Towards Scene Understanding: Deep and Layered Recognition and Heuristic Parsing of Objects

Dissertation Submitted to

Xi'an Jiaotong University

In partial fulfillment of the requirement

for the degree of

Doctor of Engineering Science

By

Yang Wu

(Control Science and Engineering)

Supervisor: Prof. Nanning Zheng

May 2010

Title: Towards Scene Understanding: Deep and Layered Recognition and Heuristic Parsing of Objects
Speciality: Control Science and Engineering
Applicant: Yang Wu
Supervisor: Prof. Nanning Zheng

ABSTRACT

Scene understanding is one of the most essential functionalities of human vision and also a major goal of computer vision research. Object recognition and object parsing are two critical components of scene understanding and their progresses largely determine the level of it. Due to the rapid development and popularization of cameras, a wide range of applications such as image and video search, intelligent human-computer interaction, surveillance, medical image analysis, etc. demand more and more from object recognition and parsing. Though many aspects of scene understanding in computer vision have significantly advanced in the past several decades, the recognizing and parsing of complex objects in real scenes still remain largely unsolved. The challenges are the great within-category variations of objects and the complexity of the environment due to occlusion, viewpoint, illumination, background clutter, etc. By absorbing essentials from the latest progresses in multiple disciplines including neuroscience, psychophysics, machine learning and computer vision, this thesis seeks to advance the research on object recognition in human vision by psychophysical experiments, modeling and learning deep and layered recognition in computer vision, and robust object parsing with saliency-based search.

1. Currently object recognition is driven by top-down tasks, which can output different semantics such as the category labels and the within-category attributes or states, according to the predefined semantic level of the output space. However, such a task-dependent recognition strategy hasn't taken into account the data itself and the properties of object recognition in human vision, therefore the results of it are either not deep enough or overdetermined, but not the satisfying ones adaptive to the input data. So can we break the rules to properly interpret the input data by outputting deep and layered recognition results just like what seems to be done in human vision? To support such a proposal, we collected a gallery of 3132 images with 8567 instances of two common and representative objects: car and pedestrian, and designed a group of strict psychophysical experiments to test the deep and layered recognition in human vision according to the recent progress on human rapid object recognition in psychophysics. By doing so, we built a new object recognition dataset representing the deep and layered human recognition property with annotations from as many as 20 subjects, which is named "IAIR-CarPed". The annotated results of IAIR-CarPed

The above researches were supported by The National Basic Research Program of China under Grant Nos. 2006CB708303 (advanced research) and 2007CB311005, The National High-Tech Research and Development Plan of China under Grant Nos. 2006AA01Z192 and 2005AA147060, and The National Natural Science Foundation of China under Grant No. 90820017.

show that the human rapid recognition of objects without specific visual difficulties is semantically layered, which reveals the fact that the object recognition results of humans depend on the input stimuli. This dataset and the human recognition results can serve as the first benchmark for deep and layered object recognition, and the evaluation criterion based on the human confusions between different semantics can well represent the performance of a computer vision algorithm compared to the human recognition results. Unlike other datasets, we annotated the visual difficulties separately so that it can be used to analyze the robustness of the recognition system in details and compare different systems.

2. It is a new challenge to mimic human deep and layered recognition in computers, as the output is structured and the evaluation criterion is special. Based on the latest progress in machine learning on structured prediction, a generic structured prediction model has been built for solving the problem of Deep and Layered Recognition (DLR), along with the analysis of several possible loss functions and feature representation strategies. To efficiently optimize such a structured learning problem and make it scalable to large amounts of high dimensional visual data, we present the first structured online learning algorithm (SOnline). Such an algorithm works well on the concrete deep and layered object recognition problem in the IAIR-CarPed dataset. Comparative results show the superiority of the proposed model on both deep and layered object classification and object detection compared to traditional multiclass recognition and binary recognition models. The experimental results demonstrate that DLR not only generates rich and adaptive outputs, but also improves the performance on traditional object categorization.

3. Object parsing aims at extracting the object parts and labeling their states. Existing approaches can be grouped into two categories: one is top-down template matching, and the other is sequential bottom-up perceptual grouping and top-down template matching. These two either ignore the intrinsic structure of the data or limit itself to the results generated by bottom-up grouping. Instead, we propose a novel approach called saliency-based opportunistic search which effectively fuses the bottom-up grouping and top-down matching. Such an approach optimizes both the grouping loss and the matching loss. It encourages more and better object part matching results while at the same time constrains their positions and saliency; therefore it can avoid false matching and explore bottom-up grouping gradually. Experiments on challenging statue faces demonstrate the robustness of our approach to partial occlusions, inner clutter and data defacement, and show that it generates significantly better results than the currently dominant approach using much fewer exemplars.

KEY WORDS: Scene understanding; Object recognition; Loss function; Object parsing; Heuristic search
TYPE OF DISSERTATION: Application Fundamentals

CONTENTS

List of Symbols

C	Penalty factor
K	Kernal function
Z	Normalization/Partition function
I	Image
f	Compatability function
l	Loss function
p	Propability
$\mathbf{u}$	Correspondences of control points between the image and model
$\mathbf{w}$	Model parameter vector
$\mathbf{x}$	Input data vector/Selection indicator of model segments/parts
$\mathbf{y}$	Output label vector/Selection indicator of image segments/parts
$\mathbf{z}$	Selection vector of editing operations
$\mathbf{\Phi}$	Joint input-output feature vector
β	Controlling factor
ξ_i	Slack for sample $\mathbf{x}_i$

CHAPTER 1
Introduction

Scene understanding is a major goal of computer vision, while object recognition and object parsing are two critical and fundamental problems in scene understanding [①]. This chapter introduces the research background of object recognition and parsing, and presents the research content and contributions of this thesis, followed by an overview of its structure.

1.1 Research Background

"Vision is the process of discovering from images what is presented in the world, and where it is. " said by David Marr in his book [1]. The keywords of "what" and "where" show that the ultimate goal of vision is to understand the content within the scene, though the discovering process may include many other low-level and middle-level visual computations such as sensing, filtering, grouping and abstraction. Due to its great importance and fascination, scene understanding is always of great interest to vision researchers including those from brain and neural science, psychology, cognitive science and computer science. Generally speaking, scene understanding involves both recognition (e.g. categorization) and parsing (into meaningful regions), typically at the scene level and the object level. Object recognition and parsing is of fundamental importance, because it is not only an important component of scene understanding, but also the basis of the scene-level recognition and parsing. Therefore, this thesis focuses on promoting the research on it.

Object recognition and parsing has a long history of at least 50 years [2], and it has been greatly developed in the last two decades due to the rapid advances in both academics and industry. In the academic perspective, as shown in Figure 1-1, new progresses in three different research areas have significantly influenced the development of the research on object recognition and parsing. In neuroscience and psychophysics, more and more publications are on or closely related to human vision or primate vision on object recognition, among which the work done by Tomaso Poggio's team in MIT on a hierarchical computational model for feedforward rapid recognition [3–5] has the most significant influence, which is a successful case on modeling new findings in neuroscience and psychophysics for machine recognition. In machine learning, numerous learning

[①] The "object recognition" mentioned in this thesis is a general term which may refer to one or more specific visual object recognition problems including object presence vs. absence classification, object localization, object detection, object categorization, within-category object classification and object identification, while "object parsing" means to extract the semantic parts of the objects and label them.

