ABSTRACT

Title of dissertation: BRIDGING THE SEMANTIC GAP : IMAGE AND VIDEO UNDERSTANDING BY EXPLOITING ATTRIBUTES

Xiaodong Yu, Doctor of Philosophy, 2013

Dissertation directed by: Professor Yiannis Aloimonos
Department of Electrical and Computer Engineering

Understanding image and video is one of the fundamental problems in the field of computer vision. Traditionally, the research in this area focused on extracting low level features from images and videos and learning classifiers to categorize these features to pre-defined classes of objects, scenes or activities. However, it is well known that there exists a "semantic gap" between low level features and high level semantic concepts, which greatly obstructs the progress of research on image and video understanding.

Our work departs from the traditional view of image and video understanding in that we add a middle layer between high level concepts and low level features, which is called as attribute, and use this layer to facilitate the description of concepts and detection of entities from images and videos. On one hand, attributes are relatively simple and thus can be more reliably detected from the low level features; on the other hand, we can exploit high level knowledge about the relationship between the attributes and the high level concepts and the relationship among at-

tributes, and therefore reduce the semantic gap. Our ideas are demonstrates in three applications as follows:

First, we presented an attribute-based learning approach for object recognition, where attributes are used to transfer knowledge on object properties from known classes to unknown classes and consequently reduce the number of training examples needed to learn the new object classes.

Next, we illustrate an active framework to recognize scenes based on the objects therein, which are considered as the attributes of the scenes. The active framework utilizes the correlation among objects in a scene and thus significantly reduces the number of objects to be detected in order to recognize the scene.

Finally, we propose a novel approach to detect the activity attributes from sports videos, where the contextual constraints are explored to decrease the ambiguity in attribute detection. The activity attributes enable us to go beyond naming the activity categories and achieve a fine-grained description of the activities in the videos.

BRIDGING THE SEMANTIC GAP : IMAGE AND VIDEO UNDERSTANDING BY EXPLOITING ATTRIBUTES

by

Xiaodong Yu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2013

Advisory Committee:
Professor Yiannis Aloimonos, Chair/Advisor
Professor John S. Baras
Professor Min Wu
Professor David Jacobs
Professor Gang Qu

Dedication

It is with a grateful heart that I dedicate my dissertation to my wife, Jing Zhang, to our children, Victoria and Henry, and to my parents, Xiquan Yu and Yumei Zhang, for all the unconditional love, guidance, and support that they have always given me.

Acknowledgments

I would like to thank my advisor Professor Yiannis Aloimonos who guided me throughout my Ph.D. study. This dissertation could not be made possible without his help. I am grateful as well to Dr. Cornelia Fermüller for supporting me through my Ph.D. study, which was one of the most important and inspiring experiences in my life.

The members of my dissertation committee, John S. Baras, Min Wu, David Jacobs and Gang Qu have generously given their time and expertise to better my work. I thank them for their contribution and comments to my dissertation.

Amongst my peers, I wish to thank Ching Lik Teo, Yezhou Yang, Zhe Lin, Guangyu Zhu and Yi Li, each of whom have been there for me not only as colleagues, but as close friends.

I am also in debt for Dr. Daniel DeMenthon for his inspirations and advices in many aspects of the computer vision domain.

After all I need to thank my wife, Jing Zhang, who is always there with me, for her constant love and unfailing support through my journey to Ph.D. I am also deeply grateful to our precious children Victoria and Henry, whose smiles and hugs have greeted me each day as I return home. Last, but not least, I would like to thank my parents who have raised me to be the person I am today.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement and Motivation

The problem studied in this thesis is image and video understanding. The goal of this research area is to produce descriptions for both the images/videos and the world scenes where the images/videos represent from the given images/videos [1]. Traditionally, research works in this area focus on feature extraction and object/scene/action classification and detection. Over the past decades, enormous research efforts have been directed at this field, and numerous approaches and systems have been proposed in literature. However, the state-of-the-art is still far from real image and video *understanding*. The most significant obstacle in this area is the so-called *semantic gap*. In literature [2], the semantic gap is defined as

> ... the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.

To fill this semantic gap, the machine must interpret the visual contents from many different aspects and on multiple granularity levels just as a human being does. Obviously, it will be very difficult, if not impossible, for traditional classification/detection paradigm to achieve this goal. In this thesis, we proposed a new

paradigm of understanding image and video towards filling the semantic gap. Comparing the the proposed paradigm in Figure 1.2 and the traditional paradigm in Figure 1.1, we can see a new layer is added between the high level entities and low level features, which is called *attributes* in this thesis. Instead of detecting/classifying objects/scenes/actions directly from low level features, we first detect attributes from the low level features and then use them to infer high level entities. Using attributes as a middle layer results the following key advantages:

The first advantage is that automatic learning and detection of attributes algorithms enables us to describe the entity beyond the basic category level [3] and to produce much richer perception. This fine-grained description is very important to understand the appearance of the interested entity and describe it to other people. Thus attributes complement traditional category-level recognition and improve the degree to which machines perceive visual world.

Figure 1.3 is a picture of a Rocky Mountain horse, which is downloaded from wikipedia [1] and credit to [4]. The authors' description of this picture is as follows:

"A silver colored Rocky Mountain Horse. The typical shiny white mane and tail as well as a slightly diluted body color with dapples is seen in this genetically black silver colored horse. The phenotype is caused by dilution of eumelanin in the hair to white or grey. The dilution is most visible in the long hairs of the mane and tail. The horse has also been diagnosed with MCOA."

With traditional paradigm, the machine can only say that there is a *horse* in the picture; while using the proposed approach, the machine can produce a more

[1] `http://en.wikipedia.org/wiki/Rocky_Mountain_Horse`

desirable description as follows:

"a horse with black color and white dots, white mane and white tail, running on grassland and in woods".

Obviously, the later description is much closer to what a human being can perceive from that image.

This advantage can be further illustrated in the language domain. In the traditional paradigm (Figure 1.1), the high level entities can only be represented with nouns and verbs; while in the proposed paradigm (Figure 1.2), the middle level attributes can be represented as nouns, verbs, adjectives, adverbs, prepositions and conjunctions. This leads to much richer descriptions of the visual world and thus greatly reduces the semantic gap.

The second advantage is that we can achieve considerable economy of representation, because complex high level entities (objects, scenes, actions) can be described by a relatively small set of attributes. Suppose a basic level object category with 20 attributes, there are $\binom{20}{5} = 15504$ possible combinations of 5 attributes. If we consider each combination of 5 attributes as an object subcategory, in traditional paradigm, we need to train 15504 object classifiers; while in the proposed paradigm, we only need to train 20 attribute classifiers. The attribute layer provides an information bottleneck between the low level features and the high level concepts, and thus greatly reduce the computational complexity.

The third advantage is that attributes are considerably general across entity categories. Therefore, we can transfer our knowledge of attribute across different basic level entity categories. This advantage greatly extend the generality of the

Figure 1.1: The traditional paradigm for image and and video understanding.

state-of-the-art approaches for image and video understanding, which can never be achieved by the traditional paradigm.

1.2 Related Work

1.2.1 Attributes for Objects, Scenes and Activities

Object attributes have been studied from multiple perspectives in the last decade: [5, 6, 7] described objects and human with various attributes; [8] used attributes to report unusual aspects of a familiar objects, or say something about unfamiliar objects; [9, 10, 11, 12, 13] used attribute detection as a component in object detection and recognition, and showed improved performance on both tasks; [14] presented methods to transfer knowledge via object attribute and facilitate the task of *zero-shot* learning. In this thesis, we will propose a new approach of transfer learning via object attributes for both the tasks of *zero-shot* learning and

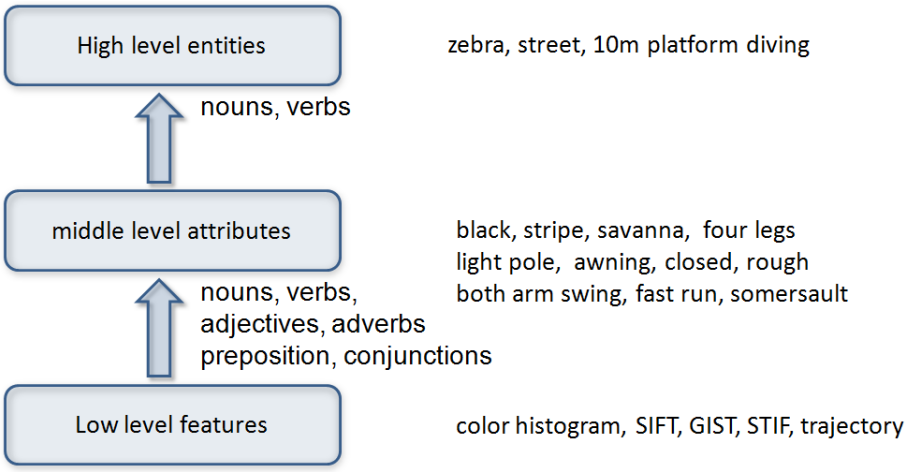


Figure 1.2: The proposed paradigm for image and and video understanding.

Figure 1.3: A picture of Rocky Mountain horse.

one-shot learning. Our main contribution is in two new methods for combining prior knowledge in object attributes and the observations of the new object class to learn a new object classifier.

Oliva and Torralba[15] first proposed a computational model for scene recognition based on some high-level property of the scene, such as openness, naturalness, etc, which is called the spatial envelope, a special set of scene attributes. Recently, Li et al proposed to use objects as scene attributes and recognize scene categories by the outputs of a set of object classifiers. The idea of treating objects as scene attributes have been also explored in [16, 17]. A more comprehensive study on scene attributes are described in [18] with a large-scale dataset. In this thesis, we will study an active approach to recognize scenes by its attributes so that we can achieve great efficiency compared to the traditional passive approaches.

Action attributes share lots of properties with object and scene attributes. In literature, [19] first introduce the idea of activity attributes. But their approach has fundamental limitations, which restrict its application in real world problems. In this thesis, we will address these limitations by introducing a new graphical model to model the temporal and semantic correlation among activity attributes.

1.2.2 Comparison to Biederman's Recognition-by-Components Theory

The recognition-by-components theory by Biederman [20] was proposed to recognize object by detecting objects main component parts, which is called *geons*.

Biederman suggested that geons are based on basic 3-dimensional shapes (cylinders, cones, etc.) that can be assembled in various arrangements to form a virtually unlimited amount of objects. Indeed, geons and attributes share similarities in many aspects: they are all middle layer between high level entity and low level features; they are all general across different entities. However, there exists significant difference between geons and attributes. While geons only represent the shape properties of an object, attributes represent much broader properties of objects, scenes and activities. For objects, attributes can be color, shape, textures, parts, environments, etc. More important, all attributes contain semantic meanings and can be presented by language. Thus, not only attributes can be used to facilitate recognizing entities, but also they can be used to *describe* entities. Finally, the prior knowledge stored in language domain can be used to simplify the recognition tasks in vision domain, which is not available in Biederman's approach.

1.3 Assumptions in the Approaches

In this thesis, we assume the relationship between the entities (object, scenes, activities) and their attributes are known in prior. This prior knowledge can be either obtained from annotations in the datasets or the knowledge database constructed by experts. Our focus is to show how these prior knowledge can be exploited in facilitating to solve various challenging vision problems.

1.4 Contributions of This Thesis

In this thesis, we study three applications of attributes, namely, describing entities, recognizing entities, and transfer learning, in three domains: objects, scene, and activities. We specially consider the following three problems:

- transfer learning by object attributes for one/zero-shot learning;
- recognize scenes by actively detecting scene attributes; and
- detecting activity attributes for describing sports videos.

For each of these problems, we provide both background on the problem and a discussion of previous work in the respective domain and other related research. We then describe our solution, detailing the novel characteristics of our proposed approach and validating with experimental evaluations.

1.4.1 Transfer Learning by Object Attributes

We first consider the problem of transfer learning by object attributes. In traditional learning paradigm, we need to train classifiers for each category separately; while in transfer learning paradigm, we can use knowledge learned from some categories to facilitate learning in another category. Thus we can learn classifiers for new categories with much fewer training examples or even no training examples.

Due to the generality of attributes across different categories, attributes are especially suitable for transfer learning problems. For example, after training an attribute detector for "furry" from images of *bear*, we can use it to filter out images

that do not contain "furry" attributes to facilitate the detection of *cheetah*, as we know both animals have attribute of "furry".

In this work, we designed a generative attribute model that offer flexible representations for attribute knowledge transfer. Also we proposed two methods that effectively employ attribute prior in the learning of target classifiers and combine the training examples of target categories when they are available. Thus the attribute prior can help improving performance in both zero-shot and one-shot learning task. At the end, we evaluated the performance of our transfer learning system on the popular Animal with Attributes [14] data set.

1.4.2 Recognize Scenes by Detecting Scene Attributes

The second problem we considered in this thesis is to recognize scene by actively detect scene attributes. In this problem, we consider objects as scene attributes. But the proposed approach can be easily extend to many other scene attributes, such as the spatial structure of the scene [15]. Similar to the idea of recognizing object by components, many people have studied the problem of recognizing scene by attributes, e.g., objects in the scene. However, most of existing approaches are passive and need to run through all object/attribute detectors over the testing images before making the final conclusion. In this work we explore an active approach, which aims at greatly reducing the number of object/attribute detectors needed for recognition of scenes.

Human beings are active explorers. We continuously shift our gaze to different

locations in the scene and focus our attention on the objects that are most important to the current situation and tasks. After recognizing objects, we will fixate again at a new location, and so on. By optimizing the locations and objects to recognize, we can quickly understand the scene and make prompt reactions. It is clear that when we analyze a complex scene, visual processes continuously interact with our high-level knowledge, some of which is represented in the form of language. In some sense, perception and language are engaged in an interaction, as they exchange information that leads to meaning and understanding.

In this work, we proposed an active vision for scene recognition, which consists of a reasoning module and a sensory module. The sensory module is object detector, which extracts features from images, detect and localize objects. The reasoning module obtains higher level knowledge about scene and object relations, proposes attentional instructions to the sensory module and draws conclusions about the contents of the scene. The key difference between the proposed active approach and the traditional passive approaches is that the sensory module does not passively process the image; instead, it is guided by the reasoning module, which decides what and where the sensory module should process next. Thus the sensory module shifts the focus of attention to a small number of objects at selected locations of the scene. Thus the machine can recognize the scene with faster speed and higher accuracy. This approach is also extended to a hierarchical active scheme to recognize dynamic scenes.

1.4.3 Detecting Activity Attributes for Describing Sports Videos

In the last problem, we study how to detect activity attributes from videos; in particular, we focus on the action attributes related to motion patterns of human body; nevertheless, our model can be easily extended to detect the other types of action attributes as well.

The concept of action attribute was first introduced in [19]. However, the approach proposed in [19] has several severe limitations that restrict the applicability of action attributes. One of the most noticeable problems is that action attributes are labeled at the level of an action class, instead of being labeled at the level of a frame or at the level of a video. As a result, all videos from the same action class are assumed to have the same set of action attributes, regardless of the exact content of a specific video, and the temporal structure of the action attributes is totally discarded. But in reality, not every video belonging to the same action class have the same set of action attributes; more often than not, real-world videos would have some exceptions. For example, in a video of the `snatch` activity, the athlete may not be able to completely lift the barbell above his head at the end so we cannot say this video has an action attribute *two arms raise pose*. Furthermore, the temporal structure is a unique property of action attributes and we lose lots of descriptive capacity if we ignore it. For example, given a video of `basketball layup`, a description *"the athlete starts with a slow run and lasts for half a second, then jumps forward with single leg in the next second, finally jumps up and throws the ball (into the basket), and maintains a slow run at the end of the video"* will be more useful

than simply saying *"there are slow running, jumping forward, jumping up, throwing in this video"*. The goal of this work is thus to detect the key action attributes at each frame from a given video so that we can generate video descriptions at a much finer granularity than those from previous work.

1.5 Organization of Thesis

The rest of this thesis is organized as follows: in the next chapter, we introduce the approach of transfer learning by object attributes; then we describe the idea of active scene recognition with scene attributes in Chapter 3; our approach for detecting activity attributes is presented in Chapter 4; Chapter 5, the conclusion, both provides a summary of the contributions contained in this work and suggests some research directions for future investigation.

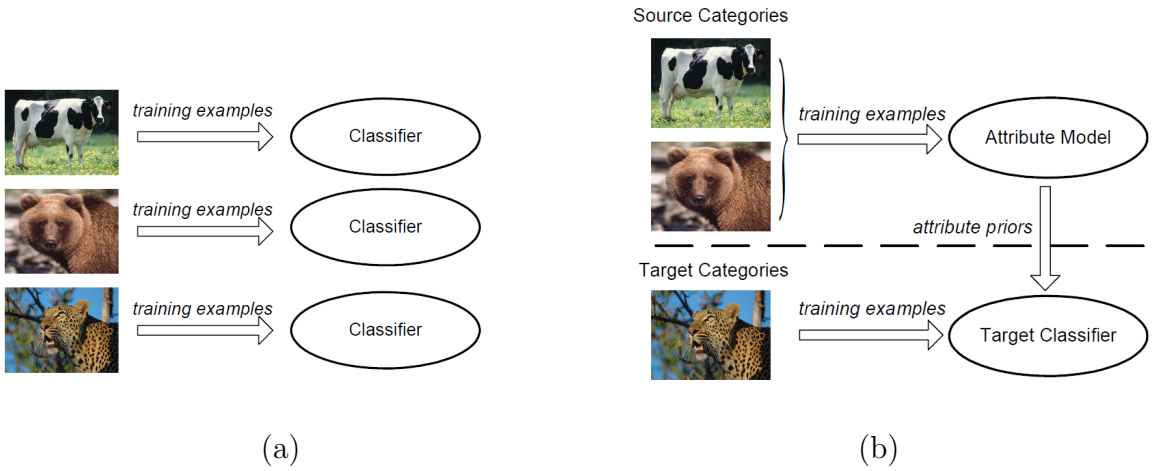
Chapter 2

Attribute-Based Transfer Learning for Object Recognition

2.1 Introduction

In this chapter we present an attribute-based transfer learning approach for object recognition with zero or one training examples. Object attributes are high-level descriptions about properties of object categories such as color, texture, shape, parts, context, etc. Human beings have a remarkable capability in recognizing unseen objects purely based on object attributes. For example, people who have never seen a zebra still could reliably identify an image of zebra if we tell them that "a zebra is a wild quadrupedal with distinctive white and black strips living on African savannas". As long as they have prior knowledge about the related object attributes, e.g., *quadrupedal*, *white and black strips*, *African savannas*, they can transfer these knowledge to facilitate prediction of unseen categories. The attribute-based transfer learning framework is motivated by this insight.

