$ such that

$$\hat{\mathbf{S}} = \arg\min \|\mathbf{S} - \hat{\mathbf{S}}\|^2, \tag{7.8}$$

where $\hat{\mathbf{S}}$ is positive semidefinite. We also assumed $\Phi(\mathbf{x}_i)$ to be centered in the above

derivation. Please refer to [104] for details of how to drop this assumption.

7.3.4 Updating the object functionality model

In each iteration, we update the model of object detection and pose estimation for each cluster of human-object interaction. In each cluster, we re-train the models by using object detection and pose estimation results from this iteration as "ground-truth". Although there will be mistakes in these detection and estimation results, putting all the images together can still provide us more accurate priors that are tailored to each cluster.

In the step of object detection and pose estimation in the next iteration, we apply all the models from different clusters, and choose the one with the largest score of object detection and pose estimation. The detectors and estimators from different clusters are calibrated by fitting a probability distribution to a held-out set of images, as in [93].

7.4 Experiments

7.4.1 Dataset and experiment setup

For performance evaluation, we need a dataset that contains different interactions between humans and each object. The People Playing Musical Instrument (PPMI) dataset [126] contains images of people interacting with twelve different musical instruments: bassoon, cello, clarinet, erhu, flute, French horn, guitar, harp, recorder, saxophone, trumpet, and violin. For each instrument, there are images of people playing the instrument (PPMI+) as well as images of people holding the instrument with different pose, but not performing the playing action (PPMI-). We use the normalized training images to train our models, where there are 100 PPMI+ images and 100 PPMI- images for each musical instrument.

For each instrument, our goal is to cluster the images based on different types of human-object interactions, and obtain a model of object detection and pose estimation for each cluster. Ideally, images of humans playing the instruments should

Instrument	Object detection		Pose estimation	
	Baseline [37]	Our method	Baseline [124]	Our method
Bassoon	16.4%	21.1%	43.1%	45.5%
Cello	41.9%	44.8%	48.1%	57.4%
Clarinet	11.1%	15.8%	52.0%	55.5%
Erhu	28.2%	33.1%	55.8%	57.8%
Flute	20.3%	23.1%	57.2%	59.7%
French horn	43.2%	43.7%	48.9%	55.1%
Guitar	45.5%	48.0%	40.8%	45.5%
Harp	30.6%	34.6%	41.0%	44.5%
Recorder	13.0%	16.9%	43.2%	51.5%
Saxophone	36.0%	41.9%	54.8%	60.7%
Trumpet	22.1%	24.7%	43.1%	48.6%
Violin	33.2%	39.5%	54.3%	63.5%
Overall	28.5%	32.3%	48.5%	53.8%

Table 7.1: Results of object detection and human pose estimation. "Baseline" indicates the results obtained by the original detectors [37] and the general pose estimator [124]. "Ours" indicates the results from the final models obtained from our iterative approach.

be grouped in the same cluster. To begin with, we randomly select 10 images from each instrument and annotate the key point locations of human body parts as well as object bounding boxes, and train a detector [37] for each musical instrument and a general human pose estimator [124]. The object detectors and human pose estimator will be updated during our model learning process.

7.4.2 Object detection and pose estimation

Table 7.1 shows the results of object detection and pose estimation. For each musical instrument, we apply the "final" object detectors and pose estimators obtained from our method to the test PPMI images. For each image, we consider the models that correspond to the largest confidence score. We compare our method with the initial baseline models that are trained for all musical instruments. An object detection result is considered to be correct if the intersection of the result and the ground truth

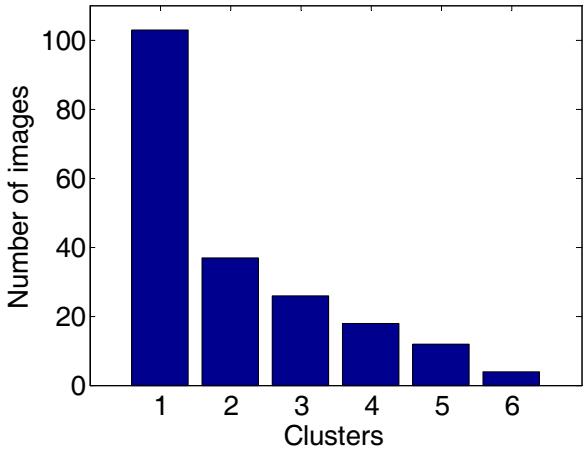


Figure 7.6: Average number of images per cluster on all musical instruments. The clusters are ordered by the number of images they contain.

divided by their union is larger than 0.5, as in [29]. For human pose estimation, a body part is considered correctly localized if the end points of its segment lie within 50% of the ground-truth segment length from their true positions [40].