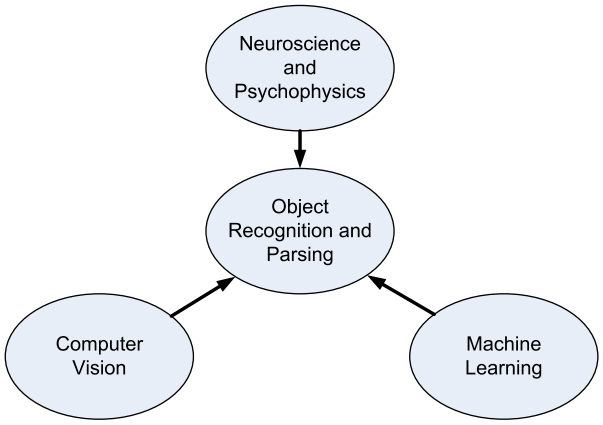

Figure 1-1 The progress of object recognition and parsing mainly dues to the advances in three different disciplines. The work presented in this thesis absorbs essentials from the recent advances in all these three disciplines and tries to promote the research in all these three directions.

algorithms have been used for object recognition, including generative methods such as Bayesian approaches, graphical models and many manifold learning algorithms, and discriminative methods such as decision trees, boosting and support vector machines (SVMs). Recently, the progress in structured prediction and inference [6] provides new tools for object recognition with complex output structures, e.g. localization [7]. In computer vision, a lot of powerful visual features (such as SIFT (Scale Invariant Feature Transform) [8], HOG (Histogram of Gradients)[9], Shape Context [10], etc.) and recognition models (such as bag of words (BoW) and part-based models, etc.) have greatly encouraged the research efforts. The research work presented in this thesis also largely depends on all of these progresses. On the industry side, the rapid development of working conditions including the hardware (cameras, computers, etc.), software and the network (especially the Internet) have greatly accelerated the research, especially towards large scale computation.

Besides of that, object recognition and parsing has a wide range of applications as follows, which have been driving the research towards solving practical problems.

1. **Image and video search**. There are billions of images and videos on the Internet, how to efficiently index and search them becomes an important practical issue. Though currently the public search engines are still based on text search, many researches are going on towards fast and robust content-based image and video search. However, there is still a long way to go for using object recognition in multimedia data search as the problem itself remains largely unsolved. Nevertheless, great efforts have been put into it. A notable mention is the construction of several large scale object recognition databases like 80 million tiny images [11] and ImageNet [12]. It is foreseeable that any significant progress in object recognition will greatly influence the image and video search.

2. **Intelligent vehicle systems**. Recently, intelligent vehicles have been extensively researched with the goal of increasing drive safety, improving operational efficiency and enhancing drive experience. To achieve these goals, many object recognition tasks may be involved, such as road sign recognition, lane marking recognition, traffic light recognition, vehicle detection and the most important pedestrian detection. As more and more competitions have been held for driverless vehicles in different countries such as the DARPA Urban Challenge in the United States and the ongoing annual "Future Challenge" of intelligent vehicles in China, the research on object recognition for intelligent vehicles gathers more and more attention. Meanwhile, two carefully annotated datasets on the real urban traffics captured from a driver's view have been published recently [13, 14], which are expected to be able to drive the research on object recognition especially pedestrian detection towards real applications.

3. **Robots**. Object recognition is an important aspect of robot intelligence. Though the difficulty of object recognition itself has limited its application in robotics, there are still many successfully cases and valuable trials. In the 1980s, robots were able to recognize and pick up regular objects like blocks and ensemble them, such researches result in applications in bin picking tasks. There are many reported progresses of robots on recognizing specific objects like faces and hand gestures, including those on the most famous robot ASIMO. The STanford Artificial Intelligence Robot (STAIR) team has designed robots that can pick plates, grasp staplers, and open doors based on automatic recognition of these objects [15]. Recently, a challenge named "The Semantic Robot Vision Challenge" has been held annually to advance the semantic object recognition in robotics[①]. Object recognition has also been used in robot localization and navigation[16], and service robots [17].

4. **Security and surveillance**. Biometric identification is an important application of specific object recognition in public security, which includes the recognition of many biological features such as fingerprints, irises, gaits, faces, and so on. Among them, face recognition is most widely researched[18]. In surveillance, human recognition and human activity recognition are very important, in which object recognition and parsing plays a critical role.

5. **Medical image analysis**. Rapid advances in medical imaging technology have dramatically increased the amount of medical image data generated daily by hospitals, pharmaceutical companies, and academic medical research[19], therefore automatic analysis of such data for both scientific research and clinical analysis becomes an urgency. Object recognition techniques have been widely used for such a purpose. Most of the applications are on the detection of particular objects, such as anatomical regions [20], disease areas [21], and functional organs[22]. Besides that,

[①] www.semantic-robot-vision-challenge.org

some others have also worked on generic object recognition and image search in medical images[19]. Due to its specificity, object recognition in medical images usually incorporates certain domain knowledge about the objects.

6. **Other applications**. Object recognition and parsing has also been used in many other applications including optical or handwritten character recognition, industrial inspection, human computer interaction, entertainments (e.g. games and movies), etc.

Currently, object recognition and parsing has become the hottest research topic in computer vision, which owns the largest number of publication in major vision conferences. Even though, only some specific object recognition problems under controlled conditions have got satisfying results, while most of the others are still far from being solved, for example pedestrian/human detection and generic object categorization. There are many challenges which have made the problem hard to solve in real applications: viewpoint and pose changes, within-class shape and appearance variations, illumination, occlusion, and background clutter. However, the human vision system has great robustness to them and it can recognize thousands of object categories rapidly without much effort. Therefore, it is still worthwhile to learn from the human vision system and find new ways to advance the research on object recognition and parsing in computer vision.

1.2 Research Contents and Contributions

Instead of trying to solve the big problem of scene understanding directly as shown in Figure 1-2, this thesis focuses on two critical and fundamental components of it: object recognition and object parsing. Object recognition seeks to interpret the objects themselves while object parsing goes down to explain the semantic parts of the objects. Despite their different focuses, these two problems are highly correlated. Part-level parsing results are precious middle-level representations for robust object-level recognition invariant to within-class appearance changes, while object-level recognition results may serve as global constraints for part extraction and labeling. Though a tightly-coupled co-optimization of these two is not involved in this thesis, the research on each of them provides a valuable foundation for integrating them towards deep scene understanding in the future.

Object recognition has been widely researched in the past five decades, and a lot of work has been done in different ways to promote it, such as building new datasets, proposing novel features, designing better models, exploring different machine learning techniques, and so on. Even though, the generic object recognition in real scenes remains largely unsolved. Instead of working on the existing datasets and trying to improve the performances of them by proposing new algorithms, we take a step back to begin with