Figure 2.1 compares the different learning processes of conventional learning approaches and attribute-based transfer learning approaches: while conventional learning approaches treat each category individually and train each classifier from scratch, the attribute-based transfer learning approaches can help improve the learning of classifiers for the target categories (unknown categories) using the attribute prior knowledge learned from source categories (i.e., known categories). Therefore,



(a) (b)

Figure 2.1: Diagrams of the learning process between conventional learning approaches (a) and attribute-based transfer learning approaches (b).

we are able to learn target classifiers with much fewer training examples, or even no examples. This is the major obstacles that we need to address in the problem of one-shot learning [21] and zero-shot learning [14]. Under these circumstances, conventional learning methods can not function due to the lack of training examples. To solve this problem, knowledge transfer becomes extremely important [22]: by transferring prior knowledge obtained from *source categories* (i.e. known categories) to *target categories* (i.e. unknown categories), we equivalently increase the number of training examples of the target categories. Thus, the difficulties raised by the scarcity of training examples can be greatly alleviated. In the proposed approach, object attributes are used to transfer knowledge from known categories to unseen categories.

2.2 Related Work

Roughly, the methods of knowledge transfer for object categorization can be divided into three groups [22]: knowledge transfer by sharing either *features* [23, 24], *model parameters* [21, 25] or *context information* [26]. Most of the early work relies on bootstrap approaches to select features or parameters to be transferred [23, 24, 21]. A very recent study [25] suggests that an explicit and controllable transfer of prior knowledge can be achieved by considering the ontological knowledge of object similarity. For example, *horse* and *giraffe* are both quadrupeds and share common topologies, so a full model can be transferred from horse to giraffe. The work presented in this paper integrates a broader ontological knowledge, i.e., object attributes, which can transfer knowledge either among similar categories (e.g., horse and giraffe), or among different categories that share common attributes(e.g., both German shepherd and seal have the attribute *black*).

Several recent studies have investigated the approach employing the object attributes in recognition problems [14, 8, 10, 27]. Among them, our work is most related to [14, 27]. However, as both studies focused on attribute prediction for zero-shot learning task, they did not attempt to combine attribute priors with the training examples of target categories. Thus, although useful, their applications in one-shot learning task are still limited. Since the framework presented in this paper includes the routes for both attribute priors and the training examples of target categories, we can benefit from these two domains whichever is available in learning a new target category. Compared to the existing work in [14, 27], our contribution is

a more complete framework for attribute-based transfer learning, which enables us to handle both zero-shot learning and one-shot learning problems. The approaches in [8, 10] are also related to ours. However, their methods need attributes annotated for each image. Although the image-level attribute annotation will facilitate intra-class feature selection [8] and object localization [10], it requires substantially more human efforts to label each image. Thus their scalability to a large number of categories is greatly restricted compared to the category-level attribute annotations advocated in [14, 27] and this paper.

2.3 Algorithms

2.3.1 Background

In the proposed approaches, the category-attribute relationship is represented by a category-attribute matrix $\mathcal{M}$ for a particular image data set, where the entry at the m-th row and the ℓ-th column is a binary value indicating whether category m has the ℓ-th attribute. Figure 2.2.a illustrates an example of $\mathcal{M}$. Each object category thus has a list of attributes whose corresponding values in $\mathcal{M}$ equal to "yes". This information is supposed to be available for both source categories and target categories. Currently, this information is obtained from user labelled dataset.

In our approach, the attribute model and the target classifier belong to an extension of topic models, which constitute an active research area in the machine learning community in recent years [28, 29, 30]. Computer vision researchers have extended them to deal with various vision problems [31, 32, 33, 34, 35, 36]. In a

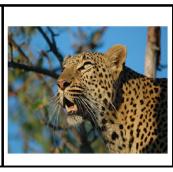
			
black	yes	yes	yes
white	no	no	yes
brown	yes	yes	yes
spots	yes	no	yes
furry	yes	yes	yes
meat	yes	yes	no
domestic	no	no	yes

Figure 2.2: Examples of ontological knowledge represented by the binary category-attribute values. Images and attributes are from the "Animals with Attributes" data set [14].

topic model, a document $\mathbf{w}$ is modeled by a mixture of topics, z's, and each topic z is represented by a probability distribution of words, w's. In the computer vision domain, a quantized image feature is often analogous to a word (a.k.a "visual words" [31]), a group of co-occurred image features to a topic (a.k.a "theme" [34]), and an image to a document. In Section 2.4, we will visualize visual words and topics using examples in the test data set.

2.3.2 The Intuition of Attribute-based Transfer Learning

The idea of transfer learning is to "transfer" some prior knowledge from the learned classes (i.e., source categories) to a new class (target categories) to facilitate the learning of the new class. Depending on the image model and application domain, there are many types of prior knowledge and methods to transfer them

(see [22] for a survey). In this work, we use the parameters in the image model to transfer the knowledge about attributes. From source categories *cow, zebra, deer, cheetah*, we learn the low-level feature distribution for attribute *black, white, bush, stripes, patches*, $F_a(\mathbf{w};\phi_a)$, where $\phi_{1,\ldots,k}$ are parameters for distribution $F_a(\cdot)$. An object class c can then be represented by a distribution of attributes, $G_c(\mathbf{a}_c;\theta_c)$, where θ_c are parameters for distribution $G_c(\cdot)$ and $\mathbf{a}_c$ are the positive attributes for class c. Overall, the distribution of low-level features for class c can be written as $\sum_{a\in\mathbf{a}_c} G_c(\mathbf{a}_c;\theta_c) \times F_a(\mathbf{w};\phi_a)$.

Given the attributes, the task to train a classifier for class c is to learn the parameter θ_c and ϕ_a from the training images. Since we have to prior knowledge about attributes, i.e., the distribution $F_a(\mathbf{w};\phi_a)$ is known, we only need to estimate θ_c from the training images. Therefore, the training process is greatly simplified and we can train the classifier for class c with less training examples or even just one example. If there is no training images but we know the positive attributes for the target class, we can treat $G_c(\cdot)$ as a uniform distribution among all positive attributes and the the distribution of low-level features for class c becomes $\sum_{a\in\mathbf{a}_c} \frac{1}{|\mathbf{a}_c|} F_a(\mathbf{w};\phi_a)$. Note that this is a non-trivial distribution since $F_a(\mathbf{w};\phi_a)$ incorporate our prior knowledge about the attributes learned from source categories. The knowledge transfer is achieved with the help of known positive attributes for class c, $\mathbf{a}_c$. Otherwise, the best we can do is to randomly select some attributes and we will show that this strategy can only produce classification accuracy of chance level.

2.3.3 Attribute Model and Object Classifier

The attribute-based transfer learning approach needs an attribute model and an object classifier. The attribute model we employed is the Author-Topic (AT) model [30]. The AT model is originally designed to model the interests of authors in a given document corpus. In this paper, we extend the AT model to model the distribution of image features related to attributes. To our best knowledge, this is the first attempt of this kind. Indeed, authors of a document and attributes of an object category have many similarities, which allow us to analogize the latter to the former: a document can have multiple authors and an object category can have multiple attributes; an author can write multiple documents and an attribute can be presented in multiple object categories. Nevertheless, there is also noticeable difference between them: each document can have a distinct list of authors, while all images within an object category share a common list of attributes.

The graphic representation of AT model is illustrated in Figure 2.3.a. The AT model is a generative model. In this model, an image j has a list of attributes, denoted by $\mathbf{a}_j$. An attribute ℓ in $\mathbf{a}_j$ is modeled by a discrete distribution of K topics, which parameterized by a K-dim vector $\theta_\ell = (\theta_{\ell 1}, ..., \theta_{\ell K})$ with topic k receiving weight $\theta_{\ell k}$. The topic k is modeled by a discrete distribution of W codewords in the lexicon, which is parameterized by a W-dim vector $\phi_k = (\phi_{k1}, ..., \phi_{kW})$ with codeword v receiving weight ϕ_{kv}. Symmetric Dirichlet priors are placed on θ and ϕ, with $\theta_\ell \sim Dirichlet(\alpha)$, and $\phi_k \sim Dirichlet(\lambda)$, where α and λ are hyperparameters that affect the sparsity of these distributions. The generative process is outlined in

Algorithm 1.

Algorithm 1 The generative process of the Author-Topic model

1: given the attribute list $\mathbf{a}_j$ and the desired number of visual words in image j, N_j

2: **for** $i = 1$ to N_j **do**

3: conditioning on $\mathbf{a}_j$, choose an attribute $x_{ji} \sim \text{Uniform}(\mathbf{a}_j)$

4: conditioning on x_{ji}, choose a topic $z_{ji} \sim \text{Discrete}(\theta_{x_{ji}})$

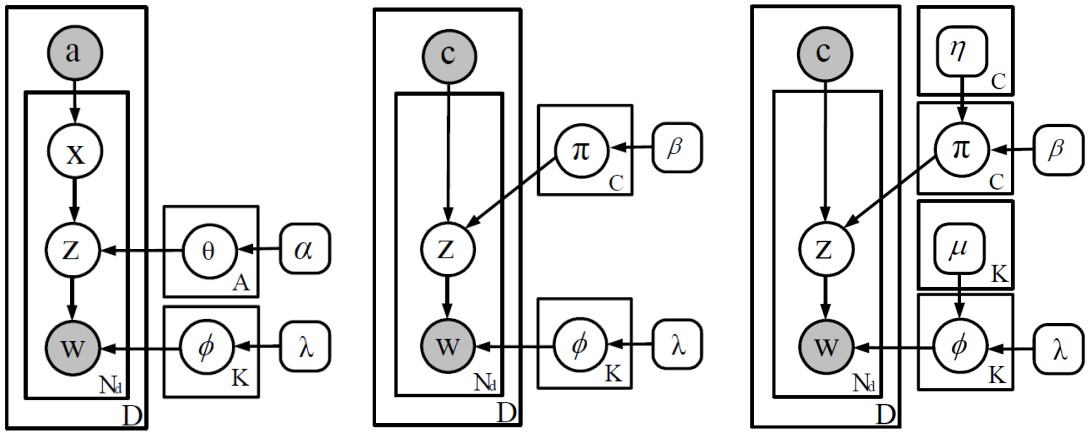
5: conditioning on z_{ji}, choose a visual word $w_{ji} \sim \text{Discrete}(\phi_{z_{ji}})$

6: **end for**

If there is only one attribute in each image and the attribute is the object category label, the AT model can be used in object categorization problems [33]. In our work, we call this approach Category-Topic (CT) model and use it as the target classifier in the proposed transfer learning framework. The graphic representation of CT model is illustrated in Figure 2.3.b and Figure 2.3.c. In Figure 2.3.b, the CT model uses non-informative prior, which is employed in previous work [33]. The CT model we employed in this work has informative prior, as illustrated in Figure 2.3.c. We will show how to use this special structure to transfer the prior knowledge about attributes later.

2.3.4 Inference and Parameter Estimation

Given a training corpus, the goal of inference in an AT model is to identify the values of ϕ and θ. In [30], Rosen-Zvi et al. presented a collapsed block Gibbs sampling method. The term "collapse" means that the parameters ϕ and θ are



(a) (b) (c)

Figure 2.3: Graphical representations of the Author-Topic (AT) model (a), the Category-Topic (CT) model (b) and the CT model with informative Dirichlet priors over π and ϕ (c). See text for detailed discussions of these models.

analytically integrated out, and the term "block" means that we draw the pair of (x_{ji}, z_{ji}) together. The pair of (x_{ji}, z_{ji}) is drawn according to the following conditional distribution

$$p(x_{ji} = \ell, z_{ji} = k | w_{ji} = v, \Omega) \propto \frac{\alpha/K + N^k_{\ell,\backslash ji}}{\alpha + \sum_{k'=1}^{K} N^{k'}_{\ell,\backslash ji}} \frac{\lambda/W + C^v_{k,\backslash ji}}{\lambda + \sum_{v'=1}^{W} C^{v'}_{k,\backslash ji}}, \tag{2.1}$$

where $\Omega \equiv \{\mathbf{a}_j, \mathbf{z}_{\backslash ji}, \mathbf{x}_{\backslash ji}, \mathbf{w}_{\backslash ji}, \alpha, \lambda\}$, the subscript ji represents the i-th visual word in image j, $x_{ji} = \ell$ and $z_{ji} = k$ represent the assignments of current visual word to attribute ℓ and topic k respectively, $w_{ji} = v$ represents the observation that the current visual word is the v-th codeword in the lexicon, $\mathbf{z}_{\backslash ji}$ and $\mathbf{x}_{\backslash ji}$ represent all topic and attribute assignments in the training corpus excluding the current visual word, $N^k_{\ell,\backslash ji}$ is the total number of visual words that are assigned to attribute ℓ and topic k, excluding w_{ji}, and $C^v_{k,\backslash ji}$ is the total number of visual words with value v

that are assigned to topic k, excluding w_{ji}.

To run the Gibbs sampling algorithm, we first initialize $\mathbf{x}$ and $\mathbf{z}$ with random assignments. In each Gibbs sampling iteration, we draw samples of x_{ji} and z_{ji} for all visual words in the training corpus according to the distribution in Equation (2.1) in a randomly permuted order of i and j. The samples of $\mathbf{x}$ and $\mathbf{z}$ are recorded after the burn-in period. In experiments, we observe 200 iterations are sufficient for the sampler to be stable. The posterior means of θ and ϕ can then be estimated using the recorded samples as follows:

$$\hat{\theta}_{\ell k} = \frac{\alpha/K + N_{\ell}^{k}}{\alpha + \sum_{k'=1}^{K} N_{\ell}^{k'}},\ \hat{\phi}_{kv} = \frac{\lambda/W + C_{k}^{v}}{\lambda + \sum_{v'=1}^{W} C_{k}^{v'}}, \tag{2.2}$$

where N_{ℓ}^{k} and C_{k}^{v} are defined in a similar fashion as in Equation (2.1), but without excluding the instance indexed by ji.

The inference of a CT model can be performed in a similar way to the AT model. In the Gibbs sampling, we draw samples z_{ji} according to the following conditional distribution

$$p(z_{ji} = k | w_{ji} = v, c_j = m, \Omega) \propto \frac{\beta/K + M_{m,\backslash ji}^{k}}{\beta + \sum_{k'=1}^{K} M_{m,\backslash ji}^{k'}} \frac{\lambda/W + C_{k,\backslash ji}^{v}}{\lambda + \sum_{v'=1}^{W} C_{k,\backslash ji}^{v'}}, \tag{2.3}$$

where $\Omega \equiv \{\mathbf{z}_{\backslash ji}, \mathbf{w}_{\backslash ji}, \beta, \lambda\}$, $M_{m,\backslash ji}^{k}$ is the number of visual words in images of category m assigned to topic k, excluding the current instance. The posterior mean of π can be estimated as follows:

$$\hat{\pi}_{mk} = \frac{\beta/K + M_{m}^{k}}{\beta + \sum_{k'=1}^{K} M_{m}^{k'}}, \tag{2.4}$$

and the posterior mean of ϕ is the same as in Equation (2.2).

After learning a CT model, we can use it to classify a test image $\mathbf{w}_t = \{w_{t1}, ..., w_{tN_t}\}$ by choosing the target classifier that yields the highest likelihood, where the likelihood for category $c = m$ is estimated as

$$p(\mathbf{w}_t|c = m, \mathcal{D}^{train}) \approx \prod_{i=1}^{N_t} \sum_{k=1}^{K} \hat{\phi}_{kw_{ti}} \hat{\pi}_{mk}. \tag{2.5}$$

If the attribute list is unique in each category, an AT model can also be used to classify a new image by the maximum likelihood criterion. Suppose we have learned θ_ℓ for every $\ell = 1, ..., A$ from the source categories, we can then use them in classifying an image of a target category using the approximate likelihood

$$p(\mathbf{w}_t|c = m, a_m, \mathcal{D}^{train}) \approx \prod_{i=1}^{N_t} \sum_{k=1}^{K} \hat{\phi}_{kw_{ti}} \left(\frac{1}{A_m} \sum_{\ell \in a_m} \hat{\theta}_{\ell k} \right) \equiv \prod_{i=1}^{N_t} \sum_{k=1}^{K} \hat{\phi}_{kw_{ti}} \tilde{\pi}_{mk}, \tag{2.6}$$

where $\mathbf{a}_m$ is the attribute list associated to a target category $c = m$, A_m the length of $\mathbf{a}_m$. In the above equations, we have constructed a pseudo weight for the category-specified topic distribution of a new category from $\hat{\theta}_\ell$, i.e., $\tilde{\pi}_{mk} \equiv \left(\frac{1}{A_m} \sum_{\ell \in a_m} \hat{\theta}_{\ell k} \right)$. This pseudo weight can be viewed as the prior of π_m before we see the real training examples of the new category. Although the unique-attribute-list assumption does not hold in general, it is necessary for attribute-only classifiers, including the AT model discussed in this paper and the approaches in [14, 8], to predict unseen categories. The data set tested in this paper satisfies this assumption.

While the AT model can be used to deal with the zero-shot learning problem, it is ineffective for the one-shot learning problem. One may conjecture to add the training examples of target categories to those of source categories and then re-train the AT model. However, this naive approach will not work well in practice because the number of training examples of source categories is usually higher than

that of target categories by several orders. Consequently the AT model can not well represent the new observations of target categories in the training examples. Thus we need approaches to control the balance between the prior information from source categories and the new information in target categories. We will propose two approaches to achieve this goal in the rest of this section.

2.3.5 Methods for Knowledge Transfer

2.3.5.1 Knowledge Transfer by Informative Parameter Prior

The first knowledge transfer approach is to synthesize training examples for target categories. The idea is as follows: first, we learn the attribute model from the training examples of the source categories; second, for each target category, we run the generative process in Algorithm 1 to produce S synthesized training examples using the estimated $\hat{\theta}$ and $\hat{\phi}$ as well as the attribute list associated to this target category. Each synthesized training example contains $\bar{N}$ visual words, where $\bar{N}$ is the mean number of visual words per image in the source categories. In this procedure, S represents our confidence about the attribute priors. We can use it to adjust the balance between the attribute priors and new observations from the training images of target categories.

Since we adopt the bag-of-features representation, the synthesized example is actually composed of a set of image features without spatial information. So they are indeed "artificial" examples in that we can not visualize them like a real image. This is different from the image synthesis approaches in the literature [37, 38], which

output viewable images. Nevertheless, since our goal is to generate training examples for the target categories to assist the learning process, this is not an issue providing the classifiers take these bag-of-features as inputs.