The results show that our method outperforms the baselines by a large margin. This demonstrates the effectiveness of our approach that iteratively updates pose estimators and object detectors. Furthermore, our pose estimation result (53.8%) even performs slightly better than that in [129] (52.0%), where the models are trained with all PPMI training images annotated. The method in [129] (37.0%) obtains better object detection result than ours (32.3%), but was solving a simpler problem where object orientations were ignored.

7.4.3 Discovering object functionality

The PPMI dataset contains ground truths for which images contain people playing the instrument (PPMI+) and which images contain people only holding the instrument but not playing (PPMI-). This provides us the opportunity to evaluate the quality of clustering results. For each instrument, ideally, there exists a big cluster of humans playing the instrument, and many other clusters of humans holding the instruments but not playing. To get such clusters, we make use of the prior knowledge that

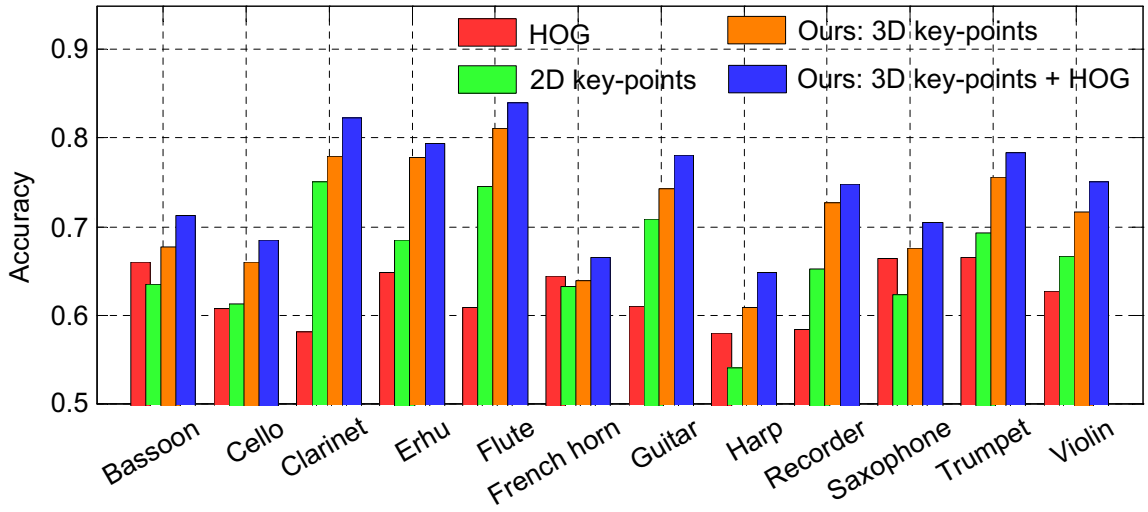


Figure 7.7: Comparison of our functionality discovery method with the approaches that based on low-level features or 2D key point locations.

there are 100 PPMI+ images for each instrument. We choose the number of clusters such that the number of images in the largest cluster is as close to 100 as possible. Figure 7.6 visualizes the average distribution of number of images in each cluster on all musical instruments.

We compare our clustering approach with two baselines. One is based on low-level image descriptors, where we represent an image with HOG [16] descriptors and then use PCA to reduce the feature dimension to 35, and then perform image clustering in the 35-dimensional space. In the other baseline, we cluster images based on 2D positions of key-points of objects and human poses without converting them to 3D. For these two methods, we also choose the number of clusters on each instrument such that the number of images in the largest cluster is as close to 100 as possible.