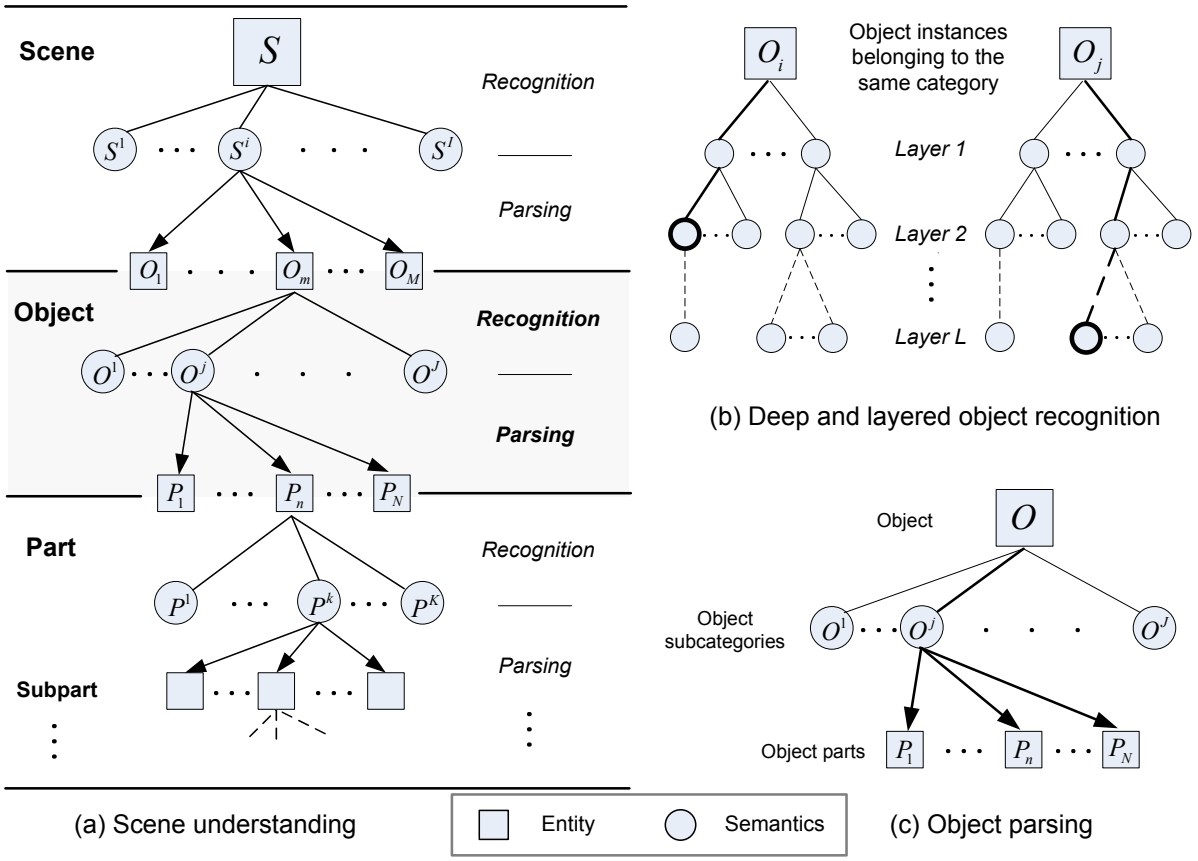

Figure 1-2 The research contents of this thesis in the context of scene understanding. (a) A detailed scene understanding framework which can be viewed as an interlaced process of recognition and parsing of entities at different semantic levels. (b) and (c) are the two research focuses of this thesis which aim at solving the two critical and fundamental components of scene understanding at the object level, and they can be potentially extended to other semantic levels. The visual content of the scene may vary a lot due to both the intra-class variations of the objects and the changes of the environment. Instead of doing simple object categorization or within-category classification disregarding these variations, we propose to interpret the objects properly based on their visual appearance, i.e. dynamic deep and layered object recognition representing the scene content variations as shown in (b). Object instances belonging to the same semantic category may be interpreted into different within-category semantic layers constructed by the attributes or states of the objects. (c) presents a typical object parsing problem with the help of object subcategorization. This thesis proposes a new robust approach for solving this problem.

comparing the way we do object recognition in computer vision with the way humans recognize objects. We found that the deep and layered interpretations of object instances belonging to the same category not only enrich human recognition results, but also help people to remember objects of great within-category variations. Inspired by this, we focus on the problem of deep and layered recognition by learning how people recognize objects and designing computer vision models and algorithms to mimic it.

Compared to object recognition, object parsing is more constrained. Usually the

part model of the object is given, and the goal is to match each object part to the image. The difficulty is the variation of the model (such as viewpoint changes and deformation, etc.) and the visual difficulties within the data (such as illumination changes, image clutters, occlusions, low contrast, etc.). Traditional approaches either concentrate on top-down model based matching or sequentially do bottom-up grouping and top-down matching, so that they are sensitive to visual difficulties especially image clutters. Our research focuses on finding a new robust approach for parsing objects with partial occlusions, heavy clutters and data defacement.

Specifically, this thesis makes four important contributions.

1. **A novel problem called *deep and layered object recognition* for adaptively interpreting the objects within a scene, and a new benchmark dataset (IAIR-CarPed) built for the research on it.** To better understand object recognition, two critical properties of human vision on object recognition have been researched recently: the deep within-category interpretation and the layered categorization. However, currently these two promising trends are still separate. Deep non-categorial semantics are treated as flat multiclass labels while layered semantics are restricted to object categories. Inspired by human visual experiences, we propose to do flexible deep and layered recognition of objects, even when they are of the same category and in the same scene. To support such a proposal, we built the "IAIR-CarPed" dataset by having 20 subjects to recognize 8567 cars and pedestrians in 3132 images via a strict psychophysical experiment. The human recognition results well prove this proposal and provide valuable annotations for mimicking the deep and layered recognition in computer vision. The layered semantics are on the specificity of the object orientation and the clearness of the key object part. We propose an evaluation criterion called confusion loss for this new recognition problem based on the statistical human confusions on these semantics. We are making this novel dataset along with the psychophysical annotations publicly available for advancing the research on it.

2. **A generic formulation for the deep and layered object recognition (DLR) problem, with detailed analyses on modeling, learning, evaluating and implementing it; a case study on recognizing cars and pedestrians in IAIR-CarPed dataset demonstrating the superiority of the proposed model against traditional recognition models.** We present a general definition of DLR which can be modeled as a structured prediction problem. A discriminative max-margin based learning model is recommended for solving it, in which the loss function represents the output structure and also suggests proper evaluation criterion. Different loss functions have been discussed for both categorization and localization in deep and layered recognition. Moreover, two key issues on feature representation have been analysed for implementing the strategy. We instantiate

this generic strategy for deep and layered recognition of cars and pedestrians in IAIR-CarPed dataset and show that DLR is superior to multiclass recognition and binary classification strategies on object recognition tasks.

3. **A discriminative structured online learning algorithm (*SOnline*) which can learn a structured prediction model efficiently and incrementally.** Though the max-margin based structured learning formulation of DLR can be solved using off-the-shelf tools like SVMStruct [23], in practise these tools have two major drawbacks for our application. One is that the training is usually very slow when there are many training examples with a high dimensionality, and the other is that the object localization in DLR needs recursive bootstrapping which results in expensive retraining when new non-object examples are mined. In contrast, the proposed SOnline algorithm, which is extended from a successful multi-class classification algorithm LaRank[24], is so far the first structured online learning algorithm which can get near optimal solution quickly, and it updates the model incrementally thus needs no retraining. We present details on using this algorithm for deep and layered recognition. Potentially, this algorithm can be used for many other structured prediction problems with large amounts of training data, especially those online learning and prediction applications.

4. **A new approach named *saliency based opportunistic search* fusing bottom-up grouping and top-down matching for object parsing robust to partial occlusion, inner clutter and data defacement.** Object parsing is very important for understanding the objects within a scene, as it can reveal the states of the objects by reasoning about the configurations of their parts. Existing methods either do pure top-down template matching with the part-level geometric context which is sensitive to image clutter or do unsupervised bottom-up segment grouping and top-down template matching sequentially whose performance is limited by the grouping results. We propose a saliency based opportunistic search to fuse the bottom-up grouping and top-down matching so that bottom-up grouping can be guided by top-down matching while at the same time provides it with high quality salient image segments. Experiments on challenging statue faces demonstrate the robustness of our approach compared to the currently dominant approach.

1.3 Arrangements

Focusing on object recognition and parsing with the ultimate goal of deep scene understanding, this thesis presents the research work as follows.

Chapter 2 is an overview of the literature on the topic of object recognition and parsing. It starts with a brief review of its half-century history by partitioning it into three typical eras so that the evolution trace can be clearly identified. Then the represen-

tative research progress on both representation and computation is stated and analyzed respectively. Finally, four core and unsolved issues are discussed along with the state-of-the-art research efforts on them.