2.3.5.2 Knowledge Transfer by Synthesis of Training Examples

The second knowledge transfer approach is to give parameters of the CT model in the target classifiers informative priors. Figure 2.3.c illustrates the complete CT model, where π and ϕ are given Dirichlet distributions as priors. In these Dirichlet distributions, μ and η are base measurements that represent the mean of ϕ and π, and λ and β are scaling parameters that control the sparsity of the samples drawn from the Dirichlet distribution. When we have no clue about the prior of ϕ and π, we usually give symmetric Dirichelt priors, whose base measures are uniform distributions. The graphical representations of CT models often neglect such uniform distributed base measures and only retain the scaling parameters λ and β, as shown in Figure 2.3.b. This rule also applies to the AT model. In this paper, these scaling parameters are given vague values when performaning Gibbs sampling, $\lambda = W$, $\alpha = \beta = K$.

However, after we learn the attribute model from source categories, our uncertainty about the ϕ and π of target categories will be greatly reduced. Our knowledge on these parameters are represented by the estimated $\hat{\phi}$ in Equation (2.2) and $\tilde{\pi}$ in Equation (2.6). Since $E(\phi_k) = \mu_k$ and $E(\pi_m) = \eta_m$, now we can give informative priors to ϕ and π by setting $\mu_k = \hat{\phi}_k$ and $\eta_m = \tilde{\pi}_m$. The basic equation of Gibbs

sampling of the CT model with informative prior the becomes

$$p(z_{ji} = k|w_{ji} = v, \Omega) \propto \frac{\beta\tilde{\pi}_{mk} + M^k_{m,\backslash ji}}{\beta + \sum_{k'=1}^{K} M^{k'}_{m,\backslash ji}} \frac{\lambda\hat{\phi}_{kv} + C^v_{k,\backslash ji}}{\lambda + \sum_{v'=1}^{W} C^{v'}_{k,\backslash ji}}, \tag{2.7}$$

where $\Omega \equiv \{c_j = m, \mathbf{z}_{\backslash ji}, \mathbf{w}_{\backslash ji}, \beta\eta, \lambda\mu\}$. The posterior means of π and ϕ in Equation (2.4) and (2.2) are updated accordingly. The values of λ and β represent our confidence on these priors, which can be used to control the balance between attribute priors and the new observations of target categories. In the experiments, we set $\lambda = \beta = \bar{N}S$, where $\bar{N}$ and S are defined as in Section 2.3.5.2.

By comparing Equation (2.7) and Equation (2.3), we can appreciate the importance of informative priors for the zero-shot learning task. If we have no prior knowledge about π, we can only give it a symmetric Dirichlet prior where $\eta_{mk} = 1/K$. In this scenario, the CT model have to see some training examples of target categories; otherwise, π_{mk} will be assigned to vague value $1/K$, which is useless for categorization tasks. Thus the CT model can not be used in zero-shot learning task. With the attribute knowledge, we can give π informative priors $\eta_{mk} = \tilde{\pi}_{mk}$, which permits us to perform zero-shot learning task using the CT model. Similar impact of the informative priors can be observed in the one-shot learning task.

2.4 Experiments

2.4.1 Data Set and Image Features

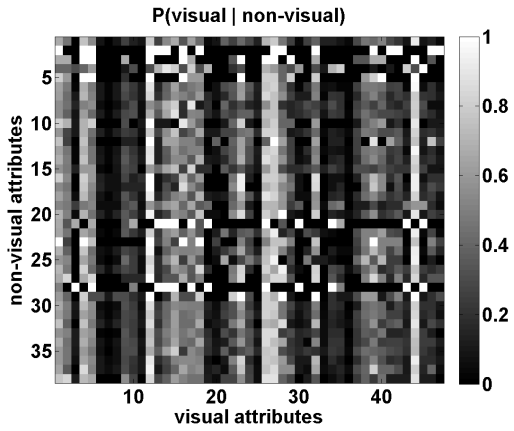
In the experiments, we use the "Animals with Attributes" (AwA) data set described in [14]. This data set includes 30475 images from 50 animal categories,

and 85 attributes to describe these categories. The category-attribute relationship is labeled by human subjects and presented in a 50×85 matrix $\mathcal{M}$. Figure 2.4.a illustrates a subset of this matrix. 40 categories are selected as source categories and the rest 10 categories are used as target categories. The division of source and target categories is the same as in [14]. The 85 attributes can be informally divided into two groups: visual attributes such as *black*, *furry*, *big*, *arctic*, etc., and non-visual attributes such as *fast*, *weak*, *fierce*, *domestic*, etc. Totally there are 38 non-visual attributes (attribute No.34 to No.64 and attribute No.79 to No.85) and 47 visual attributes. While non-visual attributes are not directly linked to visual features, it turns out that the non-visual attributes have strong correlation to the visual attributes, as shown in Figure 2.4.b. Take the attribute *fast* as an example, the top three most related visual attributes are *furry* ($P(furry|fast) = 0.833$), *tail* ($P(tail|fast) = 0.833$) and *ground* ($P(ground|fast) = 0.786$).

All images are resized such that the longest side has 300 pixels. From each image, we extract four types of image features: SIFT [39], rgSIFT [40], local color histogram and local self-similarity histogram (LSS) [41] at regular grids on two scales. Then for each type of feature, we build a visual lexicon of size 1000 by applying K-means clustering algorithms over features from 250 images randomly selected from source categories. Codewords from four type of features are combined into a single lexicon with 4000 codewords. Features in all images are quantized into one of the codewords in this lexicon. On average, there are about 5000 features in each image. So we set $\bar{N} = 5000$ in the approaches of attribute knowledge transfer in Section 2.3.5.2 and Section 2.3.5.1.

			
black	yes	yes	yes
white	no	no	yes
brown	yes	yes	yes
spots	yes	no	yes
furry	yes	yes	yes
meat	yes	yes	no
domestic	no	no	yes

(a)



(b)

Figure 2.4: (a): examples of ontological knowledge represented by the binary category-attribute values; (b): the empirical probability of nonvisual attributes conditioned on visual attributes measured by $P(visual|non-visual) \equiv N(visual, non-visual)/N(non-visual)$, where $N(\cdot)$ denote the number of categories that have the particular attributes in the given data set. Images and attributes are from the "Animals with Attributes" data set [14].

2.4.2 Experiment Setup and Implementation Details

Baseline Algorithms. In the experiments, we use Direct Attribute Prediction (DAP) [14] and SVM as baselines in the zero/one-shot learning tasks.

The DAP is selected as a baseline because it is the state-of-the-art approach for zero-shot learning on the AwA data set. DAP uses a SVM classifier that is trained from source categories to predict the presence of each attribute in the images of target categories. Then the attribute predictions are combined into a category label prediction in an MAP formulation. The original DAP can only perform zero-shot

learning. For one-shot learning, we use predicted attributes as features and choose a 1NN classifier following the idea in [8]. We call this classifier as "DAP+NN" in this paper.

When we use the synthesized training examples to transfer attribute knowledge, many existing classifiers can be used as the target classifier. We choose SVM as a baseline in this case, mainly because SVM is one of the state-of-the-art classifiers with bag-of-features image representation [42].

Implementation Details. The AT model has $K_0 = 10$ unshared topics per attribute in all tests. When using synthesized training examples, the CT model has 100 topics; when using informative priors, the number of topics in the CT model is the same as the total number of topics in the AT model. The SVM in the target classifiers is implemented using the C-SVC in LIBSVM with a χ^2 kernel. The kernel bandwidth and the parameter C are obtained by cross-validation on a subset of the source categories.

Evaluation Methodology. In the zero-shot learning scenario, both AT and DAP are trained using the first 100 images of each source category. Then we use the AT model to generate $S = \{10, 20, 100\}$ synthesized examples for each target category. The CT and SVM classifiers will be trained using these synthesized examples. We denote them as "CT+S" and "SVM+S" respectively in the reported results. Also we use the learned $\hat{\phi}$ and $\tilde{\pi}$ in the AT model as informative priors for the CT model as described in Section 2.3.5.1, where we set $S = \{2, 5, 10\}$. We denote it as "CT+P" in the reported results.

In the one-shot learning scenario, CT and SVM classifiers are trained with the

synthesized training examples/informative priors obtained in the zero-shot learning test plus the first $M = \{1, 5, 10\}$ images of each target category. The AT model is trained with the first 100 images of each source category plus the first M images of each target category. DAP+NN uses the attribute predictions of the first M images of each target category as training data points to classify new images of target categories based on the nearest neighbor criterion.

In both zero-shot and one-shot learning tests, all classifiers are tested over the last 100 images of each target category and the mean of the diagonal of the confusion matrix is reported as the measurement of performance.

2.4.3 Results

Test 1: Overall Performance of Zero/One-Shot Learning. The overall performance of zero/one-shot learning are presented in the top row of Figure 2.5. These results show that the proposed approach outperforms the baseline algorithms in the following three aspects:

1. We have proposed a better attribute model for knowledge transfer. In both zero/one-shot learning tests, the AT model surpasses DAP and DAP+NN by 5.9% to 7.9%. Furthermore, all target classifiers that employ the prior knowledge from the AT model (SVM+S, CT+S and CT+P) achieve higher accuracy than DAP and DAP+NN. These results clearly show the advantages of the AT model in the attribute-based transfer learning framework.

2. We have proposed better methods of knowledge transfer for one-shot learn-

ing. In the one-shot learning test, the performance of the AT model does not improve compared to the zero-shot learning test. It is not a surprise: there are total 4000 images of source categories while only 10 images of target categories in training the AT model, thus the learned AT model will be almost the same as the one trained only with the 4000 source images. This result shows that the naive method of knowledge transfer will not work for the one-shot learning task. CT+S and CT+P achieve better balance between the prior attribute knowledge and the real example of target categories, and the additional single training example improves their accuracies by 0.9%-3% (CT+S) and 0.4%-1.4% (CT+P) respectively compared to their zero-shot learning results.

3. We have proposed a better target classifier. In both the zero-shot and one-shot learning tasks, the CT models (CT+S and CT+P) consistently exceed the baseline SVM classifier. Our experiments confirm the advantage of CT over the baseline algorithm, SVM, in the zero/one-shot learning tasks.

In addition to the above conclusion, we also have the following observations.

4. CT+S generally outperforms CT+P. CT+P can be viewed as an online version of CT+S, where the informative priors are equivalent to the initial values estimated from the synthesized examples in the initialization stage. Thus, samples drawn with CT+P are not distributed according to the true posterior distribution $P(z_{ji}|\mathbf{z}_{\backslash ji}, \mathbf{w})$, which includes all the synthesized and real training examples. As a result, the categorization performance is degraded.

5. With the increasing number of real training examples, the improvement on classification due to the prior knowledge decreases accordingly. This suggests that

the attributes do not contain all the information in target categories. Furthermore, some attributes may be difficult to learn and some are less informative to the categories. Thus when we have sufficient number of real training examples, the prior knowledge behaves more and more like noise and inevitably degrades the classification performance. We can thus derive a practical guideline from this observation to select an appropriate parameter S: when there is no or only one real training example, we can set a large value of S, e.g., 100 in CT+S or 10 in CT+P; when more and more real training examples are available, we then gradually reduce the value of S to zero.

Illustrations of the Attribute Models. We show two attribute models for *black* and *fast* in Figure 2.6. Though we employ the bag-of-features image representation and discard the spatial information in the image representation, the visual features related to the visual attributes *black* roughly localize the regions of interest. As discussed in Section 2.4.1, the non-visual attribute, *fast*, is most correlated to visual attributes *furry*, *tail* and *ground*. So the visual features related to these visual attributes are implicitly linked to *fast*. Visual examples in Figure 2.6 support this assumption. The influence of the non-visual attributes on the classification performance will be evaluated quantitatively in Test 2.

Test 2: The Influence of the Non-visual Attributes in the Transfer Learning. In this experiment, we remove the non-visual attributes from the class-attribute matrix and repeat the above tests. Results are illustrated in the middle row of Figure 2.5. Clearly, the absence of non-visual attributes degrades the classification performance enormously for all classifiers in both zero-shot and one-shot learning

scenarios. This test illustrates the importance of the non-visual attributes in the transfer learning approaches.

Test 3: The Effectiveness of the Knowledge of Attribute in the Transfer Learning. In this experiment, we use the AT model learned from source categories to generate synthesized training examples or compute informative priors following **randomly selected** attributes for each target category, where the number of random attributes are the same as that of the true attributes in each target category. The results show that the classification performance is at the chance level in the zero-shot learning tasks. In the one-shot learning task, the prior knowledge from the randomly selected attributes does not improve the classification performance compared to those not using attribute priors. This experiment highlights the effectiveness of the knowledge of the attribute.

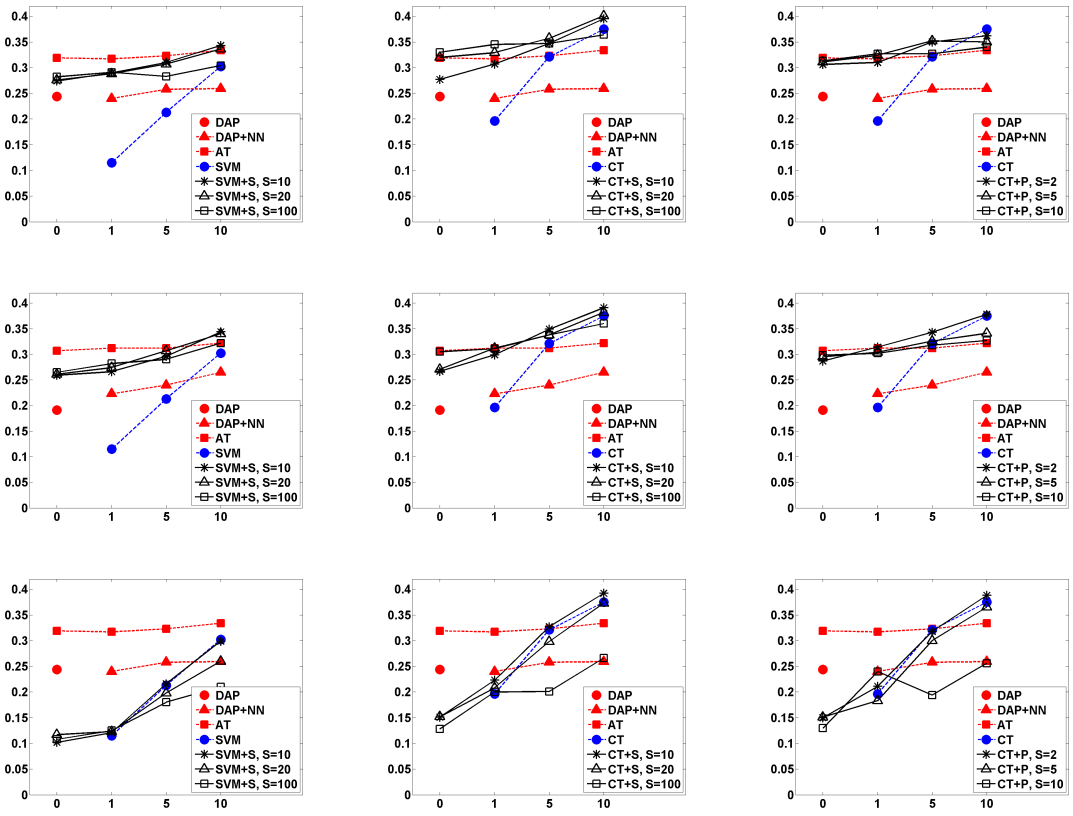


Figure 2.5: Results of zero-shot and one-shot learning in Test 1 (top row, using all attributes), Test 2 (middle row, using visual attributes only) and Test 3 (bottom row, using randomly selected attributes) for SVM+S (column 1), CT+S (column 2, CT model with synthesized training examples) and CT+P (column 3, CT model with informative parameter prior) respectively. The x-axis represents the number of real examples, M, and the y-axis represents the mean classification accuracy, i.e., the mean of the diagonal of the confusion matrix.

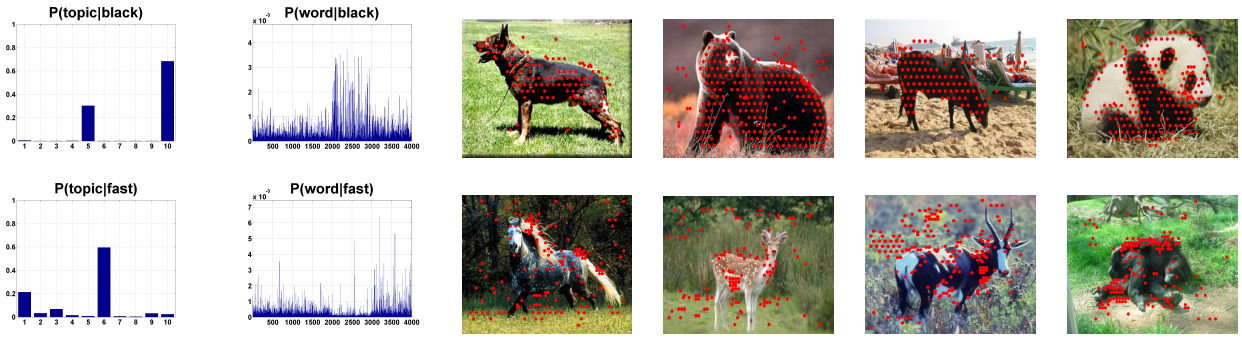


Figure 2.6: Illustrations of two attribute models for *black* and *fast* from the top to the bottom. Column 1: the distribution of the 10 topics assigned to a particular attribute; Column 2: the distribution of codewords for a particular attribute, where the codewords of SIFT, rgSIFT, local color histogram and LSS are presented from left to right on the x-axis respectively, with 1000 codewords for each type of feature; Column 3-6: examples of images from source categories (Column 3-5) and target categories (Column 6), superposed with the top 100 most likely codewords (solid red dots) for the attributes of the same row. Figures are best viewed in color.

Chapter 3

Active Scene Recognition Based on Its Attributes

3.1 Introduction

The paradigm of Active Vision [43, 44, 45, 46, 47] had invigorated Computer Vision research in the early 1990s. The ideas were inspired by the observation that in nature vision is used by systems that are active and purposive. By studying visual perception in isolation, we often end up with more complicated formulations and under-constrained problems. Thus, the Active Vision paradigm proposed that visual perception should be studied as a dynamic and purposive process for active observers that can control their imaging mechanism. Most previous work in this paradigm was concerned with low level robot vision problems, and applied the ideas to shape reconstruction and navigational problems, such as motion estimation, obstacle avoidance, surveillance and path planning. Higher level tasks of scene understanding and recognition have not been sufficiently studied in this framework. These problems require combining high level knowledge and reasoning procedures with low-level image processing and a systematic mechanism for doing so.