For each instrument, we assume the largest cluster contains images of people playing the instrument, while all other clusters contain images of people holding the instrument but not playing it. A comparison of the accuracy of the different methods is shown in Figure 7.7. We observe that using 2D key points performs on par with low-level HOG features. The reason might be due to the errors in 2D pose estimation and the lack of accurate pose matching because of camera angle changes. On almost all the instruments, our method based on 3D key point locations

Figure 7.8: Examples of image clusters obtained by our approach. For each instrument, images with the same border color belong to the same cluster. Solid lines indicate images of people playing the instrument (PPMI+) in the ground truth, while dashed lines indicate images of people holding the instrument but not playing it (PPMI-).

significantly outperforms both low-level features and 2D key point locations. The only exception is on French horn, where all three approaches have similar performance. This is due to the large size of French horns, and the fact that the human poses as well as human-object spatial relationship are very similar in images of people playing French horn and people holding French horn but not playing. Finally, the performance can be further improved by combining 3D key points and low-level HOG features, as shown in Figure 7.7.

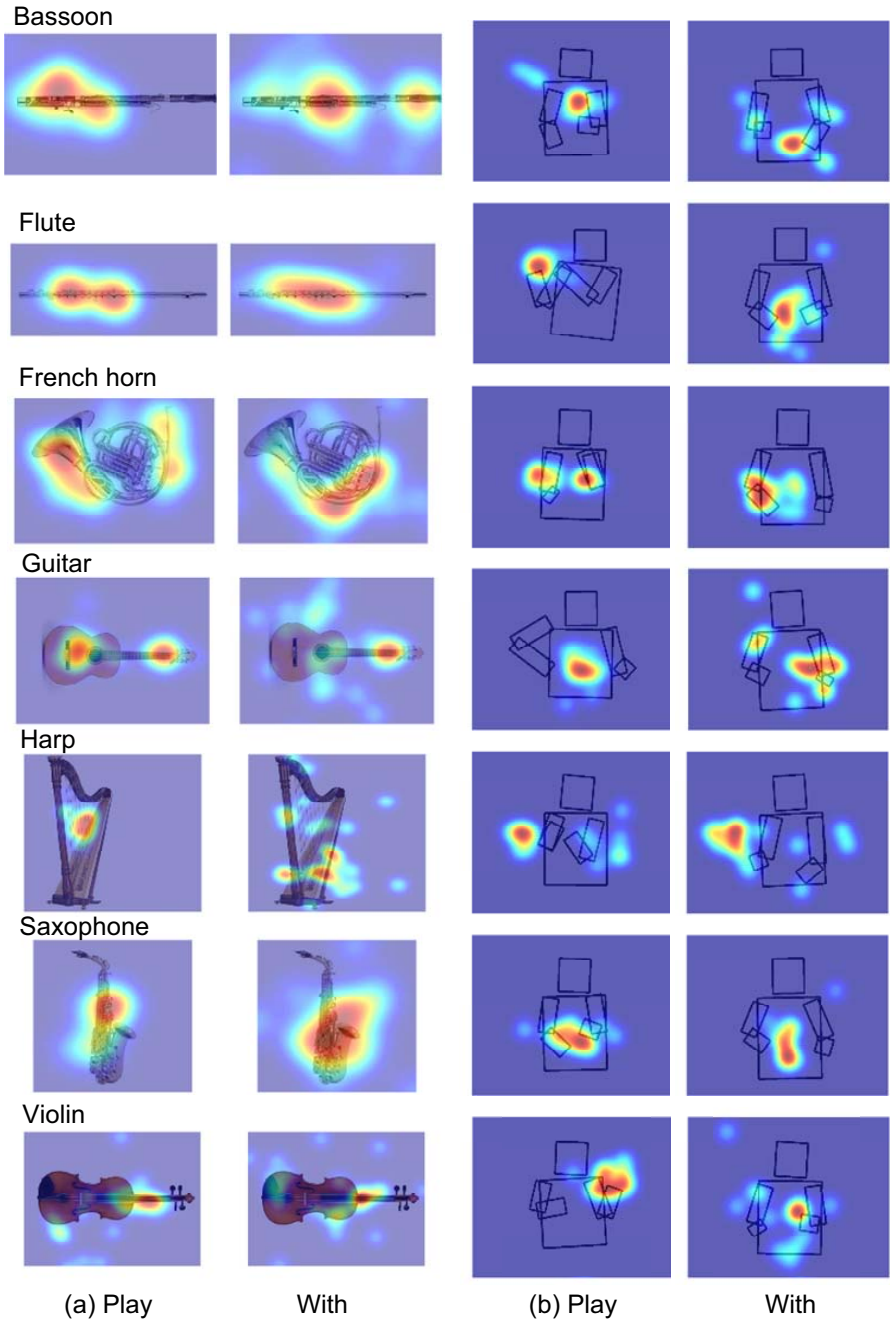


Figure 7.9: **(a)** Heatmaps of the locations of human hands with respect to musical instruments. **(b)** Heatmaps of the locations of objects with respect to the average human pose. For each instrument, "play" corresponds to the largest cluster.

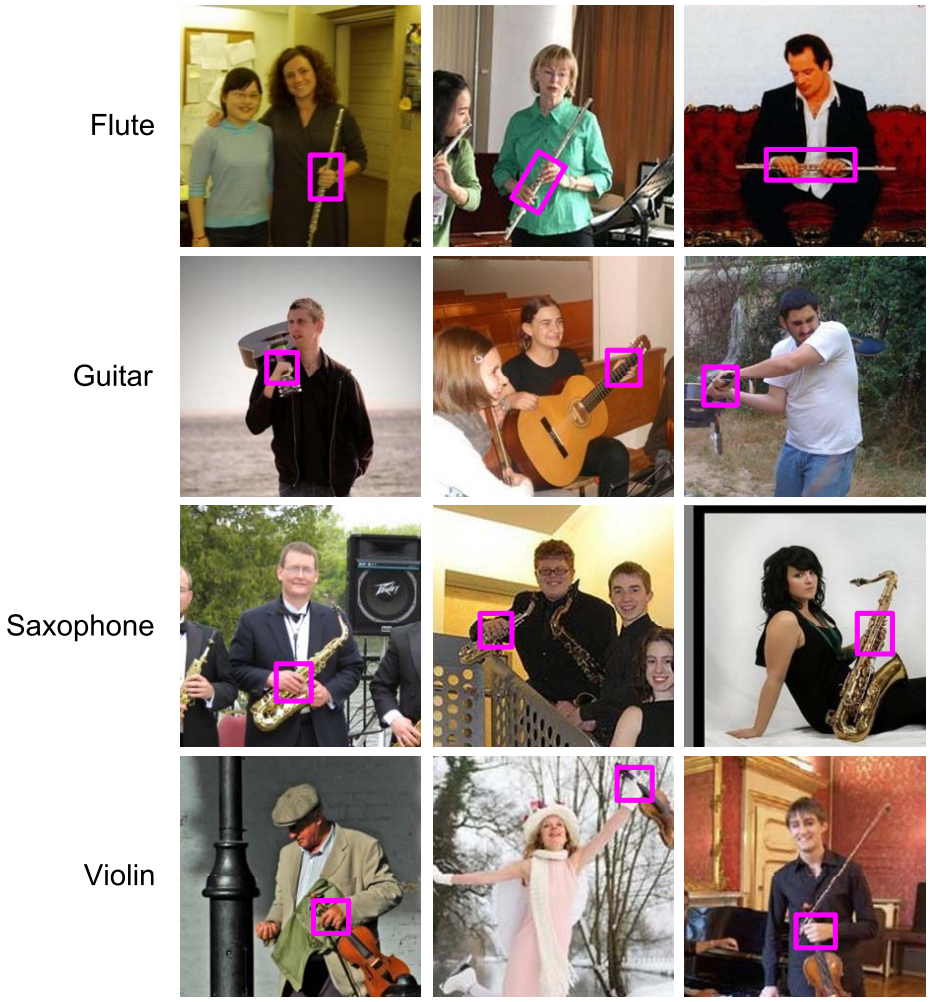


Figure 7.10: Humans tend to touch similar locations of some musical instruments, even when they are not playing it.