Chapter 3 introduces a brand new benchmark dataset named "IAIR-CarPed" which proposes a novel research problem: deep and layered object recognition. First, the reasons for building this dataset are given after analyzing the current trends on dataset building. Then the details on the construction and annotation of this dataset are stated. Unlike other datasets, the annotation is done by a psychophysical experiment with 20 participants, so the setup and the process of the experiment are described thoroughly. The statistics of the dataset are given for revealing the content and properties of the data which can help the designing of experiments and also the recognition models and methods. Finally, three typical applications of this dataset are discussed and some primary experiments have been done on the first two of them with evaluation methods and baseline experimental results.

Chapter 4 focuses on formulating and solving the deep and layered recognition problem proposed in Chapter 3. The generic definition of DLR is given in the beginning, followed by its modeling and learning methods. The evaluation of this problem and the important feature representation issue are discussed thereafter. Specifically, an instantiation of DLR for recognizing cars and pedestrians in the IAIR-CarPed dataset is presented. Experimental results show that the proposed model with human confusion losses is superior to other recognition models on deep and layered object recognition, within-category classification and the category-level object detection. Moreover, the effectiveness of the adopted features, the overall robustness of the algorithm and its complexity are discussed.

Chapter 5 proposes a new approach named saliency based opportunistic search for robust object parsing. The object parsing problem is conditioned on the object recognition problem presented in Chapter 3 and 4 which selects the right model for parsing when the object cannot be represented by a single model. After presenting the motivation and related work, the key idea of the proposed approach on fusing the bottom-up grouping and top-down matching is explained in details. Then a case study on face parsing is given with implementation details. Finally, experimental results demonstrate that the proposed approach significantly outperforms the dominant approach on parsing faces with heavy clutters and data defacement.

Chapter 6 concludes the research work presented in this thesis, and points out possible extensions for future research.

The relationship between different chapters are shown in Figure 1-3.

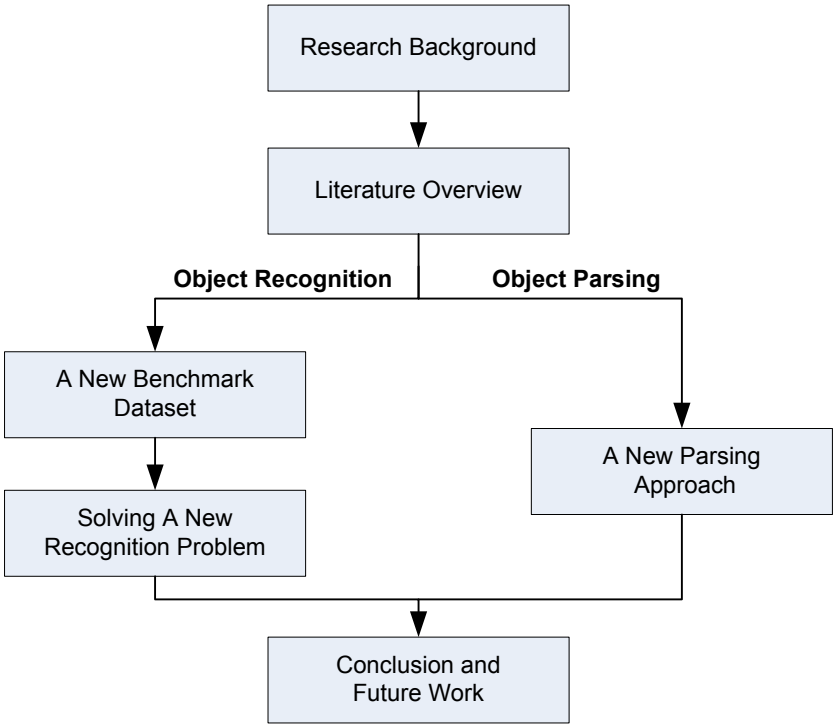

Figure 1-3 The relationship between the chapters.

CHAPTER 2
Object Recognition and Parsing: An Overview

Object recognition and parsing has a long history of about 50 years, and it is currently a very active research topic in computer vision. Due to its wide applications and fundamental difficulties related to all levels of vision problems, such a high-level vision problem has been researched a lot in many different ways, resulting in a huge literature. This chapter presents an overview of it, which is organized as follows. First, the history is briefly reviewed by tracing the evolutionary process. Then we state the major progresses of the two main components of the research topic (representation and computation). Finally, some core and unsolved issues along with the state-of-the-art efforts on them are discussed.

2.1 A Brief History

The research on visual object recognition dates back to the 1960's, when people started working on the perception of 3-D solid objects like blocks [25]. At that time, these objects are restricted to contrived textureless objects with clean backgrounds, and therefore the research was focused on the shape and geometry of objects. Since the beginning of 1990's, the focus has been clearly shifted to textured objects, in which appearance plays an important role. To build a robust appearance model of the interested object, the algorithms are usually learning-based, firstly on exemplars and then on categories, and the representation changed from global features to local ones which are robust to cluttered backgrounds. Following this path, the robustness and generalization ability of object recognition algorithms are growing rapidly as shown in Figure 2-1. We categorize the history into three different eras and briefly review the representative researches of each of them as follows.

2.1.1 The Geometric Era

From 1950's to early 1990's, geometric representations have dominated the research on object recognition [2, 26]. One of the earliest work used the moment-based geometric invariants to describe the characters for recognition [27], while the others mainly focused on geometric measurements using real geometric elements of objects, such as corners, lines, planes and spheres for regular man-made objects and generalized cylinders for natural objects with curved shapes.

Projecting the whole objects *(3-D to 2-D projection).* In the late 1960's and early 1970's, *the blocks world* was the main stream of the research, in which objects are

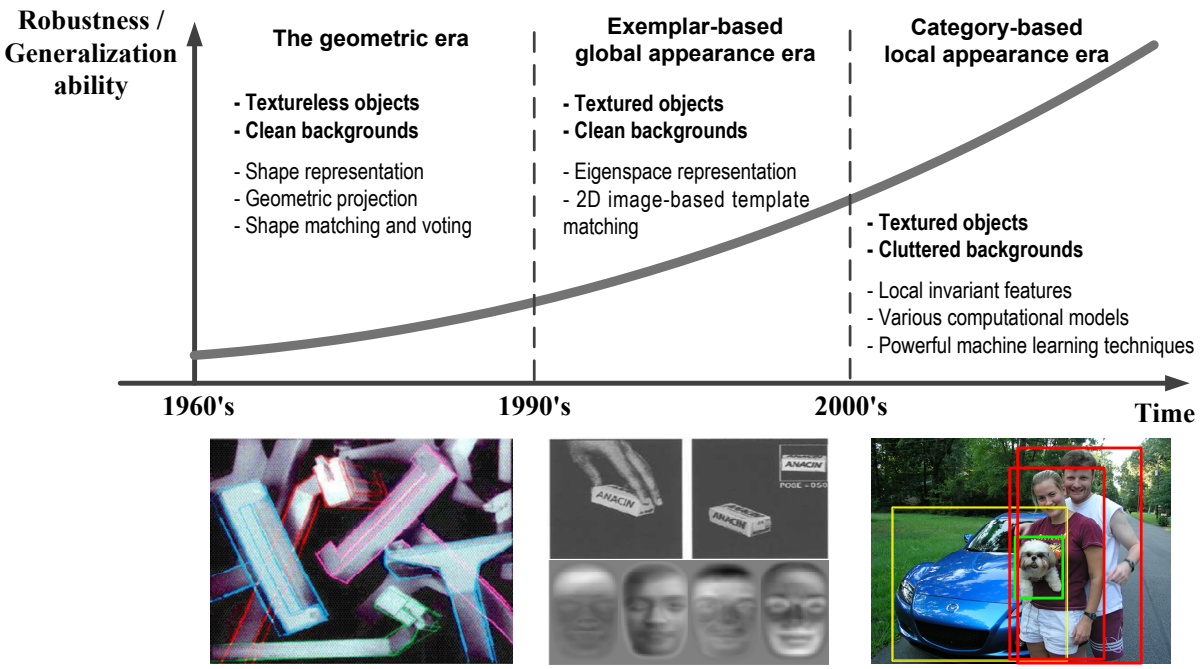

Figure 2-1 Visual object recognition in the past five decades. Generally speaking, the research can be roughly categorized into three eras, and there is a clear trend that the data is becoming harder and harder while the representation and modeling is more and more flexible. Thus the robustness and generalization ability of the algorithms are growing rapidly.

restricted to polyhedral shapes and the background is uniform. The goal is to recognize polyhedral shapes in 2-D images, which may be arbitrarily placed in the image with occlusions, using the 3-D models of these shapes. The most significant work on it was done by L. G. Roberts [25], who had detailed research on the problem of projecting polyhedra into the perspective images.