This idea is applied to the simplest interpretation problem, scene recognition, in this chapter. The proposed system consists of two modules: (1) the reasoning module, which obtains higher level knowledge about scene and object relations, proposes attentional instructions to the sensory module and draws conclusions about

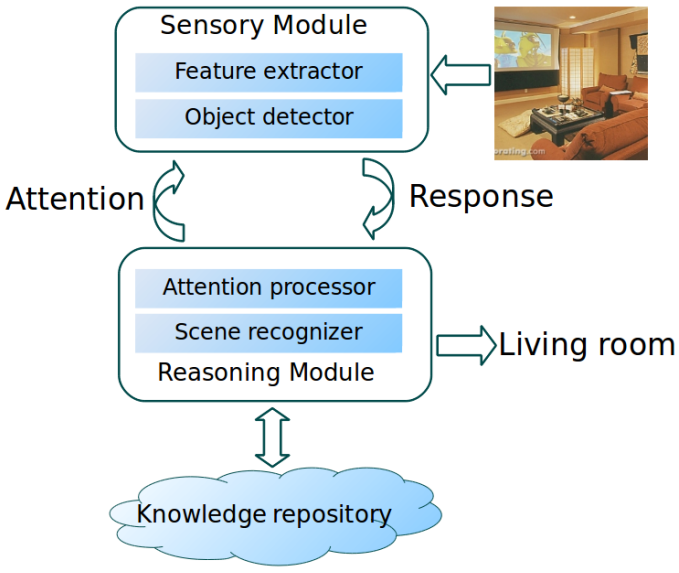


Figure 3.1: Overview of the active approach for scene recognition.

the contents of the scene; (2) the sensory module, which includes a set of visual operators responsible for extracting features from images, detecting and localizing objects and actions. The novelty of the proposed active paradigm is that the sensory module does not passively process the image; instead, it is guided by the reasoning module, which decides what and where the sensory module should process next. Thus the sensory module shifts the focus of attention to a small number of objects at selected locations of the scene. This leads to faster and more accurate scene recognition.

Figure 3.1 illustrates the interaction between the two modules, which is modeled as an iterative process. Within each iteration, the reasoning module decides on what and where to detect next and expects the sensory module to reply with some results after applying the visual operators. The reasoning module thus provides a focus of attention for the sensory module, which can be an object to be detected

and a place to be examined. For the problem of scene recognition, the interaction between the two modules is simple. (See Figure 3.7 for examples of the interaction over a given image.) However, our framework is more general. In Section 3.5 we discuss the extension of the framework to dynamic scene understanding. In this case the goal is to interpret the activity in the video. An activity is described by a set of quantities: the human, the tools, the objects, the motion, and the scene involved in the activity. Each of the quantities has many possible instances which can be described by their attributes (e.g., adjectives of nouns and adverbs of verbs). Thus the reasoning module at every iteration has to decide which quantity and which attribute to compute next. This procedure can be implemented in a hierarchical model of the proposed active scheme.

3.2 Related Work

3.2.1 Recognition by Components

: The methodology for object, scene and activity recognition in this paper follows the idea of "recognition by components", which can be traced back to early work by Biederman [20]. In this methodology, scenes are recognized by detecting the inside objects [48], objects are recognized by detecting their parts or attributes [14], and activities are recognized by detecting the motions, objects and contexts involved in the activities [49]. However, all previous works employ passive approaches. As a result, they need to run through all object/attribute detectors over the testing images and videos before making the final conclusion. In this paper we explore

an active approach, which aims at greatly reducing the number of object/attribute detectors needed for recognition of objects, scenes and activities.

3.2.2 Active Learning and Active Testing

: Our work is a type of active testing and is closely related to the visual "20 question" game described in [50]. While the approach in [50] needs human annotators to answer the questions posed by the computer, our approach is fully automated without a human in the loop.

To select the optimal objects/attributes, we use the criterion of Maximum Information Gain, which have been widely used for active learning of objects and scenes [51, 52]. Information theory also have been used for object localization in application of face detection [53].

3.2.3 Employing Ontological Knowledge in Computer Vision System for Scene Interpretation

: Ontological knowledge plays an important role in the reasoning and learning system of human. For example, in the problem of scene recognition, if we know that *coast* is a type of outdoor scene and also know that it is unlikely to find bookshelves therein. Hence, we do not need to apply the bookshelves detectors in the possible *coast* scene image. The work in [54] takes advantage of this type of knowledge in object detection. Similarly, the knowledge about objects and attributes is employed in [14]. Extending the knowledge about object hierarchy is employed in [55]. In this

paper, we further explore the ontological knowledge about activities and attributes and present a pilot study using a hand activity dataset.

3.3 The Approach

3.3.1 System Overview

The proposed active scene recognizer classifies a scene by iteratively detecting the objects inside it. In the k-th iteration, the reasoning module provides an attentional instruction to the sensory module to search for an object O_k within a particular region L_k in the image. Then the sensory module runs the corresponding object detector and returns a response, which is the highest detection score d_k and the object's location l_k. The reasoning module receives this response, analyses it and starts a new iteration. This iteration continues until some terminating criteria are satisfied. To implement such an active scene recognizer, we need to implement the following components: (1)a sensory module for object detection; (2) a reasoning module for predicting the scene class based on the sensory module's responses; (3) a strategy for deciding which object and where in the scene the sensory module should process in the next iteration; and (4) a strategy for initializing and terminating the iteration. We will describe these components in the rest of this section.

3.3.2 Scene Recognition by Object Detection

In the proposed framework, the reasoning module decides the scene class S based on the responses X from the sensory module, which we call Scene Recognition

by Object Detection (SROD). The optimal scene class of the given image belongs to the one that maximizes the probability:

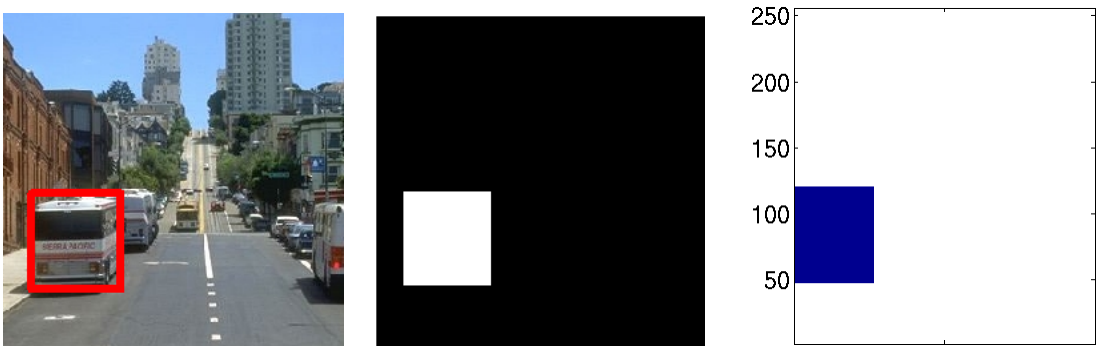
$$S^* = argmax_{S\in[1:M]} p(S|X), \tag{3.1}$$

where M is the number of scene classes.

The responses from the sensory module are a detection score and a detection bounding box. We only consider the objects' vertical positions, since they are more consistent within the images of the same scene class [54]. An object's vertical position is represented by a profile of the mask formed by the object's bounding box (see Figure 3.2 for an example). The object's mask formed by the object's bounding box is normalized to 256 $\times$ 256 pixels, and the profile is the histogram of pixels within the object's mask along the vertical axis. By this compact representation, we not only record the object's vertical location, but also record the object's scales along the horizontal and vertical axes. In the following, we denote this representation of an object's location as l_k.

As described above, in each iteration, the sensory module returns a detection score d_i and a detected location l_i for the expected object O_i. Thus at step k, we have accumulated a list of detected score $d_{1:k}$ and corresponding locations $l_{1:k}$. Given $X = (d_{1:k}, l_{1:k})$, the probability of a scene S is :

$$\begin{aligned} P(S|X) &= p(S|d_{1:k}, l_{1:k}) \\ &\propto p(d_{1:k}, l_{1:k}|S) \\ &= p(d_{1:k}|S)p(l_{1:k}|S). \end{aligned} \tag{3.2}$$



(a) (b) (c)

Figure 3.2: Representation of the object's location: (a) an object's bounding box; (b) the binary mask formed by the bounding box; (c) the profile of the object's binary mask along the vertical direction.

In the above equation, we assume $d_{1:k}$ and $l_{1:k}$ are independent given S. We approximate $p(d_{1:k}|S)$ by the inner product of $d_{1:k}$ and $\tilde{d}^S_{1:k}$, where $\tilde{d}^S_{1:k}$ is the mean $d_{1:k}$ of training examples for scene class S. Similarly, $p(l_{1:k}|S)$ is approximated by the inner product of $l_{1:k}$ and $\tilde{l}^S_{1:k}$. The advantage of this approximation is its simplicity and flexibility. We need to update the list of selected object in each iteration. If we adopt a parametric model for $p(d_{1:k}|S)$ and $p(l_{1:k}|S)$, we would need to learn the parameters for all permutations of $O_{1:k}, k = 1, ..., N$, where N is the total number of object categories in the dataset. For large N[1], such scheme would not work simply because of the computational constraints. Using a parameter-free approach, we avoid this difficulty.

[1]In our dataset, $N > 100$

3.3.3 Detecting Objects by The Sensory Module

The task of the sensory module is to detect the object required by the reasoning module and return a response. In this chapter, we applied three object detectors: a Spatial Pyramid Matching object detector [56], a latent SVM object detector [57] and the texture classifier by Hoiem [58]. For each object class, we train all three object detectors and then select the one with the highest detection accuracy on a validation dataset to use in the test. Given a test image, the object detector will find a few candidates with corresponding detection scores. The one with the highest score is selected and sent to the reasoning module. The detection scores are normalized by Platt scaling [59] to obtain probabilistic estimates.

3.3.4 Attentional Instructions by The Reasoning Module

The interaction between the reasoning and sensory modules at iteration k starts from an attentional instruction issued by the reasoning module, based on its observation history. In this chapter, the attentional instruction in iteration k includes *what* to look for, i.e., the object to detect (denoted as O_k) and *where* to look, i.e., the regions to detect (denoted as L_k). The criterion to select O_k and L_k is to maximize the expected information gain about the scene in the test image due to the response of this object detector:

$$\{O_k^*, L_k^*\} = \arg\max_{\substack{O_k \in \tilde{\mathcal{N}}_{k-1}, \\ L_k \in \mathcal{L}_k}} \mathrm{I}(S; d_k, l_k | d_{1:k-1}, l_{1:k-1}), \tag{3.3}$$

where $\tilde{\mathcal{N}}_{k-1}$ denotes the set of indices of objects that have not been detected until iteration k, $\mathcal{L}_k$ denotes the search space of O_k's location. The global optimization

procedure is approximated by two local optimization procedures. In the first step, we select O_k based on the maximum expected information gain criterion:

$$O_k^* = \arg\max_{O_k \in \tilde{\mathcal{N}}_{k-1}} \mathrm{I}(S; d_k, l_k | d_{1:k-1}, l_{1:k-1}). \tag{3.4}$$

Then L_k^* is selected by thresholding $\mathbb{E}_S[\tilde{l}_{O_k^*}^S]$, the expected location of object O_k^* across all scene classes.

The expected information gain of O_k given the previous response $d_{1:k-1}$ and $l_{1:k-1}$ is defined as:

$$\begin{aligned} &\mathrm{I}(S; d_k, l_k | d_{1:k-1}, l_{1:k-1}) \\ &= \sum_{d_k \in \mathcal{D}, l_k \in \mathcal{L}_k} p(d_k, l_k | d_{1:k-1}, l_{1:k-1}) \\ &\quad \times \mathrm{KL}[p(S | d_{1:k}, l_{1:k}), p(S | d_{1:k-1}, l_{1:k-1})]. \end{aligned} \tag{3.5}$$

The KL divergence on the right side of Equation 3.5 can easily be computed after applying Equation 3.2. To compute the first term on the right side of Equation 3.5, we factorize it as follows:

$$\begin{aligned} &p(d_k, l_k | d_{1:k-1}, l_{1:k-1}) \\ &\quad = p(d_k | d_{1:k-1}, l_{1:k-1}) p(l_k | d_{1:k}, l_{1:k-1}). \end{aligned} \tag{3.6}$$

The two terms on the right side of the above equation can be efficiently computed by their conditional probability with respect to S:

$$\begin{aligned} &p(d_k | d_{1:k-1}, l_{1:k-1}) \\ &= \sum_{S=1}^{M} p(d_k | S, d_{1:k-1}, l_{1:k-1}) p(S | d_{1:k-1}, l_{1:k-1}) \\ &= \sum_{S=1}^{M} p(d_k | S) p(S | d_{1:k-1}, l_{1:k-1}), \end{aligned} \tag{3.7}$$

where we assume d_k is independent of $d_{1:k-1}$ and $l_{1:k-1}$ given S. $p(d_k|S)$ can be computed by introducing the binary variable e_k, which indicates whether object O_k appears in the scene or not:

$$p(d_k|S) = \sum_{e_k \in \{0,1\}} p(d_k|e_k, S)p(e_k|S) \tag{3.8}$$

$$= \sum_{e_k \in \{0,1\}} p(d_k|e_k)p(e_k|S). \tag{3.9}$$

$p(e_k|S)$ encodes the high-level knowledge about the relationship between scene S and object O_k. One way to obtain it is to count the object labels in the training image set. Otherwise, we can obtain it from textual corpus. $p(d_k|e_k)$ encodes the information about the accuracy of different object detectors. The method to estimate its value is discussed in Section 3.4.1. The procedures described above are also employed to compute $p(l_k|d_{1:k}, l_{1:k-1})$ in a similar fashion.

Finally, we note that the expectation in Equation (3.5) needs to be computed at a set of sampling points of d_k (denoted as $\mathcal{D}$) and a set of sampling points of l_k (denoted as $\mathcal{L}_k$). $\mathcal{D}$ is within a one dimensional space between 0 and 1 and we draw samples of d_k uniformly. $\mathcal{L}_k$ can be parameterized by three parameters: the center position of O_k, y_k; the horizontal extent of O_k, w_k; and the vertical extent of O_k, h_k. We model these parameters by Gaussian distributions

$$y_k \sim \mathcal{N}(\mu_{y_k}, \sigma^2_{y_k}), \tag{3.10}$$

$$h_k \sim \mathcal{N}(\mu_{h_k}, \sigma^2_{h_k}), \tag{3.11}$$

$$w_k \sim \mathcal{N}(\mu_{w_k}, \sigma^2_{w_k}). \tag{3.12}$$

The means and variances of these Gaussian distributions are estimated from the

training set. Thus the problem of drawing a sample of l_k becomes the problem of drawing a sample of y_k, h_k, w_k from three Gaussian distributions.

After drawing samples of d_k and l_k, we substitute them into Equation 3.5 to compute the expected information gain for O_k. Then among all possible O_k's, we select the object that yields the maximum expected information gain, O_k^*. The distribution of O_k^*'s location in a particular scene S is approximated by $\tilde{l}^S_{O_k^*}$, which is computed as follows: first, we aggregate $l_{O_k^*}$ in all training samples of scene class S in the training stage; then we normalize the accumulated values into $[0, 1]$. Thus the expectation of $\tilde{l}^S_{O_k^*}$ across all scene classes, $\mathbb{E}_S[\tilde{l}^S_{O_k^*}]$, represents the distribution of O_k^*'s location in an image of any scene class. Finally, we threshold this value by 0.5 and obtain a binary L_k^*, which provides the focus of attention for the sensory module in the next iteration.

3.3.5 Initializing and Terminating the Iteration

The interaction between two modules starts from the first object and its expected location, which are provided by the reasoning module. We select the object O_1 that maximizes the mutual information

$$O_1^* = \arg\max_{O_1 \in [1:N]} \mathrm{I}(S; d_1, l_1). \tag{3.13}$$

To terminate the iteration, we can either stop after a fixed number of iterations (e.g., the 20 question game), or stop when the expected information gain at each iteration is below a threshold. In our experiments, we followed the former approach and found that 30 iterations are sufficient to produce competitive recognition results.

3.4 Experiments

3.4.1 Image Datasets

We evaluate the proposed approach using a subset of the SUN image set from [60]. Overall, the SUN dataset[60] contains 12K images, 1K scene classes and more than 200 object classes. We sort the scene classes by the number of examples and select the top 20 that have more than 50 examples per scene class. The remaining scene classes are discarded since they do not have sufficient number of examples to evaluate our algorithms. For each of the 20 selected scene classes, we randomly select 50 examples, where 30 of them are used for training and the rest 20 for testing. At the end, there are 127 object classes within our subset of SUN image set but only a handful of object classes appear in a particular scene class. As discussed in [60], a typical scene image contains seven object classes. The object detectors are trained using an additional dataset of 26,000 objects that is disjoint from the training/testing scene images as described in [60]. The obtained object detectors are then evaluated in the 600 scene training examples. The detection score, d_k, is normalized into $[0, 1]$ and evenly quantized into 10 discrete values. For each e_k (0 or 1), we accumulate the counts of d_k for each of its 10 values. Due to its discrete nature, $p(d_k|e_k)$ can be modeled as a multinomial distribution. Since Dirichlet is the conjugate prior for multinomial, we use a Dirichlet distribution $\mathrm{Dir}(\alpha)$ as the prior for $p(d_k|e_k)$, where α represents the number of prior observations of d_k given a particular e_k. Through all experiments in this chapter, we set the parameter $\alpha = 1$. We also tried other values of α and found no significant performance impact.

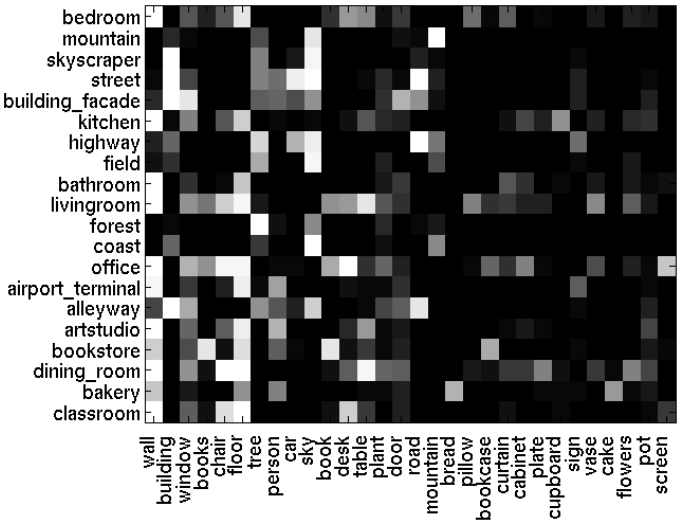


Figure 3.3: Co-occurrence of the 20 scene and 30 objects in the scene dataset, measured by $p(e = 1|S)$. The brighter the boxes, the more is the object associated with the scene.

Figure 3.3 visualizes the scene-object co-occurrence relationship between the 20 scenes and the most frequent 30 objects.