7.4.4 Affordance visualization

Examples of image clusters obtained by our approach are shown in Figure 7.8. On the instruments such as flute and trumpet, we are able to separate PPMI+ images from the others with high accuracy, because of the unique human poses and human-object spatial relationships on PPMI+ images. This partly explains why we can obtain high accuracy on those instruments in Figure 7.7. The poor clustering performance on French horn can also be explained from this figure, where the spatial

Instrument	DPM	Ours	Instrument	DPM	Ours
Bassoon	**47%**	38%	Cello	39%	**49%**
Clarinet	32%	**38%**	Erhu	**53%**	23%
Flute	41%	**60%**	French horn	**78%**	37%
Guitar	**46%**	26%	Harp	51%	**53%**
Recorder	32%	**42%**	Saxophone	**53%**	29%
Trumpet	**59%**	53%	Violin	34%	**48%**

Table 7.2: Comparison of using appearance and using human pose to predict object categories. For each instrument, bold fonts indicate better results. Chance performance is 8%.

relationship between humans and French horns are very similar in images of all types of interactions.

Figure 7.9 visualizes the heatmap of the locations of human hands with respect to the musical instruments, as well as the locations of objects with respect to the average human pose in different interactions. On most instruments, we observe more consistent human hand locations on the clusters of people playing the instrument than that on the other clusters. However, we still observe some points that are frequently touched by the humans even for the cases of "holding but not playing" for some instruments, e.g. flute and guitar as shown in Figure 7.10. This shows some general rules when humans interact with a specific type of object, no matter what the functionality of the interaction is. Interestingly, people usually touch different parts of French horn when they are playing or not playing it, as shown in Figure 7.8.

7.4.5 Predicting objects based on human pose

Our method learns the interaction between humans and objects. Given a human pose, we would like to know whether we can infer what object the human is manipulating. On the PPMI test images, we apply all the human pose models to each image, and select the human that corresponds to the largest score. We say that the object involved in the selected model is manipulated by this human. We only consider PPMI+ test images in this experiment. We compare our approach with a baseline

Figure 7.11: Humans might manipulate different objects with very similar poses.

that runs deformable parts models [37] of all instruments on each image, and output the instrument that corresponds to the largest calibrated score. The results are shown in Table 7.2.

Table 7.2 shows that on the musical instruments where the human pose is different from the others, such as flute and violin, our method has good prediction performance. On musical instruments which are played with a similar human pose, such as bassoon, clarinet and saxophone (shown in Figure 7.11), the appearance-based models perform better. This confirms that both object appearance and functionality are important in perceiving objects and provide complementary information [88].

7.5 Summary

In this chapter, we propose a weakly supervised approach to learn object functionality, e.g. how humans interact with objects. We consider multiple possible interactions between humans and a certain object, and use an approach that iteratively clusters images based on object functionality and updates models of object detection and pose estimation. On a dataset of people interacting with musical instruments, we show that our model is able to effectively infer object functionalities.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions

This dissertation deals with action understanding in still images. We have contributed to human action understanding in two major aspects. One is to classify human actions, and the other is to understand human behaviors deeply.

In Chapter 2, 3, and 4, we treated action recognition as an image classification problem. We proposed a representation that captures the structured information of an image by encoding a number of discriminative visual features and their spatial configurations (Chapter 2). For the classification stage, we proposed to combine randomization and discrimination to make a good trade off between classifier bias and variance (Chapter 3). In addition to low level image descriptors, we also used higher level image representations such as action attributes and parts, which further improved the action classification performance (Chapter 4).

The higher level concepts, such as human poses and objects, can not only be used for achieving higher action classification accuracy, but also be directly modeled in the context of human actions. In Chapter 5 and 6, we proposed a mutual context model, which allows human poses and objects to serve as the context to each other and mutually boost each other's performance. In Chapter 5, we considered the interactions between one human and one object. In Chapter 6, we extended the model so that it can deal with the interaction between one human and any number of objects. We

have also shown in Chapter 6 that our method can be used for action retrieval.