Decomposing and constructing curved objects *(2-D to 2-D, and 3-D to 3-D recognition).* An extension of the blocks world is the work on line drawings extensively researched by Guzman[28]. It goes beyond the regular polyhedra to curved objects with curved parts. Guzman proposed to use a set of hand drawn parts for representing generic objects, while taking into account of the contextual relationships between these parts. Another significant advance is the generalized cylinders (GC) proposed by Thomas Binford in 1971 [29], and then further developed by his students Gerald Agin [30] and Ram Nevatia [31, 32]. Unlike the line drawings which is 2-D, the generalized cylinders are generated automatically from 3-D range data and the recognition is usually done in 3-D directly. Later on GC was also been used by Nevatia to recognize simple objects like coffee pots in 2-D images [33].

Hypothesizing objects by model-based matching *(mainly 3-D to 2-D projection).* From the beginning of 1980's, people start to deal with the recognition of real objects (not blocks anymore) in 2-D images with significant illumination changes and occlusions, for

example the plastic razors recognized by David Lowe's SCERPO [①] system [34]. The main stream was to hypothesize the 2D projection of the 3D object exemplars by local image features (such as corners, lines, etc.), and the hypotheses are generated by checking the consistency of the features respect to the projection determined by a minimum feature set (e.g. the geometric transform of three points can define an affine projection). Such methods are referred to as minimum feature set matching [2], a typical example of which is geometric hashing[35]. Unlike the blocks world which is trying to build a generic object representation using regular blocks, the minimum feature set matching mainly focuses on specific 3-D models capturing the exact shape of an object, namely, the recognition shifted from category recognition to exemplar recognition. Among many successful real applications of it, Joseph Mundy and Dan Thompson's work on vertex-pair constrained recognition showed encouraging results on detecting aircrafts at airfields which have bland backgrounds and limited occlusions [36].

Achievements and Limitations. The research on geometric object representation for recognition has several achievements including:

- Convinced that shape is important for recognizing objects especially man-made ones.
- Developed several shape representation approaches and 3-D shape to 2-D image projection and matching techniques, and demonstrated their advantage of being invariant to viewpoint changes.
- Showed the power of distributed representations of objects using sharable parts and the relationships between them.
- Provided successful applications in some restricted tasks.

While at the same time it suffers from some critical drawbacks which have limited its further development:

- The lack of reliable image segmentation and feature extraction methods.
- Its high demands on model construction.
- Its inability to deal with deformable objects with textures.

At the beginning of the 1990's, a group of outstanding researchers enthusiastically looked into the problem of finding geometric invariance for modeling and recognizing general objects but soon get defeated by two facts [2]: a) it was proved independently by several researchers that no viewpoint invariants exist for general 3-d shapes and b) the feature grouping problem is unsolved. While at the same time, faster machines and cheaper cameras made possible the recognition using dense appearance based methods which works on pixels directly without suffering from the error-prone image segmentation

[①] Spatial Correspondence, Evidential Reasoning, and Perceptual Organization.

and the demanding modeling for geometric description. Due to these reasons, a new era of using appearance came.

2.1.2 Exemplar-based Global Appearance Era

The research on using object appearance for recognition starts from 1990 on human face recognition using eigen-functions [37] and then the eigenfaces [38]. From then on, the strategy of using the raw images with dimension reduction techniques like eigenspace decomposition has dominated the research until the beginning of 2000's. Such a movement has a clear characteristic: using the global appearance as the representation of objects and the recognition is exemplar-based. In another word, the recognition is mainly about object identification using 2-D image templates learned from the exemplars. Despite its simplicity, recognition systems constructed using this strategy were able to recognize arbitrarily complex objects with texture and surface markings, which is a significant advance over geometric methods.

Murase and Nayar's 3-D object recognition and pose estimation system [39] based on eigenspace representation and nearest neighbor search in the low-dimensional manifold of the objects showed very good results on 20 objects with complex shape and reflectance properties in the COIL-100 database[40]. Murase and Nayar's work generated tremendous interest, overshadowing ongoing recognition research based on geometry [2]. After that, some researchers have worked on improving the ability of the eigenspace based approaches, such as making it capable to deal with occlusions [41, 42], associating it with density estimation and maximum-likelihood estimation [43], building a hierarchical representation by exploring object parts and their relationships [44] and applying it to the visual tracking of articulated objects [45]. Even though, the global appearance based methods are still too rigid to generalize to object categories with large intra-class transformations and appearance variations.

2.1.3 Category-based Local Appearance Era

Since 2000, a clear trend has driven the research to object categorization based on local features. At the beginning of 2000's, a lot of local invariant features have been proposed and they soon got great successes in object recognition when combined with the simple bag of words model and the later part-based models with pair-wise geometric constraints. From then on, local appearance based methods have dominated the research. In the following sections, major progresses of the category-based local appearance era will be reviewed in details, so they are not mentioned here.

2.2 Representation

Though generally representation includes input data representation, output space representation and also certain recognition models which bridge these two, this section narrows the concept to input data representation only, leaving the others to be discussed elsewhere. As can be seen from the historic review in 2.1, data representation plays a central role in the evolution of object recognition. Usually, data representation is also referred to as feature representation when the concept of feature means arbitrary mapping of the data. We review only the features that have been used in the past two decades for the purpose of object recognition and parsing by roughly dividing them into three groups: local features, global features, and combined local and global features. Due to the richness of the literature, presenting a complete list of the all the features is almost intractable and also unnecessary. Instead, only the representative features which have significant impacts are mentioned, while discussions on the general properties of these types of features are given.

Notes on Terminology A recent survey on local invariant feature detectors [46] defines a local feature as "an image pattern which differs from its immediate neighborhood" which concerns on only the local features representing image changes. To cover all possible instantiations actually been used, a more intuitive definition is proposed here: *a local feature is a representation or mapping of a part of the visual data (e.g. a region/patch of the image or object) which has a restricted extent.* In contrast, a global feature means a holistic representation of all the visual data, i.e. the whole image or object. Note that a global representation based on local features may be referred to as global features in the literature. To clarify the concepts, we propose a simple strategy based on the feature extraction/mapping function to differentiate local and global features: if each element of the feature is a function of all the visual data then the feature is global, and if all the elements are functions of local areas of the data then the feature is local, leaving the mixed cases as combined local and global features.

2.2.1 Local Features

The power of local features Local features have in fact become the standard data representation for most object recognition and parsing tasks. The powerfulness of such features is ensured by their great properties including the followings.

- **Flexibility and Richness**. Their amounts, locations, spatial extents, types and structures are all flexible. Therefore in general there are large amounts of local features, and one can easily design new ones for his/her needs.
- **Invariance and Robustness**. The representation flexibility of local features makes them possible to be invariant to image transformations and viewpoint changes,

while their locality ensures a certain degree of robustness to occlusion, nonuniform illumination and object deformations.