3.4.2 Performance of the Scene Recognizer

In the first experiment, we evaluate the scene recognizer (SROD) as described in Equation 3.2 while all objects are detected. The "ideal" SROD, where we use the object ground truths as the outputs of object detectors, is also evaluated to illustrate the upper limit of the performance of SROD. Three baseline algorithms are evaluated as listed below:

- SVM using GIST [15] features.
- SVM using Bag-of-Words (BoW). We used two types of local features, SIFT

[39] and opponent SIFT [61], and the size of visual word dictionary is set as 500 for each of them.

- Classification and Regression Tree (CART) [62] that uses the object detection scores as attributes to predict the scene classes of a given image. The "ideal" CART, where the object ground truth is used as attributes, is also evaluated to illustrate the upper limit of the performance of CART.

Figure 3.4 compares the scene classification accuracy of these baseline algorithms and the SROD approach. The SROD approach significantly outperforms all the baseline algorithms. This result confirms the effectiveness of object-based approaches in interpreting complex scenes and the robustness of the SROD approach to the errors in object detection. It is worth to emphasize that there is still a lot of room to improve the current object-based scene recognizer, as suggested by the performance of the ideal SROD.

In addition, we evaluate the robustness of these scene recognition approaches with respect to the size of training samples. We randomly select a number of training examples from the training image set for each scene class and repeat the experiments three times. The mean and standard deviation of the average accuracy when using 5, 10, 15, 20, 25 and 30 training examples are reported in Figure 3.5. The proposed SROD method achieves substantially better performance than all baseline algorithms, including the CART algorithm that uses the same outputs of object detectors.

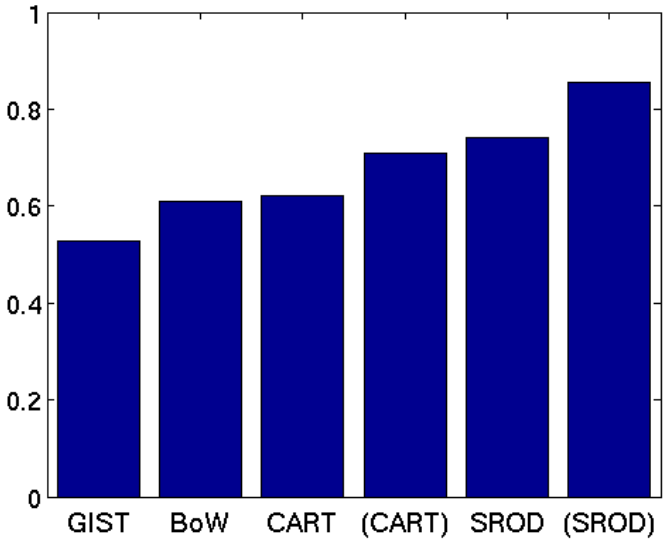


Figure 3.4: Comparison of scene classification accuracy of different approaches (GIST+SVM vs. BoW+SVM vs. CART vs. SROD). We also illustrate the "ideal" performance of CART and SROD, where we use the object ground truths as the outputs of object detectors. They are represented as "(CART)" and "(SROD)" in the figure respectively.

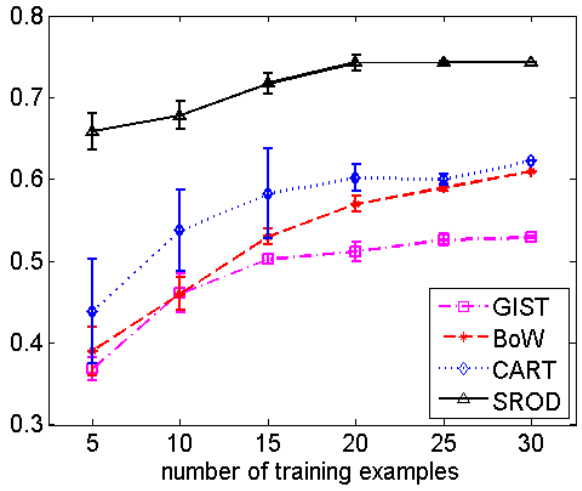


Figure 3.5: Classification accuracies of different approaches (GIST+SVM vs. BoW+SVM vs. CART vs. SROD) with respect to the number of training images.

3.4.3 Comparison of the Active Scene Recognizer vs. the Passive Scene Recognizer

In this experiment, we compare the proposed active scene recognizer with two baseline algorithms and the results are presented in Figure 3.6. Both baseline algorithms recognize scene class by iterative object detection, which is similar to the proposed SROD method. But they employ different strategies to select the to-be-detected object in each iteration. The first baseline (denoted as "DT" in Figure 3.6) follows a fixed object order, which is provided by the CART algorithm; while the second baseline (denoted as "Rand" in Figure 3.6) just randomly selects an object from the remaining object pool. Object selection obviously has a big impact on the performance of scene recognition, since both the proposed active approach and the "DT" approach significantly outperform the "Rand" approach. The result also

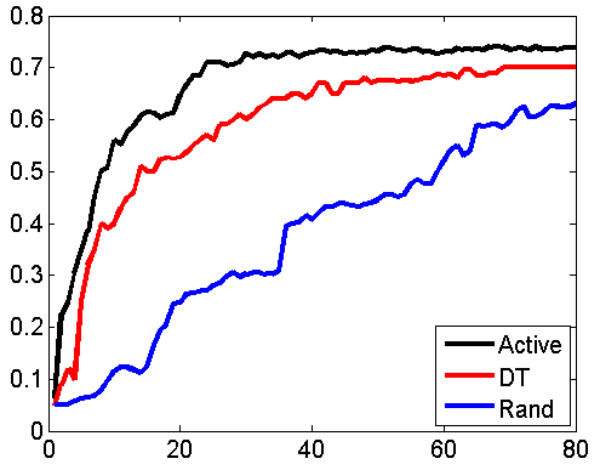


Figure 3.6: Comparison of classification accuracy among different object selection strategies (active vs. passive vs. random) in the object-based scene recognizers, with respect to the number of training images.

shows that the active approach is superior to the "DT" approach that is passive: the active approach can achieve stable performance after selecting 30 objects while the passive "DT" approach needs 60 objects. Furthermore, the object's expected location provided by the reasoning module in the active approach not only reduces the spatial search space to be about 1/3 to 1/2 of the whole image but also reduces the false positives in the sensory module's response. As a result, the proposed active approach achieves 3% to 4% performance gain compared to the passive approaches.

3.4.4 Visualization of the Interaction between the Sensory Module and the Reasoning Module

Figure 3.7 illustrates a few iterations of the active scene recognizer performed on a test image. The iteration starts from the most discriminative object *wall*. The sensory module gives a pretty confident answer, $d_1 = 0.744$. But this answer does not give us much information, since we can see from Figure 3.3, *sky* is widely shared among these scenes and $p(S|e)$ for $e_{sky} = 1$ is almost a uniform distribution among 8 scenes. So the belief of the language agent about the scene is still very ambiguous.

The second question asked by the language agent is *bus*. If the answer is positive, then the scene is likely to be *street*. But the vision agent gives a very low detection score, which means that the scene can equally belong to one of the seven scenes except the street. So the belief of the language agent is still ambiguous.

The third question is *sand* and the answer is positive. Immediately, the language agent can conclude that the scene is very likely to be *coast*, since only the *coast* scene contains *sand*. The next three questions are all objects that can not appear in the *coast* scene and their low detection scores confirm the language's belief that this is a *coast* scene. So the language agent can stop asking questions at this moment.

3.5 Dynamic Scene Recognition

There are two key premises in the proposed active scheme: (1) a quantity can be recognized by accumulating evidences from its components; (2) the components

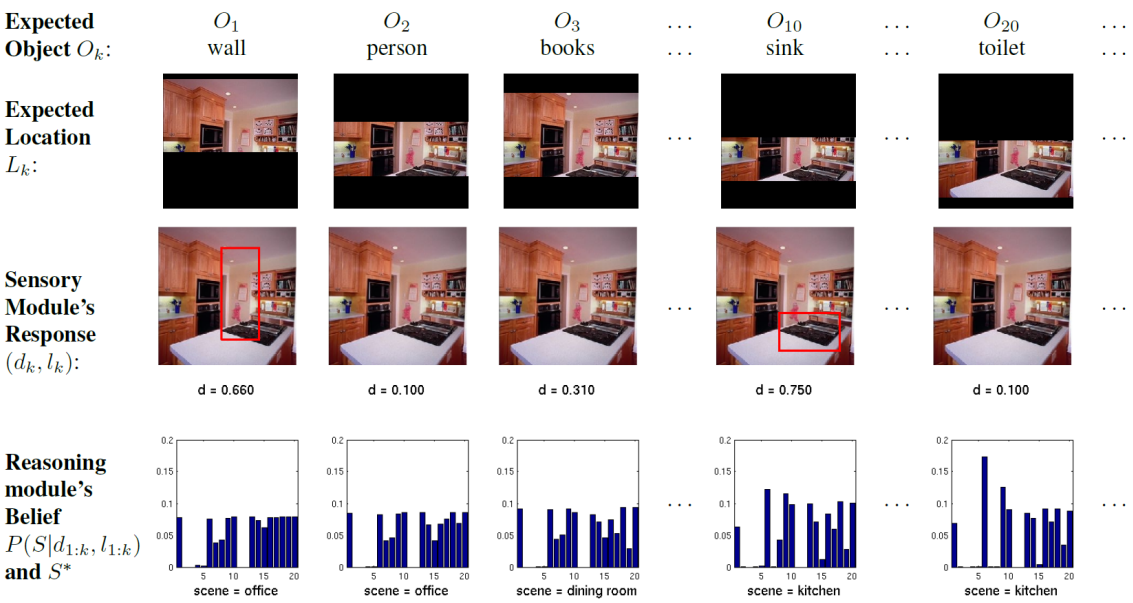


Figure 3.7: Visualization of the iterations between the reasoning module and the sensory module in an active scene recognition process. The detected regions with detection score greater than 0.5 are highlighted with a red bounding box.

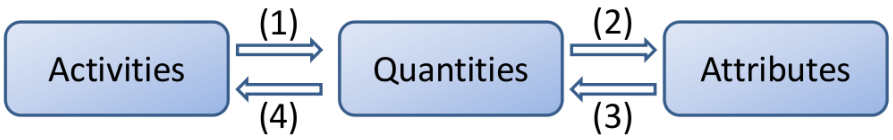


Figure 3.8: Hierarchical active scheme for dynamic scene recognition, where each iteration invokes four steps. Section 3.5 discusses the details.

can be assumed to be independent given the quantity. Given these two premises, the active scheme can be applied to select a small number of components to recognize the quantity without impairing the performance. In the previous section, we have applied this active scheme to recognize static scenes. However, this active scheme can also be applied to recognize objects by their parts and recognize activities by their motion and object properties.

In this section, we will demonstrate the application of the active scheme in an activity recognition problem. A big challenge in this problem is that the components are heterogeneous. While static scenes only involve a single quantity, i.e., objects, activities are described by different quantities, including motion, objects and tools, scenes, temporal properties, etc. To alleviate this problem, we propose a hierarchical active scheme for dynamic scene recognition. Figure 3.8 presents this method. In this scheme, each iteration performs the following four steps: (1) using the maximum information gain criterion, the activity-level reasoning module sends an attentional instruction to the quantity-level reasoning module that indicates the desired quantity (e.g., motion or objects); (2) the quantity-level reasoning module then sends an attentional instruction to the sensory module that indicates the

desired attributes (e.g., object color/texture, motion properties); (3) the sensory module applies the corresponding detectors and returns the detector's response to the quantity-level reasoning module; (4) finally, the quantity-level reasoning module returns the likelihood of the desired quantity to the activity-level reasoning module.

To demonstrate this idea, we used 30 short video sequences of 5 hand actions from a dataset collected from the commercially available PBS *Sprouts* craft show for kids (the hand activity data set). The activities are *coloring*, *drawing*, *cutting*, *painting*, and *gluing*. 20 sequences were used for training and the rest for testing. Two quantities are considered in recognizing an activity: the characteristics of tools and the characteristics of motion. Four attributes are defined for the characteristics of tools, including *color*, *texture*, *elongation*, and *convexity*; and four attributes are defined for the characteristics of motion, including *frequency*, *motion variation*, *motion spectrum*, and *duration*. The details of these quantities and attributes are described in Table 3.1.

The interaction between language and vision is as follows: First the hands and the objects in the hands (the tools) are visually segmented. Using the knowledge about action-attribute co-occurences, the dialogue then consists of the reasoning module asking repeatedly: "Which attribute should I compute next?" such that there is maximum information gain.

The sensory module includes detectors for the 8 attributes of tools/motion. To detect these attributes, we need to segment the hand and tools from the videos. Figure 3.10 illustrates these procedures, which are described as follows:

Quantity	Attribute	$e = 1$	$e = 0$
Tools	Color	silver	other colors
	Texture	bristle	non-bristle
	Elongation	yes	no
	Convexity	yes	no
Motion	Frequency	high	low
	Motion variation	large	small
	Motion spectrum	sparse	non-sparse
	Duration	long	short

Table 3.1: Activity attributes in the hand activity dataset.

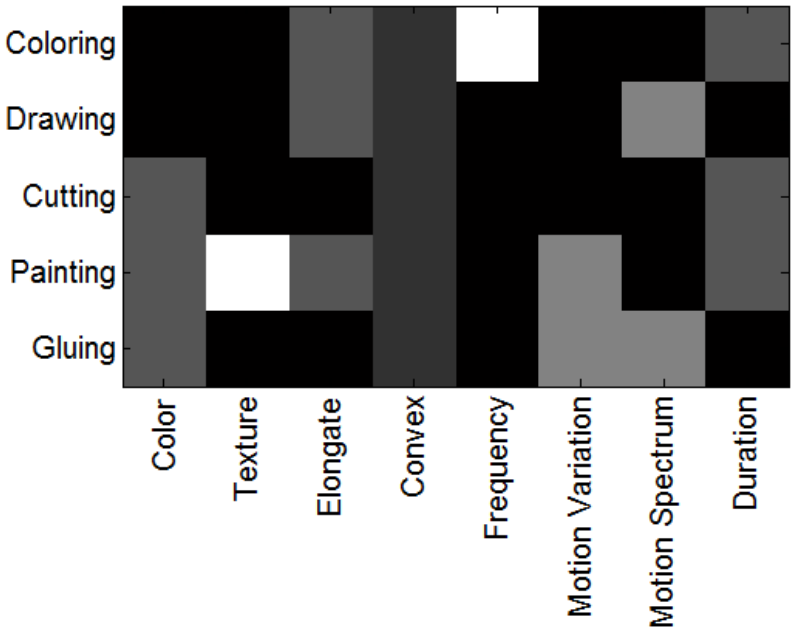


Figure 3.9: Co-occurrence of the 5 actions and 8 attributes in the hand action dataset, measured by $p(S|e = 1)$. See Figure 3.3 for more information about color codes in this figure.

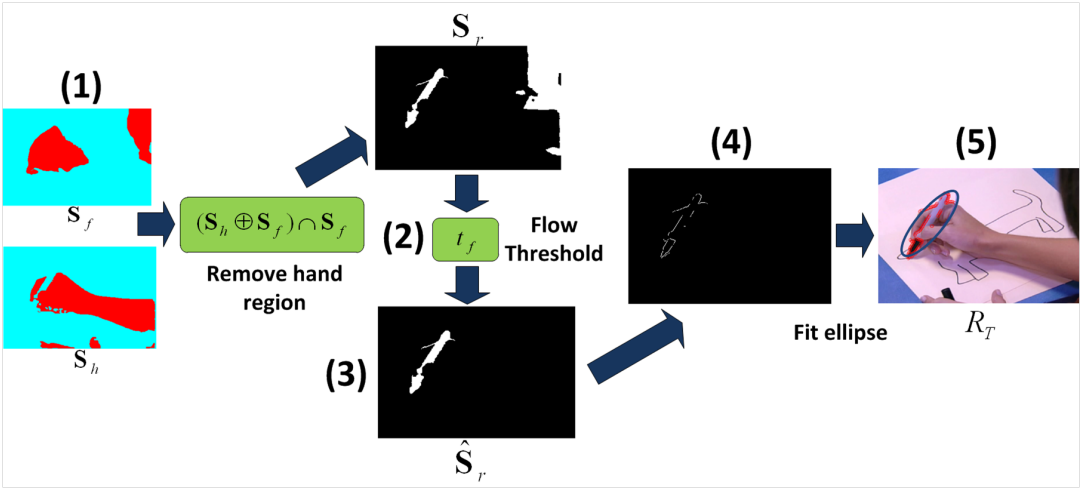


Figure 3.10: Procedures to extract hands and tools from the hand activity video sequence. Please refer to the text for details.

1. Hand regions S_h are segmented by applying a variant of the color segmentation approach based on Conditional Random Fields (CRF) [63] using a trained skin color model. Similarly, moving regions of hands and tools, S_f, are segmented by applying another CRF over the optical flow fields.

2. A binary XOR operation is applied on S_h and S_f to remove the moving hand regions and produce a segmentation of tools, S_T.

3. Apply a threshold t_f to remove regions with flows that are different from the hand regions and obtain a candidate region for tool, $\hat{S}_r$.

4. Detect edges in $\hat{S}_r$.

5. Fitting a minimum volume ellipse over the edge map of $\hat{S}_r$, which estimates the region of the detected tool.

Figure 3.11 shows the estimated ellipse enclosing the detected tool over some

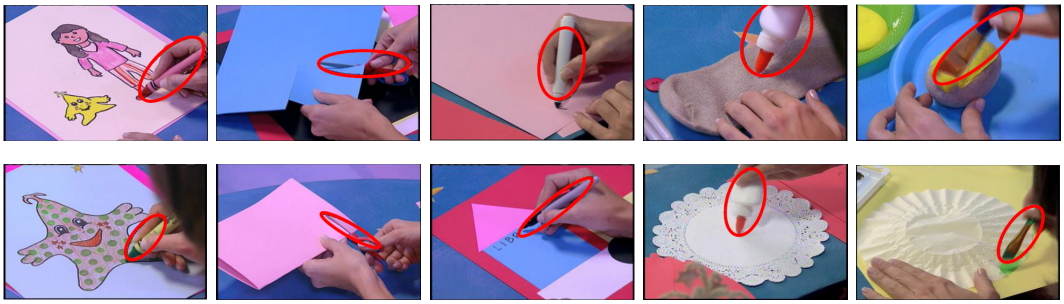


Figure 3.11: Sample frames for 10 testing videos in the hand action dataset: (from left to right) coloring, cutting, drawing, gluing, painting. The detected tool is fit with an ellipse.

sample image frames from the dataset. This ellipse, together with $\hat{S}_r$, is then used as a mask to compute object-related attributes. The color and texture attributes were computed from histograms of color and wavelet-filter outputs, and the shape attributes were derived from region properties of the convex hull of the object and the fitted ellipse. The motion attributes were computed from the spectrum of the average optical flow over the sequence and the variation of the flow.