Considering the close relationship between humans and objects in human actions, we proposed to learn object functionality by observing human actions in still images. Given a set of images of humans manipulating the same object such as violin, we used a weakly supervised approach to discover the most typical human-object interaction such as playing violin, which corresponds to the functionality of violin.

To summary, this dissertation has studied the problem of understanding human actions in still images from a various of aspects. The research papers covered in this dissertation are among the first few publications that study this problem in the field of computer vision, and have inspired a number of work in the past few years. While there is still a long way to go towards understanding human actions in still images, our work also shed lights on many other research directions. We briefly discuss three of them in 8.2.

8.2 Future directions

8.2.1 Fine-grained recognition

As we have discussed in Chapter 3, action recognition can be regarded as a fine-grained recognition problem, since all the actions share the same image part – human. Therefore algorithms for action recognition can also be used for fine-grained image classification, and vice versa. In [138], a poselet [8] based approach has been applied to the task of classifying different bird species. We also proposed a codebook free and annotation free method [125] that can get rid of the key point annotations. How to further bridge the gap between action classification and other fine-grained recognition tasks is an interesting research direction.

8.2.2 Event classification in videos

Complex events, such as wedding ceremony and making a sandwich, are better described by video sequences. Since a video is composed of a number of frames, it is expected that event recognition in videos can benefit from recognizing human actions

in still images. In [87], a bags-of-features approach that treats each single frame separately has been applied to video event classification. Inspired by the performance improvement we obtained in PASCAL VOC 2012 compared with VOC 2011 (Chapter 3 and 4), in [111], we have shown that effectively combining multiple features can largely improve event recognition performance. However, video events are more complex than single frames, and therefore much more work needs to be done towards understanding the events in videos.

8.2.3 Social role understanding

In Chapter 5, 6, and 7, we have studied the interaction between humans and objects in still images. Another important interaction would be human-human interaction that happens frequently in both still images and video events. In [99], we proposed to recognize social roles from human event videos in a weakly supervised setting. This work enables us to automatically understand the relations between people, and discover the different roles associated with an event. Future research directions of this topic could be to perform joint event classification and social role understanding, and allow social roles to help improving the performance of human tracking.

Bibliography

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1994.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[4] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[5] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[6] I. Biederman, R. Mezzanotte, and J. Rabinowitz. Scene perception: detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14:143–177, 1982.

[7] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[8] L. Bourdev and J. Malik. Poselets: body part detectors trained using 3D human pose annotations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[9] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[10] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[11] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[12] L. Carlson-Radvansky, E. Covey, and K. Lattanzi. What effects on Where: Functional influence on spatial relations. *Psychological Science*, 10(6):519–521, 1999.

[13] Y. Chen, L. Zhu, C. Lin, A. Yuille, and H. Zhang. Rapid inference on a novel AND/OR graph for object detection, segmentation and parsing. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2007.

[14] C. A. Collin and P. A. McMullen. Subordinate-level categorization relies on high spatial frequencies to a greater degree than basic-level categorization. *Perception & Psychophysics*, 67(2):354–364, 2005.

[15] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[17] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[18] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: A study of bag-of-features and part-based representations. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[19] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2011.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[21] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[22] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *Proceedings of the Workshop on Structured Models in Computer Vision*, 2010.

[23] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000.

[24] S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert. An empirical study of context in object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[25] G. Duan, C. Huang, H. Ai, and S. Lao. Boosting associated pairing comparison features for pedestrian detection. In *Proceedings of the Workshop on Visual Surveillance*, 2009.

[26] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[27] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010.

[28] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results, 2011.

[29] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2012) Results, 2012.

[30] R.-E. Fan, K.-W. Chang, C.-J. Heish, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[31] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[32] A. Farhadi and A. Sadeghi. Recognition using visual phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[33] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proceedings of the Workshop on Generative Model Based Vision*, 2004.

[34] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. Short Course in the IEEE International Conference on Computer Vision, 2009.

[35] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[36] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[37] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminantly trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.