- **Versatility and Controllability**. Many of the outstanding local features are designed to be task independent so that they can be used in various applications. One can choose his/her own way of using them by tuning the free parameters, and their competing properties can be balanced for different scenarios, such as distinctiveness versus invariance/robustness, localization accuracy versus robustness, repeatability/stability versus quantity, quantity/richness versus efficiency, etc.

Despite their great variability, local features are usually constructed through two steps: feature localization and feature description, where localization means finding the locations for each local feature while description stands for the data representation on these locations. We briefly categorize the strategies for each one of them and discuss the feature construction problem for different applications.

Feature localization strategies. Feature localization is to decide where the features should come from. For some applications based on image matching, the repeatability, distinctiveness and localization accuracy of these locations are very important so that the problem is usually referred to as feature detection, i.e. to detect the locations with such properties. For the problem of object recognition, however, feature locations are not that important as the ultimate goal is to represent the data for better recognition performance. Common strategies for feature localization aimed for object recognition and parsing are the following three.

- *Uniform sampling.* Uniform sampling means sampling the locations from the image exhaustively without selection. Depending on the scenario, it may be referred to as dense sampling as in the case of extracting features from every possible location [47, 48], or grid sampling when the locations are placed on a fixed grid [9]. Such a strategy is very simple but may be inefficient when the number of locations is very large, and as the features are blindly placed, a feature selection or weighting step is needed after the constructing the raw features.

- *Biased sampling.* Instead of placing the features blindly, one may choose to sample them based on certain priors or cues, such as putting them along the edges [49, 50] and using a bottom-up saliency map to guide the sampling[51].

- *Interest point detection.* Numerous detectors have been proposed for extracting interest points with different levels of invariance, such as rotation invariant detectors (e.g. Harris corner detector [52], Hessian blob detector[53], and SUSAN corner detector[54]), scale invariant detectors (e.g. Harris-Laplace [55], Hessian-Laplace [56], DoG [57] and SURF [58]), affine invariant detectors (e.g. Harris-Affine [55], Hessian-Affine[55], and Salient Regions [59]), and perspective invariant detectors (e.g. MSER [60] and Superpixels[61]). The first three groups represent corners and blobs while

the last one represents regions. Besides the level of invariance, their properties and computational efficiency also vary a lot. Detailed descriptions and comparisons of them can be found in Tuytelaars and Mikolajczyk's survey on local invariant feature detectors [46] and an earlier comparison presented by Mikolajczyk et al. [62] on affine region detectors only. Locations proposed by the invariant feature detectors have two major advantages: a) being sparse, distinctive (informative), and stable, and b) the information of the characteristic scale and affine transformation parameters can be used for invariant feature description.

Feature description strategies. There are mainly three different strategies for feature description:

- *Quantization and histograming.* This is the most popular feature description strategy due to its simplicity, flexibility and representation ability. One can quantize different data such as the intensity values (e.g. spin image [63]), color values (e.g. color histograms [64]), gradient orientations (e.g. Shape Context [10], SIFT [8], GLOH [65], and HOG [9]), etc. Then a histogram can be build upon the image space and the quantized value space. Spatially, grid histograms and log-polar histograms are two widely used ones. One may normalize the histogram globally like SIFT or locally like HOG. The quantization and histograming make the feature description invariant to spatial shifts and slight variations within the space of the quantized value.

- *Filtering.* In general any filters are applicable as long as they are able to generate effective features. Typical ones are wavelets (e.g. haar-like features [47], Gabor filters [66]), differential invariants proposed by Florack et al. [67], steerable filters developed by Freeman and Adelson [68] and complex filters introduced by Schaffalitzky and Zisserman [69].

- *Computing spatial statistics.* Van Gool et al. [70] proposed a set of generalize moment invariants to describe the multispectral nature of the image data. Though it can generate moments of any order and degree, moments with high order and degree are two sensitive to small variations of the image data. Therefore usually only up to the second order and the second degree are taken into account, while all the color channels are described independently [65].

Feature construction. As mentioned before, a local feature representation is a combination of feature localization and description. To construct proper local features for certain application, one has to choose the right strategy for each of these two steps. In object recognition tasks where the object candidates are well aligned (including object detection based the exhaustive sliding window approach), uniform sampling along with histograming (e.g. HOG features) or filtering (e.g. wavelets) can generate good results using feature weighting and selection learning tools like SVM and Boosting [9, 47]. For

object categorization or presence/absence classification when the objects are not well aligned, extracting local invariant features like SIFT on interest points is a plausible choice and has been proved to be effective when combined with Bag-of-words (BoW) model [71], and biased sampling upon the interest points (a combination of the two strategies) can boost the performance a bit [51]. About choosing the concrete detectors and descriptors for constructing local invariant features aimed for object recognition, Hessian-Laplace, Hessian-Affine and MSER are good detectors while SIFT, GLOH and Shape Context are more likely better than other descriptors. Such a judgement has been proved by two recent performance comparison papers [72, 73] and many other experiments in the literature. Note that the setting of the free parameters for both localization and description may significant influence the recognition performance [9], and one should balance the invariance and discriminability of the features based on the concrete problem he/she is working on.

2.2.2 Global Features

Global features provide a holistic representation of an image or object. Generally, the scale or size of an object is one of the simplest global features. Global transforms like Fourier transform has been introduced for representing object shapes in the early ages. The most important global features are those extracted by dimension reduction approaches such as Principle Component Analysis(PCA), Independent Component Analysis(ICA), and numerous manifold learning methods proposed in 2000's, for example ISOMAP, LLE, and Laplacian Eigenmaps[74]. These approaches aim at finding a low-dimensional embedding of the training data which presents the intrinsic geometric structures of the data so that the objects can be better classified in the embedding space. These methods have mostly been tested on face recognition datasets and also some character recognition datasets [74], in which the data contains only well-aligned objects without background clutter and occlusions. Another global feature for representing the whole image is the gist feature proposed by Torralba et al.[75] for scene understanding, which can be used in context modeling for object recognition.

Global features have the advantages of being able to capture the holistic information of the object/image and usually being computationally efficient, but they are weak for representing objects with significant background clutters, occlusions and intra-class variations. Even though, global features may be good complementary features to local ones (as discussed in the next subsection), and the approaches for extracting global features may be used to compress the high dimensional local features for efficient classifier learning without losing the discriminative power (e.g. PCA-SIFT [76] or PCA-HOG [77]).

2.2.3 Combined Local and Global Features

There are evidences in psychophysics and neurophysiology that both global and local features are crucial for face recognition [18, 78], therefore many people have looked into the problem of combining local and global features for improving face recognition results. To name just a few: Gao et al. [79] fused the results of Adaboost on multi-scale and multiorientation Gabor features and global features generated by Linear Discriminant Analysis (LDA); Huang et al. [80]also used local Gabor features, but fused three types of global features(eigenface, spectroface, and ICA); Chen et al. [81] used Gabor filters and Local Binary Patterns (LBP) together with global features generated by Fourier transforms instead.

Besides faces, combined local and global features have also been used for the recognition of other objects. Murphy et al. [82] showed that using both global gist features and local features produced by various filters can significantly improve the detection performance of generic objects while at the same time gain an increase in speed. Lisin et al.[83] tried two different methods (stacking and hierarchical classification) for combining the local and global features and got a significant performance boost on classifying gray-scale images of zooplankton. All these examples show that local features and global features may be representationally complementary and when properly combined they can generate better results than using either one of them.

2.3 Computational Models and Methods

There are many computational models and learning methods in the literature (as shown in Figure 2-2), of which the following three groups are considered to be most representative.