Table 3.2 shows the interactions between the reasoning modules and the sensory modules for one of the testing videos. Here the sensory module only needed to detect two attributes before the reasoning module arrived at the correct conclusion. Overall, 8 out of 10 testing videos were recognized correctly after detecting two to three attributes, while the remaining two testing videos could not be recognized correctly even after detecting all the attributes. This is because of errors in the segmentation, the choice of attributes and the small set of training samples.

Note that the active approach needed to detect only two to three attributes while a passive Nearest Neighbor classifier needs all 8 attributes. This is theoretically

Iteration	1	2	3	4
Expected quantity	Tools	Tools	Tools	Motion
Expected attribute	Elongation	Color	Texture	Duration
Sensory module's response	0.770	1.000	0.656	0.813
Reasoning module's conclusion	Coloring	Painting	Painting	Painting
Reasoning module's confidence	0.257	0.770	0.865	0.838

Table 3.2: An example of interactions between the reasoning module and the sensory module for hand activity recognition, where the ground truth of the activity class is *painting*.

the minimum number of attributes we can detect in the ideal case since $3 = \log(8)$. We observe that this excellent performance is largely due to two reasons: first, the attributes we selected are very discriminative; second and more importantly, the assumption made in Equation 3.7 that the an attributes are action class is independent of attribute location is satisfied very well in the action domain.

Chapter 4

Action Attribute Detection from Sports Videos with Contextual Constraints

4.1 Introduction

In this chapter, we study the problem of detecting action attributes from sport videos. Action attributes include atomic components of action classes (such as the motion patterns of human limbs and body), contextual components of action classes (such as the objects and scenes involved in the action), and non-semantic attributes, a.k.a data-driven attributes [19]. A common property of action attributes is that they can be generalized into different action classes. This is especially true in sports videos. For example, *bend*, as an action attribute that describes the motion of human body, is present in the action `tennis serve`, `bowling`, `snatch`, etc. That is why they can be learned even from training sets that contain only a few examples for each action class (*one-shot* learning) or even no example for some action classes (*zero-shot* learning). We focus on the action attributes related to motion patterns of human body in this chapter; however, our model can be easily extended to detect the other types of action attributes as well.

Though in literature the concept of action attribute has been introduced in [19], our goal of this work is much beyond the previous work. First, we want to

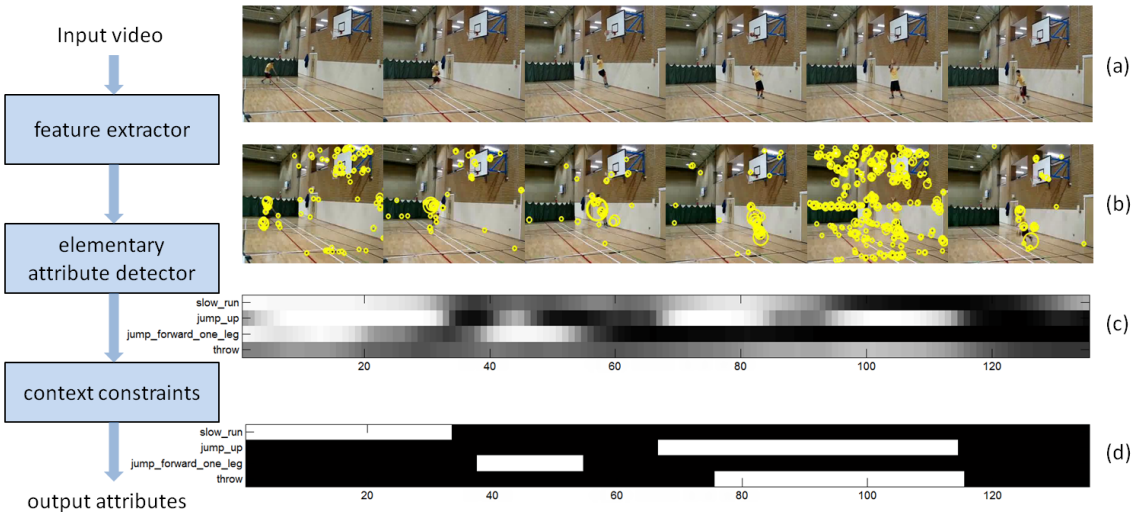


Figure 4.1: Overview of the proposed system including the key components (left) and example inputs/outputs (right): row (a) shows example video frames along the timeline; row (b) shows the detected STIP interest points; row (c) shows the probability output from the elementary attribute detector (the brighter the color, the higher the probability that corresponding attribute presents in that frame); row (d) shows the final attribute label after applying context constraints (bright color represents positive label and black represent negative). The frame numbers are illustrated on the bottom of row (c) and (d).

label the action attributes at each video frame rather than the whole video of the same activity class, as did in the previous work [19]. In reality, each video could have different attributes and this is even true for videos of the same activity class. For example, in a video of the `snatch` activity, the athlete may not be able to completely lift the barbell above his head at the end so we cannot say this video has an action attribute *two arms raise pose*. Next, we believe that the temporal structure of the action attributes are very important and should be preserved in the detection results. This aspect has been totally ignored in the previous work. For example, given a video of `basketball layup`, a description *"the athlete starts with a slow run and lasts for half a second, then jumps forward with single leg in the next second, finally jumps up and throws the ball (into the basket), and maintains a slow run at the end of the video"* will be more useful than simply saying *"there are slow running, jumping forward, jumping up, throwing in this video"*.

While having great advantages as discussed above, it is obviously a much more challenging task to locate the temporal occurrences of every action attributes in a given video. As a high-level semantic concept, a particular action attribute may exhibit significant variability due to viewpoint changes, photometric measurements, intra-class variability (e.g. attribute *two arms open* can have different opening angles between two arms, *jump up* can have different height and velocity, etc). Naive detectors that rely entirely on local features will easily be overwhelmed by a large number of false positives and/or false negatives. So we must take into account the contextual constraints in both the temporal and semantic domains, thereby reducing the noise in the local feature space to produce more reliable results. This

is exactly the theme of this chapter.

4.1.1 Related Work

Recognizing action classes in videos is a basic component towards understanding and describing videos. Recent progress in this field is achieved by introducing various type of descriptive features, e.g., HOG (Histograms of Oriented Gradients) and HOF (Histograms of Optical Flow) around 3D video patches using Spatial-Temporal Interest Points (STIP) [64], or optical flow [65]. Based on these low-level features, actions in videos can be represented by bag-of-words [66] or ballistic dynamics [67]. Such features and video representations can also be used in detecting other motion patterns in videos. For example, in this chapter, we use the bag-of-words video representation to detect action attributes from video, which are computed from HOG and HOF features around detected STIPs.

A recent advancement in this field is the emerging importance of contextual information. Due to the noise and intra-class variance in videos, approaches that rely solely on low-level features are often prone to false alarms. Instead, exploiting contextual information provides us with additional constraints so that we can produce more reliable results. In the literature, various types of contextual constraints have been explored, such as action-object context [68, 69, 70, 71], action-scene context [72], action-action context [73], etc.

4.1.2 Our Contribution

The contribution of this work is three-fold. First, we propose to detect action attributes at a finer granularity. Our work and the previous work in [19] is only comparable at the level of localizing an object in an image (e.g., with a bounding box or segmentation) and then detecting the presence of an object class in an image. Our proposed action attributes can enable a machine to understand videos at a higher level and thus provide more sensible video descriptions to humans. Second, we address the challenges in action attribute detection by utilizing contextual information in which a factorial conditional random field is used to model the rich relationships between attributes in both the temporal and semantic domains. Finally, we labeled a fully annotated action attribute dataset to evaluate our algorithms. This dataset will be made publicly available and we believe it serves to help other researchers study action attributes and many other interesting problems, such as automatic subscripting, transfer learning, human tracking and pose estimation in difficult conditions, etc.

4.2 Detecting Action Attributes using Contextual Constraints

4.2.1 Systematic Overview

A systematic overview of our approach is shown in Figure 4.1. It is composed of three parts: feature extractor, which extracts low-level features from a video; elementary attribute detector, which detects attributes in a video using local cues

only; and context constraints, which combine outputs from the elementary detectors of all attributes so as to determine a set of globally optimized attribute labels. We use off-the-shelf algorithms in the first two parts and a factorial conditional random field to incorporate the contextual constraints in the last part. The details are presented in the rest of this section.

4.2.2 Low-level feature extraction

Since our goal is to detect human action attributes, we need a module to detect human and extract motion features within the human bounding box. This problem has been thoroughly studied and numerous algorithms have been proposed in the last few decades. Interested readers can refer to [74, 75, 57, 76, 77] for a few exemplar implementations. Since this is beyond the scope of this thesis, we assume that this step has been done and simply use the annotated bounding box as the one produced by any algorithm that does human detection and tracking. In this way, we can measure the upper bound on the performance of various attribute detectors described in the next section.

The low-level features for representing motions in a video are HOG and HOF, which are extracted at the detected interest points using the author's latest implementation of STIP [78] with a default noise threshold and video resolution of 320×240. All descriptors within the human bounding box are quantized into one of the 400 visual words, which are computed using k-means from 40,000 randomly selected descriptors. At the end, each frame is represented by a histogram of vi-

sual words, $\mathbf{x} \in \mathbb{X}^d$, which counts the number of quantized descriptors within the bounding box of the human in a frame and its two closest neighbours.

4.2.3 Elementary attribute detectors

Our elementary attribute detector is an SVM with a χ^2 kernel, which takes the histogram of visual words as inputs and predicts the probability of a particular action attribute occurring in each frame of an unseen video. The kernel function $K(\mathbf{h}_i, \mathbf{h}_j)$ is given as in Section 3.4 of [42]:

$$K(\mathbf{h}_i, \mathbf{h}_j) = \exp\left\{-\frac{1}{S}\sum_{k=1}^{D}\frac{(h_{ik} - h_{jk})^2}{h_{ik} + h_{jk}}\right\}, \tag{4.1}$$

where S is scaling factor that is set as the mean distance among all training samples as suggested in [42], and D is the histogram dimensionality.

During training, we have training videos with attributes labelled at each frame. For each attribute, we select features from all positive frames as positive examples and randomly select an equal number of negative examples to train the SVM. For testing, we first extract histogram of visual words from each frame t in an unseen video, denoted as $\mathbf{x}_t$. Then the SVM for attribute a predicts the presence of this attribute in frame t, and the output probability value is denoted as $N_a(\mathbf{x}_t)$. This value will be used as the node feature of the conditional random field described in the next section. The SVM also predicts a binary value to indicate the presence/absence of attribute a at frame t by applying a threshold of 0.5 over $N_a(\mathbf{x}_t)$. This will be used as a baseline in our evaluation. See Section 4.3.2 for more details.

4.2.4 Incorporating Contextual Constraints

4.2.4.1 Factorial Conditional Random Field Model

We take into account two types of contextual constraints in this chapter: the temporal context and the semantic context. To promote agreement between the different attribute labels at different frames, we model the contextual constraints with a conditional random field (CRF) as shown in Figure 4.3. The features extracted from T frames in a video are denoted as a vector of T local observations, $\mathbf{X} = \{x_1, x_2, ..., x_T\}$; and at each local observation at frame t, by $\mathbf{x}_t$, the histogram of visual words as discussed in the preceding section. At each frame t, we wish to detect the presence of A action attributes $\mathbf{y}_t = \{y_t^1, y_t^2, ..., y_t^A\}$, which are the states in the CRF model. In the literature, this is also known as a factorial CRF [79]. To better understand this CRF model, we can compare it with a linear chain CRF [80] as shown in Figure 4.2, which has been used in action class recognition from video streams [81]. In a linear chain CRF, the state of each time point is dependent on its immediate neighbors only (Markovian assumption) and we enforce agreement between states in adjacent time points after accounting for the correlation between the neighboring temporal states. In a factorial CRF, we have multiple states at each time point, $\mathbf{y}_t = \{y_t^1, y_t^2, ..., y_t^A\}$, and there are edges between every pair of y_t^i and y_t^j, $(i, j) \in \{1, 2, ..., A\}$[1]. To avoid clutter, we only show two attributes at each

[1]For clarity, we call the edges between states of the same time points as *between-chain edges* and the edges between states of adjacent time points as *within-chain edges*. In Figure 4.3, the former are colored in red and the latter in black

time point in Figure 4.3. In the experimental dataset used, there are 24 attributes at each time point. The between-chain edges are designed to promote agreement between different attributes at the same time point. The intuition is that some attributes tend to occur together, e.g.*two arms oscillate* and *fast run* while others don't, e.g. *slow run* and *fast run*. Thus the between-chain edges take into account the co-temporal correlation among attributes.

The factorial CRF is defined as follows

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \left(\sum_{t=1}^{T} \sum_{a=1}^{A} \Upsilon_t(y_{t,a}, \mathbf{X}) \right) \left(\sum_{t=1}^{T-1} \sum_{a=1}^{A} \Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X}) \right) \left(\sum_{t=1}^{T} \sum_{a,b \in \{1,\ldots,A\}, a \neq b} \Phi_t(y_{t,a}, y_{t,b}, \mathbf{X}) \right), \tag{4.2}$$

where $Z(\mathbf{X})$ is the partition function, $\{\Upsilon_t\}$ the local (node) potential functions, $\{\Psi_t\}$ the within-chain potential functions, and $\{\Phi_t\}$ the between-chain potential functions. The potential functions are defined by a set of feature functions $\{f_k\}$ together with corresponding weights $\{\lambda_k\}$ as:

$$\begin{aligned} \Upsilon_t(y_{t,a}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, \mathbf{X}) \right) \\ \Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, y_{t+1,a}, \mathbf{X}) \right) \\ \Phi_t(y_{t,a}, y_{t,b}, \mathbf{X}) &= \exp\left(\sum_k \lambda_k f_k(y_{t,a}, y_{t,b}, \mathbf{X}) \right). \end{aligned} \tag{4.3}$$

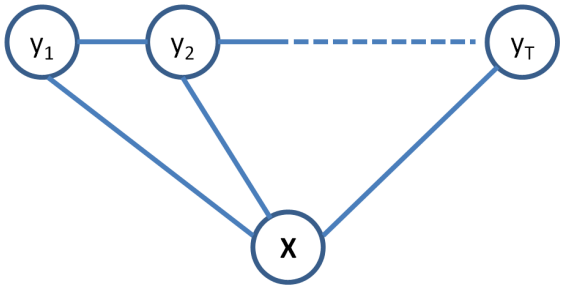


Figure 4.2: A linear chain CRF model.

The feature functions for the above potential functions are defined as follows:

$$f_k(y_{t,a}, \mathbf{X}) = \mathbb{I}[y_{t,a} = m] \log N_a(\mathbf{x}_{t-j})$$

$$f_k(y_{t,a}, y_{t+1,a}, \mathbf{X}) = \mathbb{I}[y_{t,a} = m \wedge y_{t+1,a} = n]$$

$$f_k(y_{t,a}, y_{t,b}, \mathbf{X}) = \mathbb{I}[y_{t,a} = m \wedge y_{t,b} = n]\phi(a, b) \tag{4.4}$$

where $j \in [-W, W]$[2], $m, n \in \{0, 1\}$, $a, b \in \{1, ..., A\}$, $\mathbb{I}[x = A]$ is an indicator function so that $\mathbb{I}[x = A] = 1$ if $x = A$, or 0 otherwise. Intuitively, the node feature functions encode correlations from the presence of attribute a and the observation confidence forward or backward within a time window; the within-chain edge feature functions encode the transition probability between adjacent time point for attribute a; the between-chain edge feature functions encode the compatibility of two attributes, a and b, measured by $\phi(a, b)$ and is obtained by normalizing the co-occurrence frequency between these two attributes. Figure 4.3.1 illustrates the normalized co-occurrence matrix that is computed from the training set in our experiments.

[2]where $W = 1$ is the size of the sliding window around a video frame

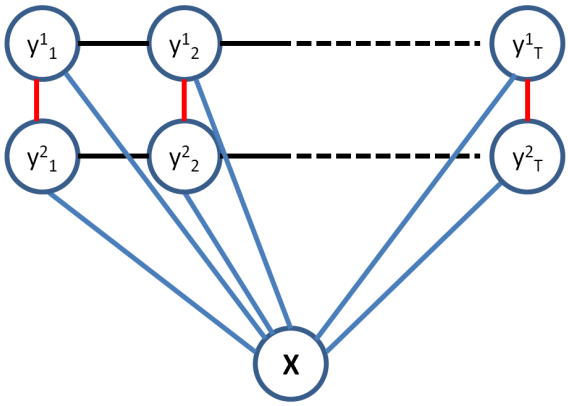


Figure 4.3: A factorial CRF model. To avoid clutter, we only show two attributes at each time points. In reality, we can have as many as attributes at each time points and they are fully connected to each other.

4.2.4.2 Learning Model Parameters

In general, given a training set consisting of N sequences $\mathcal{D} = \{\mathbf{X}^n, \mathbf{Y}^n\}_{i=1}^N$, we want to find the optimal parameter $\mathbf{\Lambda}^* = \{\lambda_k^*\}$ that maximizes the following objective function, as discussed in [80],

$$L(\mathbf{\Lambda}) = \sum_{n=1}^{N} \log P(\mathbf{Y}^n | \mathbf{X}^n, \mathbf{\Lambda}), \tag{4.5}$$

where the right hand side is the conditional log-likelihood of the training data. The partial derivative of the log-likelihood w.r.t λ_k associated with clique index c is [79]

$$\begin{aligned} \frac{\partial L}{\partial \lambda_k} &= \sum_n \sum_t f_k(\mathbf{Y}_{t,c}^n, \mathbf{X}^n) \\ &\quad - \sum_n \sum_t \sum_{\mathbf{Y}_{t,c}} p(\mathbf{Y}_{t,c} | \mathbf{X}^n) f_k(\mathbf{Y}_{t,c}^n, \mathbf{X}^n) \end{aligned} \tag{4.6}$$

where $\mathbf{Y}_{t,c}^n$ is the assignment to $\mathbf{Y}_{t,c}$ in $\mathbf{Y}^n$, and $\mathbf{Y}_{t,c}$ ranges over assignments to the clique c at time point t. The first term in the right hand side is easy to compute. The

second term computes marginal probabilities $p(\mathbf{Y}_{t,c}|\mathbf{X}^n)$, which will be discussed in Section 4.2.4.3.