[38] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[39] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[40] V. Ferrari, M. Marín-Jiménez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[41] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single view geometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[42] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[43] E. Gibson. The concept of affordance in development: The renascence of functionalism. *The Concept of development: The Innesota Symposium on Child Psychology*, 15:55–81, 1982.

[44] J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[45] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[46] H. Grabner, J. Gall, and L. V. Gool. What makes a chair a chair? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[47] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[48] A. Gupta and L. Davis. Objects in action: an approach for combining action understanding and object perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[49] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[50] A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3D scene geometry to human workspace. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[51] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[52] G. Heitz and D. Koller. Learning spatial context: using stuff to find things. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[53] J. Henderson. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences*, 7(11):498–504, 2003.

[54] M. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[55] A. B. Hillel and D. Weinshall. Subordinate class recognition using relational object models. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2007.

[56] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[57] C. Huang, H. Ai, Y. Li, and S. Lao. High performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):671–686, 2007.

[58] M. Iacoboni and J. Mazziotta. Mirror neuron system: basic findings and clinical applications. *Annals of Neurology*, 62(3):213–218, 2007.

[59] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff. Learning actions from the web. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[60] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming, Series B*, 127:203–244, 2011.

[61] X. Jiang, Y. Yao, and L. Guibas. Stable identification of cliques with restricted sensing. In *Proceedings of the Workshop on Learning with Orderings*, 2009.

[62] K. E. Johnson and A. T. Eilers. Effects of knowledge and development on subordinate level categorization. *Cognitive Development*, 13(4):515–545, 1998.

[63] L. Karlinsky, M. Dinerstein, and S. Ullman. Unsupervised feature optimization (UFO): simultaneous selection of multiple features with their detection parameters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[64] J. Kim and I. Biederman. Where do objects become scenes? *Cerebral Cortex*, 21:1738–1746, 2010.

[65] H. Kjellstrom, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2010.

[66] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[67] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: object localization by efficient subwindow search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[68] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[69] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[70] I. Laptev and G. Mori. Statistical and structural recognition of human actions. Short Course of the European Conference on Computer Vision, 2010.

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

[72] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2007.

[73] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the Workshop on Statistical Learning in Computer Vision*, 2004.

[74] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2010.

[75] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[76] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[77] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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

[79] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[80] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[81] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[82] M. Meila and J. Shi. Learning segmentation by random walks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2000.

[83] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2007.

[84] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects, and scenes. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2003.

[85] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2003.

[86] J. C. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[87] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[88] L. Oakes and K. Madole. Function revisited: How infants construe functional features in their representation of objects. *Advances in Child Development Behavior*, 36:135–185, 2008.

[89] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions.

In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 1994.

[90] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the shape envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[91] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007.

[92] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[93] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[94] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):601–614, 2012.

[95] T. Quack, V. Ferrari, B. Leibe, and L. van Gool. Efficient mining of frequent and distinctive feature configurations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[96] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[97] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[98] D. Ramanan. Learning to parse images of articulated objects. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2006.

[99] V. Ramanathan, B. Yao, and L. Fei-Fei. Social role discovery in human events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[100] X. Ren, A. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[101] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[102] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[103] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[104] B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[105] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, 2004.

[106] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[107] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[108] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[109] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[110] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[111] K. Tang, B. Yao, L. Fei-Fei, and D. Koller. Combining the right features for complex event recognition. In *Submitted to the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[112] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[113] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[114] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[115] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[116] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

[117] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[118] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[119] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[120] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[121] Y. Wang and G. Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[122] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-201, Caltech, 2010.

[123] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[124] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures of parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[125] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[126] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[127] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[128] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5D graph matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[129] B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1691–1703, 2012.

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

[131] B. Yao, A. Khosla, and L. Fei-Fei. Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

[132] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[133] B. Yao, J. Ma, and L. Fei-Fei. Discovering object functionality. In *Submitted to the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[134] B. Yao, D. Walther, D. Beck, and L. Fei-Fei. Hierarchical mixture of classification experts uncovers interactions between brain regions. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2009.

[135] J. Yuan, J. Luo, and Y. Wu. Mining compositional features for boosting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[136] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation pattern: from visual words to visual phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[137] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[138] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[139] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[140] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[141] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.