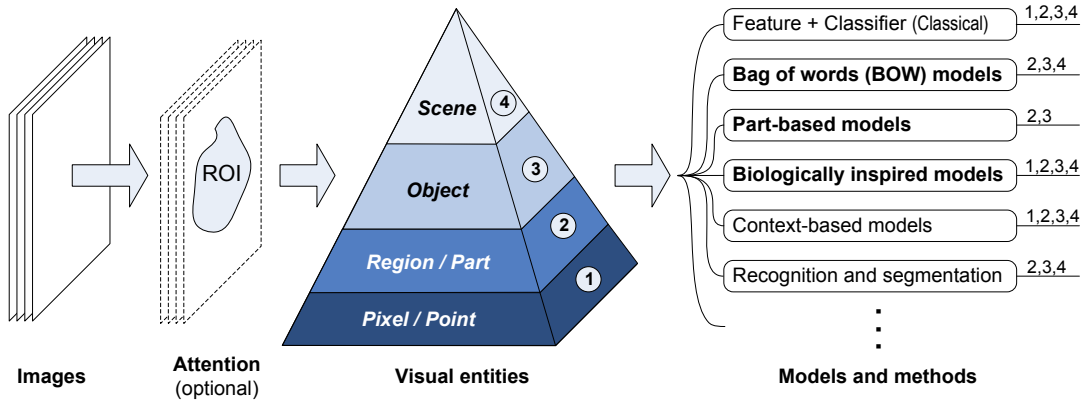

Figure 2-2 Object recognition models and methods. Different models and methods focus on different subsets of visual entities. Visual attention may be used to extract the regions of interest (ROI) from images before recognition, but it is optional.

2.3.1 Bag of Words (BoW) Models

As its name shows, "bag of words" model comes from the field of natural language processing (NLP), where the whole document is treated as a bag of words when the order of them is disregarded. Similarly, in the field of image processing and understanding, the image itself can be treated as a bag, but the "words" are not off-the-shelf as those natural words in the documents. A common strategy is to construct compact and representative visual words from the local features. These words form a codebook (i.e. codewords dictionary), which can be used to represent the images or objects by constructing histograms on it. After that, the histograms serve as feature representation for training a classifier for recognition, as shown in Figure 2-3.

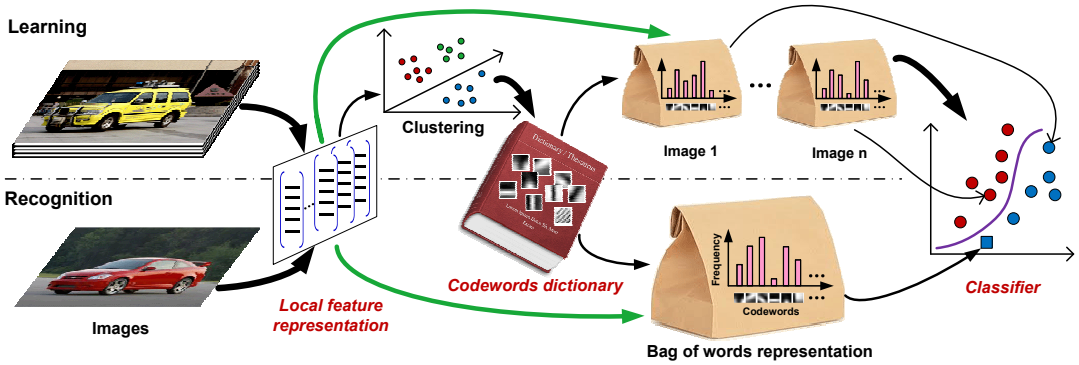

Figure 2-3 A general framework of bag of words models.

There are mainly three critical issues to be considered in designing a BoW model which make the research on BoW continue being advanced:

1. image/object representation (different local feature localization and description strategies);
2. dictionary building (by unsupervised clustering [84] or supervised learning);
3. distance function / similarity measurement (it is recently an active research topic with plenty of publications).

BoW models may have great invariance and robustness, as it can inherit them from the local features while at the same time enhancing the translation invariance and the robustness to deformations and occlusions. However, the greatest translation invariance is also the biggest shortage of BoW because it is an unordered representation which has no structure information. Many people have tried to improve it. Savarese et al. [85] proposed correlogram features to capture the spatial co-occurrences at the feature level. Sudderth et al. [86] took into account of the relative positions of the visual words. The spatial pyramid match proposed by Lazebnik et al.[87] hierarchically divides the spatial space into finer subregions and build BoW representations on each of these subregions, the ordering of which encodes the rough spatial information of the local features while

the advantages of BoW are maintained. Despite such extensions, the BoW models still cannot capture the rigorous information of object components or parts, and many of its properties such as the scale invariance and the viewpoint invariance haven't been extensively tested.

2.3.2 Part-based Models

Why part-based models? Many of the objects in the world are made of parts (either semantic parts or geometric parts), and the global shape or appearance of objects may vary a lot while the relationships of these parts are relatively more stable. For these cases, global template matching may be too rigid to adapt to the intra-class variations while bag of words models could be too loose to differentiate the structured objects from distracters with similar parts but in the wrong arrangements. Therefore, it may be better to represent the intrinsic structures of the objects and use them to handle the global shape or appearance variations. As early as 1973, Fischler and Elschlager have proposed a pictorial structure representation for matching faces as shown in the upper left part of Figure 2-4 [88]. Such a "parts and structure" model has motivated a lot of research on part-based models after 1990 for both face recognition and generic object recognition and parsing.

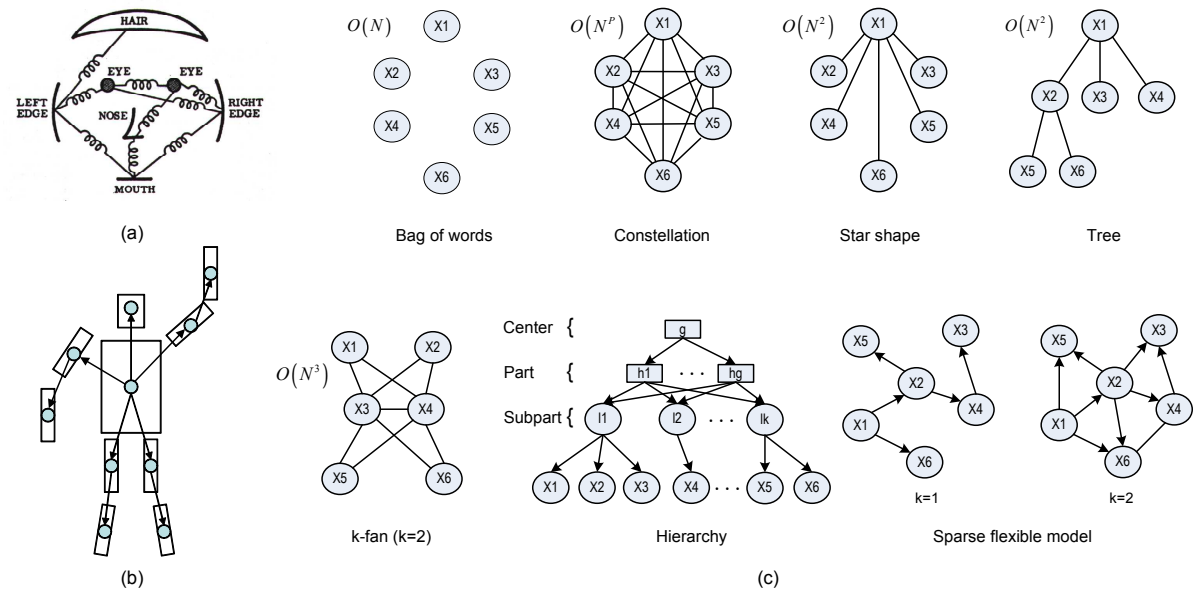

Figure 2-4 Part-based models: (a) the pictorial structure proposed by Fischler and Elschlager [88]; (b) the commonly used tree structure for human body parsing; (c) some recently proposed structures for part-based models [89]. Note that bag of words model is not a part-based model, and it is presented here only for contrasting.