To reduce overfitting, we define a spherical Gaussian prior [79] to the parameter, which has zero mean and covariance matrix $\Sigma = \sigma^2\mathbf{I}$, i.e.,

$$\mathbf{\Lambda} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{4.7}$$

and this is equivalent to optimizing $L(\mathbf{\Lambda})$ using ℓ^2 regularization:

$$L_r(\mathbf{\Lambda}) = \sum_{n=1}^{N} \log P(\mathbf{Y}^n|\mathbf{X}^n, \mathbf{\Lambda}) - \frac{1}{2\sigma^2} \parallel \mathbf{\Lambda} \parallel^2, \tag{4.8}$$

and the gradient becomes

$$\frac{\partial L_r(\mathbf{\Lambda})}{\partial \lambda_k} = \frac{\partial L}{\partial \lambda_k} - \frac{\lambda_k}{\sigma^2} \tag{4.9}$$

To speed up the training process, we use stochastic gradient ascent to search for the optimal parameters [82]. At each iteration, we randomly pick a training sequence, evaluate the gradient w.r.t λ_k on that training sequence, and update λ_k by taking a small step in the direction of the negative gradient

$$\lambda_k \leftarrow \lambda_k + \alpha\Big(\sum_t f_k(\mathbf{Y}^n_{t,c}, \mathbf{X}^n) - \sum_t \sum_{\mathbf{Y}_{t,c}} p(\mathbf{Y}_{t,c}|\mathbf{X}^n) f_k(\mathbf{Y}^n_{t,c}, \mathbf{X}^n) - \frac{\lambda_k}{\sigma^2}\Big), \tag{4.10}$$

where α is a learning rate parameter, which is set to a small value. The iteration continues until we reach the maximum iteration or the change of objective function is below a threshold for 10 iterations. In our experiments, the optimization procedure usually completes within $10N$ iterations.

4.2.4.3 Inference

Two types of inference tasks are addressed during training and testing. In training, we need to compute the marginal probability of each clique $p(\mathbf{Y}_{t,c}|\mathbf{X}^n)$. In testing, we need to perform Viterbi decoding, i.e. estimate the most probable attribute sequence $\mathbf{Y}^*$ for an unseen sequence $\mathbf{X}$ that maximizes the conditional probability

$$\mathbf{Y}^* = arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}, \mathbf{\Lambda}^*) \tag{4.11}$$

where the parameter $\mathbf{\Lambda}^* = \{\lambda_k\}$ are learned from training examples. Both tasks can be achieved by Loopy Belief Propagation (LBP) [79]. In this section, we briefly discuss the LBP procedure for computing the marginal probability. The Viterbi decoding can be performed in a similar fashion by replacing the summation in Equation (4.12) with a maximization.

Belief propagation algorithms iteratively update a vector $\mathbf{m} = (m_i(v_j))$, which are called messages between pair of vertices v_i and v_j. The message $m_i(v_j)$ sent from vertex v_i to its neighbor v_j is given by:

$$m_i(v_j) \leftarrow \sum_i \left(\Upsilon(v_i)\Omega(v_i, v_j) \prod_{k \neq j} m_k(v_i) \right), \tag{4.12}$$

where $\Upsilon(v_i)$ is the local potential, $\Omega(v_i, v_j)$ the edge potential between v_i and v_j, $m_k(v_i)$ is the message sent to v_i from its neighbors except v_j. A random schedule is adopted and messages propagate through the CRF until convergence or a maximum number of iterations is reached. Then the marginal probability of nodes v_i and v_j

are computed as:

$$p(v_i, v_j) \propto \Upsilon(v_i)\Upsilon(v_j)\Omega(v_i, v_j)\prod_{k\neq j} m_k(v_i)\prod_{k\neq i} m_k(v_j). \tag{4.13}$$

Note that we have omitted $\mathbf{X}$ in the above two equations for clarity, and we use $\Omega(v_i, v_j)$ to refer either $\Psi_t(y_{t,a}, y_{t+1,a}, \mathbf{X})$ or $\Phi_t(y_{t,a}, y_{t,b}, \mathbf{X})$, which can be determined by the clique that involves v_i and v_j.

4.3 Experiments

4.3.1 Dataset and action attributes

We tested our approach on the Olympic Sports Dataset [83]. This dataset includes 16 action classes and 783 videos. The original purpose of this dataset is for recognizing action classes and thus there is no attribute labels available. Liu et al [19] defined 39 attributes on this dataset on the action class level, i.e., each action class has a list of fixed attributes across all videos of this class. As we argued in Section 4.1, this level of action attribute labels is neither sufficient to describe the dynamics in the action videos nor capture the unique characteristics within a particular video. Thus we create a new dataset to evaluate the performance of action attribute detection using the videos from the Olympic Sport Dataset. In particular, we defined 24 action attributes, which include 9 leg motion patterns, 6 arm motion patterns, 6 whole body motion patterns, and 3 human-object interactions. The full list of attributes can be found in Figure 4.3.1 and Figure 4.3.1. For each action class, we randomly select 20 videos from the Olympic Sports Dataset to create

the Action with Attribute dataset with 320 videos. In each video, we labelled the presence/absence of each action attribute as well as the bounding box of the athlete in each frame. Figure 4.3.1 illustrates the action-attribute matrix in this dataset. Figure 4.3.1 illustrates the co-occurrence matrix of action attributes. Both figures show the fill list of attributes considered in the experiments.

To evaluate the performance of action attribute detector, we divide the Action with Attribute dataset into two disjoint subsets, which include a training set with 240 videos and a testing set with 80 videos. Both sets include all 16 action classes.

4.3.2 Baseline algorithms

The first baseline algorithm is the elementary attribute detectors described in Section 4.2.3. This algorithm treats each frame as an independent sample and does not take into account any contextual constraints.

The second baseline algorithm is a linear CRF as illustrated in Figure 4.2. The node feature functions and edge feature functions are defined in Equation (4.4). The learning algorithm is the same as the one presented in Section 4.2.4.2, except that there are no between-chain feature functions involved. Since this is a chain structured CRF, exact inference can be achieved by the forward-backward algorithm and Viterbi algorithm [79]. This algorithm takes into account the temporal contextual constraints but treats each attribute independently: the semantic context is still ignored.

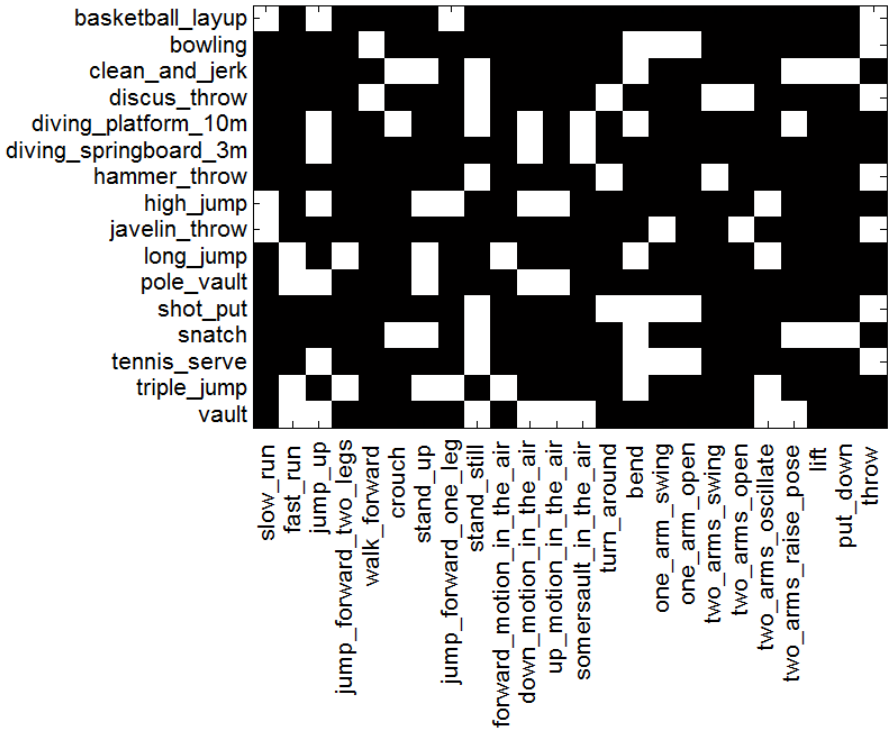


Figure 4.4: Action-attribute matrix in our dataset. White entry in row a and column b means action a has attribute b and vice versa. Notice that this figure only shows the overall relationship between actions and attributes but does not imply that *every* video of action a has (or does not have) attribute b. As discussed in Section 4.1, an attribute b may not occur in a particular video of action a.

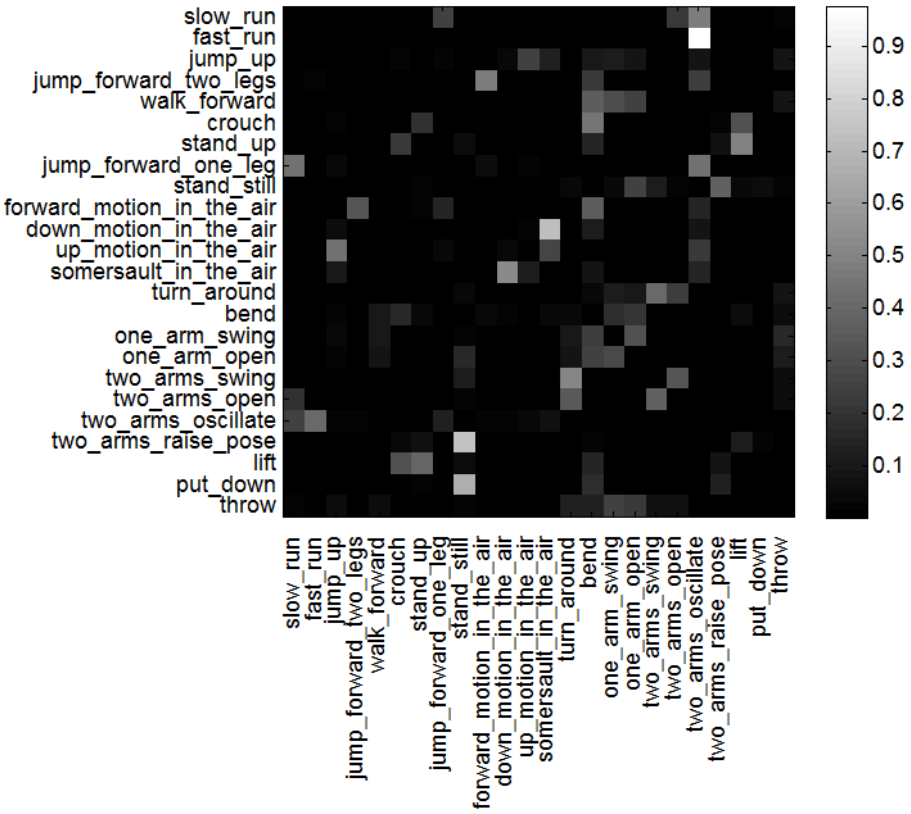


Figure 4.5: Co-occurrence matrix of action attributes obtained from training set in our dataset. The brighter an entry, the higher the probability that the corresponding attributes will co-occur at the same time point.

4.3.3 Experimental Results

We measure the performance of attribute detection by precision, recall and F1-score at the frame level. Every frame is treated as an individual sample for this purpose. A positive sample that is correctly (incorrectly) detected as positive (negative) is counted as true positive, a.k.a., TP, (false negative, a.k.a., FN); A negative sample that is correctly (incorrectly) detected as negative (positive) is counted as true negative, a.k.a., TN, (false positive, a.k.a., FP). Then the precision, recall and F1-score are defined as

$$\begin{aligned} \text{precision} &= \frac{\#TP}{\#TP + \#FP} \\ \text{recall} &= \frac{\#TP}{\#TP + \#FN} \\ \text{F1-score} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \end{aligned} \tag{4.14}$$

We compute the three measurements for each attribute and their mean and standard deviation for the baselines and the proposed factorial CRF (FCRF). Their overall detection performances are summarized in Figure 4.6. This figure shows that the proposed FCRF significantly outperforms the baseline algorithms SVM and LCRF in precision and F1-score, while the recall of all three algorithms are similar. To test the significance of these performance results, we did a t-test of the null hypothesis that the mean performances of FCRF are the same as those of SVM and LCRF, against the alternative that the mean performances of FCRF are higher than those of SVM and LCRF, with significance $\alpha = 0.05$. The result of t-test failed to reject the null hypothesis for recall but rejected the null hypothesis for precision

and F1-score, with p-value < 0.0005.

Figure 4.7 compares the precision scores across each attribute. It shows that FCRF improves the detection precision for every attribute. In particular, the baseline algorithms achieve very low precision for a few attributes, e.g. *walk forward*, *stand up*, *one arm open*, *put down* and *throw*. FCRF removes a large number false positives using contextual constraints which results in significantly better precision scores.

Figure 4.8 illustrates the outputs of the three algorithms on a video of the activity `tennis serve`. It shows that the decision values of SVM are quite noisy and there are a large number of false positives for the attribute *bend* and false negatives for the attribute *one arm swing*. The LCRF, which only considers temporal constraints, is only able to smooth out noisy detections that last for short periods but is unable to deal with persistent false positives. On the other hand, FCRF corrects many of such errors by using semantic contextual constraints: *stand still* and *bend* are less likely to co-occur while *bend* and *one arm swing* are more likely to co-occur.

Figure 4.9 shows a case where the semantic constraints in FCRF over-constrains the action attributes and results many artefacts. In the beginning of this video (until frame 60), SVM detector shows strong local evidence for *crouch* and *bend*, and they have strong semantic correlation (see Figure 4.3.1). As a result, FCRF labeled them as positive and remove other attributes that are less likely to co-occur with them, including *stand still*, which becomes false negative during this time frame.

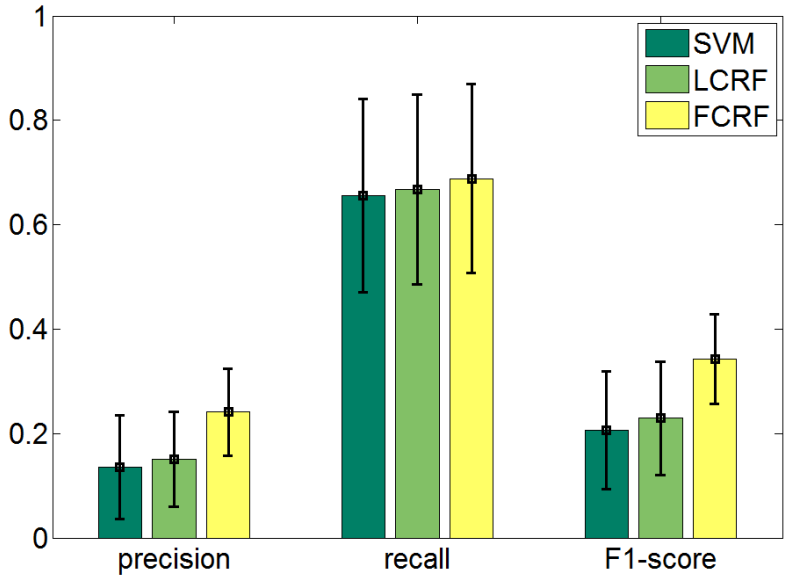


Figure 4.6: Overall performance of action attribute detection with three algorithms: SVM, linear CRF (LCRF) and factorial CRF (FCRF).

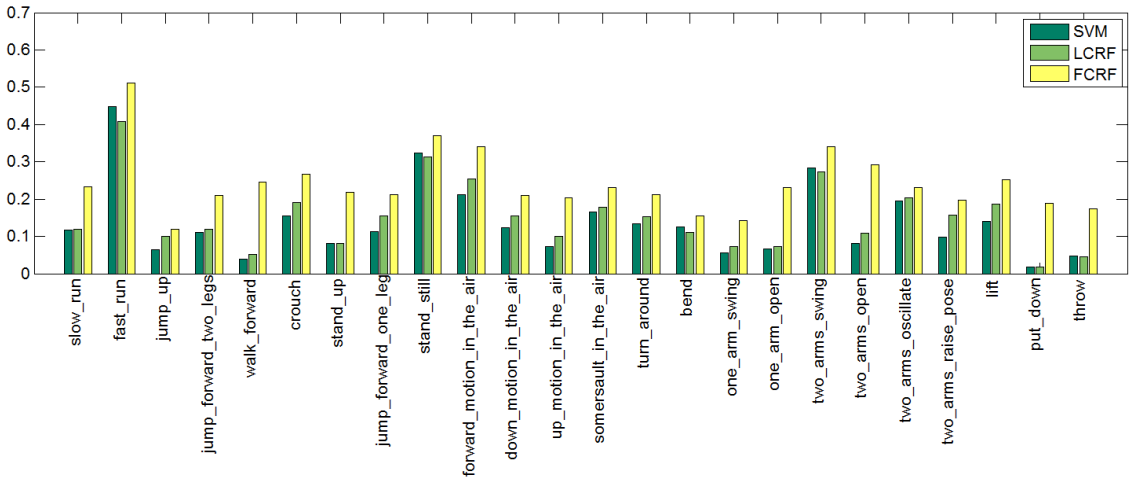


Figure 4.7: Precision of action attribute detection with three algorithms: SVM, linear CRF (LCRF) and factorial CRF (FCRF).

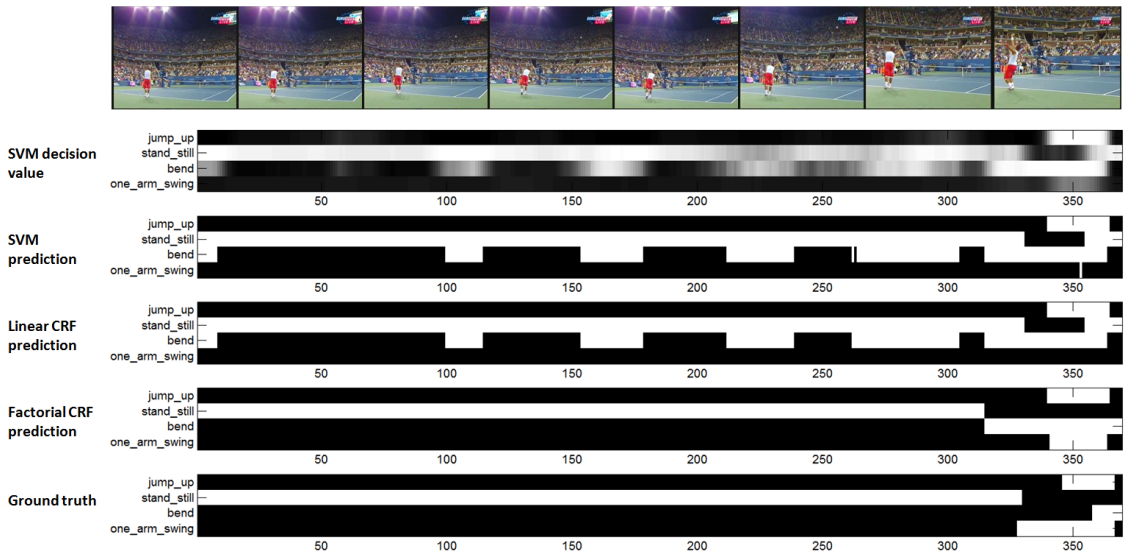


Figure 4.8: Detailed detection results of the activity `tennis serve`, where imposing contextual constraints in the FCRF improves the attribute detection precision compared to the other two baseline approaches: SVM and LCRF.

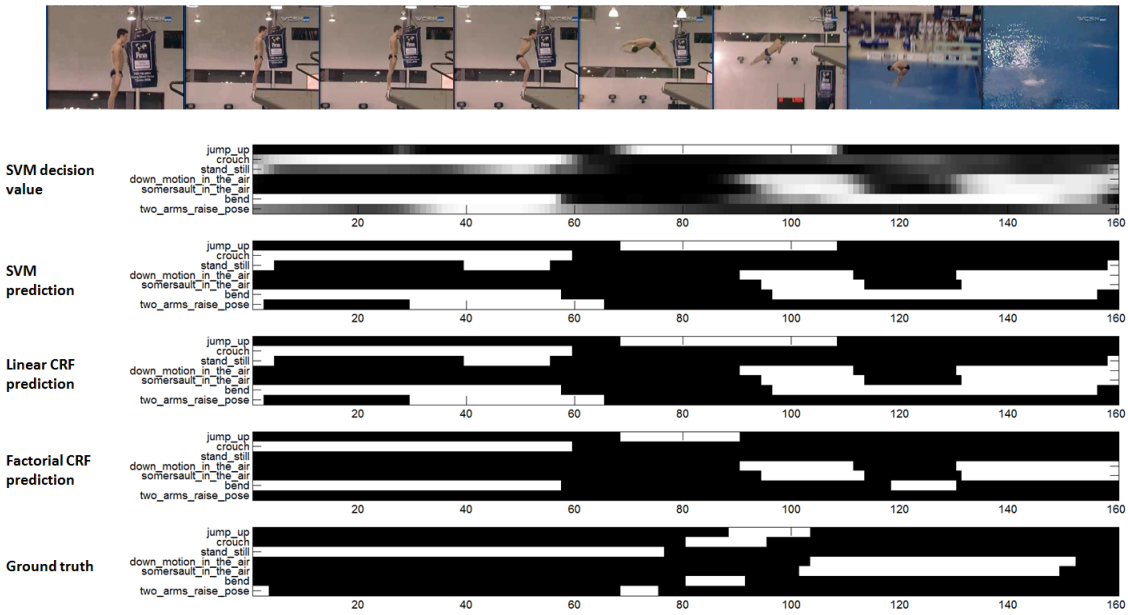


Figure 4.9: Detailed detection results of the activity `diving platform 10m`, where imposing contextual constraints in the FCRF introduces artefacts.

Chapter 5

Concluding Remarks and Future Work

As outlined in the introduction, this thesis addresses three applications of attributes, namely, describing entities, recognizing entities, transfer learning in three domains including objects, scenes and activities. Table 5.1 summarizes the work in thesis. We now provide some concluding remarks for each topic, together with directions for future research.

5.1 Attribute Based Transfer Learning

We have presented a transfer learning framework that employs object attributes to aid the learning of new categories with fewer training examples. We explore a generative model to describe the attribute-specified distributions of image features and two methods to transfer attribute prior from known categories to unknown categories. Experimental results show that the proposed approaches achieve

	Objects	Scenes	Activities
describing entities			Chapter 4
recognizing entities		Chapter 3	
transferring knowledge	Chapter 2		

Table 5.1: Contributions of this thesis.

state-of-the-art performance in both zero-shot and one-shot learning tests.

The current work has a few areas to be improved. In the Gibbs sampling procedure, each local feature is independently sampled so that the complexity of the whole system is linear with the number of local features in an image. This drawback prevents us from using large number of features in images, as did in [14]. As we have seen in the preliminary results, the features within the same superpixel tend to belong to the same topic, due to their similarity in appearance. So we can group the features within an superpixel and assign the same topic and attribute to them. By this way we reduce the complexity to be proportional to the number of superpixels instead of the number of features. Thus we can apply large number features to improve the current system.

Finally, the proposed generative model can be extended to other domain. For example, an ongoing work is to study the affordance of tools and object attributes from video streams. The affordance of tools are naturally correlated with object attributes. For example, a sharp edge affords piecing or cutting, an handle affords holding, etc. We propose to model the relationship between the affordance and the shape attributes using the Author-Topic Model, where the affordance of tools in a given video corresponds to the documents and the object attributes corresponds to the authors. We can then automatically learn the weights of each object attributes in a specific affordance from training data using this generative model. Furthermore, the proposed transfer learning approach can also be extended to this domain.

5.2 Attribute based Active Recognition

We proposed a new framework for scene recognition within the active vision paradigm. In our framework, the sensory module is guided by attentional instructions from the reasoning module and employs detectors of a small set of objects within selected regions. The attention mechanism is realized using an information theoretic approach, with the idea that every detected object should maximize the added information for scene recognition. Our framework is evaluated in a static scene dataset and shows the advantage over the passive approach. Also we discussed how it can be generalized to object recognition and dynamic scene analysis, and gave a proof of concept by implementing it for attribute based activity recognition.

In the current implementation, we have assumed that objects are independent given the scene class. Though this assumption simplifies the formulation, this is not necessarily true in general. In the future, we plan to remove this assumption and design a scene recognition model that better represents the complex scenes in the real world. Also, we will perform a comprehensive study of the proposed approach using larger image/video datasets to investigate the impact of the active paradigm.

5.3 Attribute Detection Using the Contextual Constraints

We has studied the problem of detecting action attributes from sport videos. Our work not only answers the question "is there an action attribute a in this video?", but also addresses the question "when does this attribute occur?". The attribute annotation we propose at this granularity will benefit lots of interesting

applications in video understanding and event detection. In this work, we proposed an approach that uses contextual constraints as post-processing to an off-the-shelf discriminative model for attribute detection. We observed that semantic context, i.e. the co-occurrence of attributes is an important constraint that can compensates for the ambiguity that arises from noisy video features. As a result, our approach is able to produce attribute labels that maximizes the agreement among labels at different time points and different attributes.

In our ongoing work, we are exploring more action attributes, such as the scene (e.g. *swimming pool*, *track*) and objects (e.g. *basketball/tennis ball*, *pole/javelin*). We will also incorporate more contextual constraints into our model. In particular, we are interested in the absolute and relative temporal order of attributes. An example for the absolute temporal order is in `high jump`: the athlete must first do a *slow run*, then *jump forward with one leg*, followed by *jumping up* and *moving up in the air*, and finally *move down in the air*; an example for the relative temporal order for `high jump` is that *slow run* must occur before the other attributes, *moving up in the air* must occur before *move down in the air*, etc. We believe these two types of temporal orders are both strong contextual cues that will facilitate the detection of action attributes more accurately.

5.4 Final Remarks

Though the proposed approaches was presented in the context of objects, scenes and activities respectively, they can be easily extended to other domains with

little or even no modification. For example, the generative model in the transfer learning approach for object recognition can be directly applied to scene recognition and activity recognition as long as we assume the independence among scene and activity attributes. The active scene/activity recognition approach can also be used for active object recognition without modification. The idea of using contextual constraints in detecting activity attributes can also be extended to object and scene domain after applying corresponding contextual constraints.

In summary, the attribute is a general concept that can be widely applied in the domain of objects, scenes and activities. Using attributes as a middle layer between low level features and high level entities, we have demonstrated that the proposed new paradigm has intrinsic advantages over traditional paradigm in the problems of describing entities, transfer learning, active recognition, which are crucial components towards filling the semantic gaps in image and video understanding.

Bibliography

[1] Stuart C. Shapiro, editor. *Encyclopedia of Artificial Intelligence*, chapter Image Understanding. New York: John Wiley & Sons, 1987.

[2] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based Image Retrieval at the end of the early years. *PAMI*, 22(12):1349–1380, 2000.

[3] Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny B. Braem. Basic Objects in Natural Categories. *Cognitive Psychology*, 8(3):382–439, July 1976.

[4] Lisa S Andersson, Rytis Juras, David T Ramsey, Jessica Eason-Butler, Susan Ewart, Gus Cothran, and Gabriella Lindgre. Equine Multiple Congenital Ocular Anomalies Maps to a 4.9 Megabase Interval on Horse Chromosome 6. *BMC Genetics*, (9), December 2008.

[5] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, 2010.

[6] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Describing People: A Poselet-based Approach to Attribute Classification. In *ICCV*, 2011.

[7] Sagnik Dhar, Vicente Ordonez, and Tamara L. Berg. High Level Describable Attributes for Predicting Aesthetics and Interestingness. In *CVPR*, 2011.

[8] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing Objects by Their Attributes. In *CVPR*, 2009.

[9] Yang Wang and Greg Mori. A Discriminative Latent Model of Object Classes and Attributes. In *ECCV*, 2010.

[10] Gang Wang and David Forsyth. Joint Learning of Visual Attributes, Object Classes and Visual Saliency. In *CVPR*, 2009.

[11] Dhruv Mahajan, Sundararajan Sellamanickam, Vinod NairDhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. A Joint Learning Framework for Attribute Models and Object Descriptions. In *ICCV*, 2011.

[12] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric Recognition for Cross-Category Generalization. In *CVPR*, 2010.

[13] Sung Ju Hwang, Fei Sha, and Kristen Grauman. Sharing Features between Objects and Their Attributes. In *CVPR*, 2011.

[14] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009.

[15] Aude Oliva and Antonio Torralba. Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope. *IJCV*, 42:145–175, 2001.

[16] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S.Willsky. Describing Visual Scenes Using Transformed Objects and Parts. *IJCV*, 77, 2008.

[17] Li-Jia Li, Hao Su, Yongwhan Lim, and Li Fei-Fei. Objects as Attributes for Scene Classification. In *ECCV the 1st International Workshop on Parts and Attributes*, 2010.

[18] Genevieve Patterson and James Hays. Sun Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes. In *CVPR*, 2012.

[19] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing Human Actions by Attributes. In *CVPR*, 2011.

[20] I. Biederman. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94:115–147, 1987.

[21] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-Shot Learning of Object Categories. *PAMI*, 28:594 – 611, 2006.

[22] Li Fei-Fei. Knowledge Transfer in Learning to Recognize Visual Object Classes. In *International Conference on Development and Learning*, 2006.

[23] Evgeniy Bart and Shimon Ullman. Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement. In *CVPR*, pages 672–679, 2005.

[24] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. In *CVPR*, volume 2, pages 762–769, 2004.

[25] Michael Stark, Michael Goesele, and Bernt Schiele. A Shape-Based Object Class Model for Knowledge Transfer. In *ICCV*, 2009.

[26] Kevin Murphy, Antonio Torralba, and William T. Freeman. Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes. In *NIPS*. MIT Press, 2003.

[27] Neeraj Kumar, Peter Belhumeur, and Shree Nayar. FaceTracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, pages 340–353, Oct 2008.

[28] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent Dirichlet Allocation. *JMLR*, 3, 2003.

[29] Thomas L. Griffiths and Mark Steyvers. Finding Scientific Topics. In *Proceedings of the National Academy of Sciences*, volume 101 Suppl 1, pages 5228–5235, April 2004.

[30] T. Griffiths P. Smyth M. Rosen-Zvi, C. Chemudugunta and M. Steyvers. Learning Author-Topic Models from Text Corpora. *ACM Transactions on Information System*, 2009.

[31] Josef Sivic, Bryan Russell, Alexei A. Efros, Andrew Zisserman, and Bill Freeman. Discovering Objects and Their Location in Images. In *ICCV*, pages 370–377, October 2005.

[32] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.

[33] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Learning Hierarchical Models of Scenes, Objects, and Parts. In *ICCV*, volume 2, pages 1331–1338, 2005.

[34] Li Fei-Fei and Pietro Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *CVPR*, pages 524–531, 2005.

[35] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel Gatica-Perez, and Tinne Tuytelaars. A Thousand Words in a Scene. *PAMI*, 29(9):1575–1589, 2007.

[36] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards Total Scene Understanding:Classification, Annotation and Segmentation in an Automatic Framework. In *CVPR*, 2009.

[37] Nanfei Sun, Norman Haas, Jonathan H. Connell, and Sharath Pankanti. A Model-Based Sampling and Sample Synthesis Method for Auto Identification in Computer Vision. In *IEEE Workshop on Automatic Identification Advanced Technologies*, pages 160–165, Washington, DC, USA, 2005.

[38] Dalong Jiang, Yuxiao Hu, Shuicheng Yan, Lei Zhang, HongJiang Zhang, and Wen Gao. Efficient 3D Reconstruction for Face Recognition. *Pattern Recognition*, 38(6):787–798, 2005.

[39] David G. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *IJCV*, 20:91–110, 2004.

[40] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluation of Color Descriptors for Object and Scene Recognition. In *CVPR*, 2008.

[41] Eli Shechtman Michal Irani. Matching Local Self-Similarities across Images and Videos. In *CVPR*, pages 1–8, 2007.

[42] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV*, 73(2):213–238, 2007.

[43] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active Vision. *IJCV*, 2:333–356, 1988.

[44] Ruzena Bajcsy. Active Perception. *Proceedings of the IEEE*, 76:996–1005, 1988.

[45] Dana H. Ballard. Animate Vision. *Artificial Intelligence*, 48:57–86, 1991.

[46] Jan olof Eklundh, Peter Nordlund, and Tomas Uhlin. Issues in Active Vision: Attention and Cue Integration/Selection. In *BMVC*, pages 1–12, 1996.

[47] Raymond D. Rimey and Christopher. M. Brown. Control of Selective Perception Using Bayes Nets and Decision Theory. *IJCV*, 12:173–207, 1994.

[48] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei. Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification. In *NIPS*, 2010.

[49] Nazli Ikizler-Cinbis and Stan Sclaroff. Object, Scene and Actions: Combining Multiple Features for Human Action Recognition. In *ECCV*, 2010.

[50] Steve Branson, Catherine Wah, Boris Babenko, Florian Schroff, Peter Welinder, Pietro Perona, and Serge Belongie. Visual Recognition with Humans in the Loop. In *ECCV*, 2010.

[51] Behjat Siddiquie and Abhinav Gupta. Beyond Active Noun Tagging: Modeling Contextual Interactions for Multi-Class Active Learning. In *CVPR*, 2010.

[52] Sudheendra Vijayanarasimhan and Kristen Grauman. Cost-Sensitive Active Visual Category Learning. *IJCV*, 2010.

[53] Raphael Sznitman and Bruno Jedynak. Active Testing for Face Detection and Localization. *PAMI*, 2010.

[54] Antonio Torralba. Contextual Priming for Object Detection. *IJCV*, 53(2):153–167, 2003.

[55] Marcin Marszalek and Cordelia Schmid. Semantic Hierarchies for Visual Object Recognition. In *CVPR*, 2007.

[56] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

[57] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models. *PAMI*, 32(9):1627 – 1645, 2010.

[58] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic Photo Pop-up. In *ACM SIGGRAPH*, 2005.

[59] John C. Platt. Probabilities for SV Machines. In *Advances in Large Margin Classifiers*, 1999.

[60] Myung Jin Choi, Joseph Lim, Antonio Torralba, and Alan S. Willsky. Exploiting Hierarchical Context on a Large Database of Object Categories. In *CVPR*, 2010.

[61] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluating Color Descriptors for Object and Scene Recognition. *PAMI*, 32(9):1582–1596, 2010.

[62] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[63] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive Foreground Extraction using Iterated Graph Cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[64] Ivan Laptev. On Space-Time Interest Points. *IJCV*, 64(2-3):107–123, 2005.

[65] Ce Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[66] Jingen Liu, Yang Yang, and Mubarak Shah. Learning Semantic Visual Vocabularies Using Diffusion Distance. In *CVPR*, 2009.

[67] Shiv N. Vitaladevuni, Vili Kellokumpu, and Larry S. Davis. Action Recognition using Ballistic Dynamics. In *CVPR*, 2008.

[68] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition. *PAMI*, 31, 2009.

[69] Hedvig Kjellstrom, Javier Romero, David Martinez, and Danica Kragic. Simultaneous Visual Recognition of Manipulation Actions and Manipulated Objects. In *ECCV*, 2008.

[70] Benjamin Sapp, Rizwan Chaudhry, Xiaodong Yu, Gautam Singh, Ian Perera, Francis Ferraro, Evelyne Tzoukermann, Jana Kosecka, and Jan Neumann. Recognizing Manipulation Actions in Arts and Crafts Shows Using Domain-Specific Visual and Textual Cues. In *The 3rd International Workshop on Video Event Categorization, Tagging and Retrieval for Real-World Applications (VECTaR2011), In Conjunction with ICCV 2011*, 2011.

[71] Xiaodong Yu, Cornelia Fermüller, Ching Lik Teo, Yezhou Yang, and Yiannis Aloimonos. Active Scene Recognition with Vision and Language. In *ICCV*, 2011.

[72] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in Context. In *CVPR*, 2009.

[73] Ekaterina H. Spriggs, Fernando De la Torre Frade, and Martial Hebert. Temporal Segmentation and Activity Classification from First-person Sensing. In *IEEE Workshop on Egocentric Vision, CVPR*, 2009.

[74] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *CVPR*, volume 2, pages 886–893, June 2005.

[75] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting People Using Mutually Consistent Poselet Activations. In *European Conference on Computer Vision (ECCV)*, 2010.

[76] S. M. Shahed Nejhum, Jeffrey Ho, , and Ming-Hsuan Yang. Online Visual Tracking with Histograms and Articulating Blocks. *Computer Vision and Image Understanding (CVIU)*, 114:901–914, 2010.

[77] Liam Ellis, Nicholas Dowson, Jiri Matas, and Richard Bowden. Linear Regression and Adaptive Appearance Models for Fast Simultaneous Modelling and Tracking. *IJCV*, 95:154–179, 2011.

[78] Heng Wang, Muhammad Muneeb Ullah, Alexander Klser, Ivan Laptev, and Cordelia Schmid. Evaluation of Local Spatio-temporal Features for Action Recognition. In *BMVC*, 2009.

[79] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research*, 8:693–723, 2007.

[80] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[81] Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Conditional Models for Contextual Human Motion Recognition. *CVIU*, 104:210–220, 2006.

[82] Vishwanathan Schraudolph and Schmidt Murphy. Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods. In *ICML*, 2006.

[83] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In *ECCV*, 2010.