Definition and key issues. Part-based models refer to a broad class of models that represent the objects by a set of parts and the relationships between them. *There*

are three major issues for any part-based models: 1) ***the structure of the model*** *(including the parts and the contextual relationships between them), 2)* ***the representation (appearance or shape) of the parts****, and 3)* ***an efficient inference algorithm*** *for finding the object parts in the test image.* The first one and the third one are highly correlated as the structure of the model determines how it can be inferred. For some models, the structure is fixed and its parameters are learned from the data, while for some others both the structure and the parameters are learned.

Recent progresses. As shown in part (c) of Figure 2-4, the simplest structure yet one of the pioneering work on statistical part-based models is the constellation model introduced by Dr. Perona and his colleagues[90, 91], which is a fully connected graph with a complexity of $O(N^P)$ where N is the number of possible positions for each part and P is the number of parts. To reduce the rigidity and computational complexity, several new structures have been proposed thereafter, such as star shape[92, 93], k-fan[94], trees[95], hierarchies[96], a sparse flexible model [89], and so on. About the representation of parts, image patches are most commonly used[90–92], while shape pieces (linelets, edgelets, junctions, contours, etc.) have also been investigated[97, 98]. For some specific objects like human bodies, the semantic parts and their relationships can be predefined instead of learning from the data. The lower left part of Figure 2-4 shows a typical tree structure which has been widely used in human body parsing and pose estimation[95].

Discussions. Compared to other models, part-based models find the explicit correspondences between object parts and the images, which can result in better recognition performance on object categories with large but constrained within-category variations [77], while at the same time they can provide object parsing results when the parts are semantic ones. Note that in some cases defining the structure of the model is not an easy task when the objects are not naturally separable and the appearances of the instances vary a lot. For the cases when the objects are extremely rigid or highly deformable, simpler models and methods like template matching or bag of words might be better choices.

2.3.3 A Biologically Inspired Feedforward Recognition Model

Since humans and other primates have great object recognition power that well outperforms any machine vision systems, building a system that emulates object recognition in visual cortex has always been an attractive idea. Serre et al. [4] proposed a biologically-motivated framework for robust object recognition, which used a hierarchical image representation expanded from the standard model of object recognition in primate cortex [3]. This framework alternately performs template matching (tuning) and max pooling operations to achieve a good trade-off between selectivity and invariance. Its built-in gradual shift- and scale-tolerance allowed it to outperform most contemporaneous complex computer vision systems.

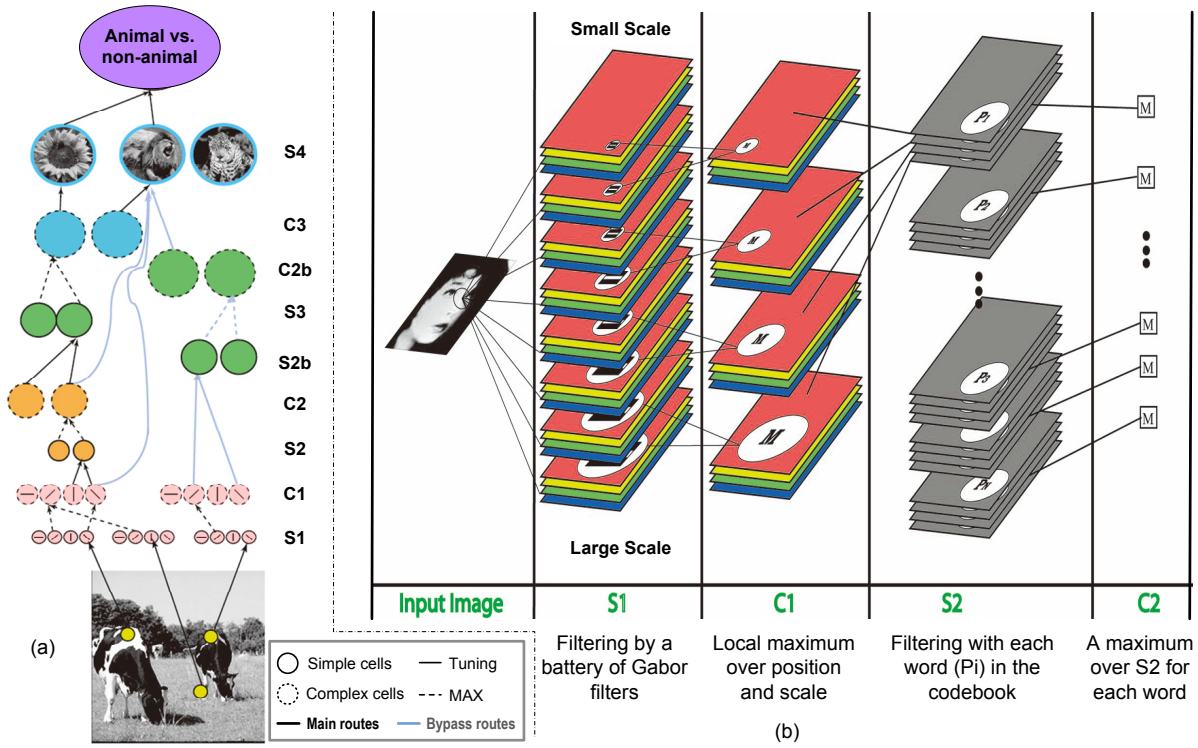

Figure 2-5 A biologically inspired feedforward recognition model: (a) the functional primitives of the feedforward model[5]; (b) the feedforward computational model[4]. Due to the space, only 8 of the 16 scales at S1 are shown. "M" stands for the MAX operation, while "P" denotes a word in the codebook (in the original model it is just an image patch).

Figure 7 shows the tentative feed-forward information passing model in visual cortex [5] and the computational model for rapid object recognition proposed in [4]. Such a feedforward recognition model showed great performances on object presence vs. absence classification [5] and multiclass object categorization [4]. Even though, many aspects of this framework could be modified to improve it. Wolf et al. [99] focused on discussing different perception strategies in hierarchical vision systems besides Serre et al.'s feed-forward framework. Mutch and Lowe [100] proposed many useful modifications that further improved recognition performance on multi-class experiments including an SVM-based supervised feature selecting technique to build the codebook. However, these efforts disobeyed the feed-forward motivation. Instead, Wu et al. [101] provided a simple yet effective unsupervised codebook generation strategy based on an information measurement.

This model is a great effort towards emulating the human vision on visual recognition, though its performance has been shadowed by some new machine learning techniques. The thoughts of it are long lasting which may be revisited years later.

2.4 Core and Unsolved Issues

The above sections give a brief review of the relatively mature and widely recognized researches on object recognition. Besides of them, there are also many important but un-solved issues and new research trends which have been or start being the focuses of current researchers.

As shown in Figure 2-6, we group these issues and trends into four different aspects. Visual perception in human vision is the centric one which inspires and influences all the other three aspects. Representation and computation is the technical support to performance evaluation and application. Performance evaluation is critical for guiding the modeling in representation and computation and evaluating its effectiveness. It can also be used as the human prior or feedback to practical applications. Different applications may require different evaluation measurements, and motivate different representation and computation methods. Therefore, these three aspects are highly correlated and the research on each of them should significantly influence the others. In the following, we analysis the recent progresses on each of the four aspects and give our comments on them, along with our predictions for future trends.

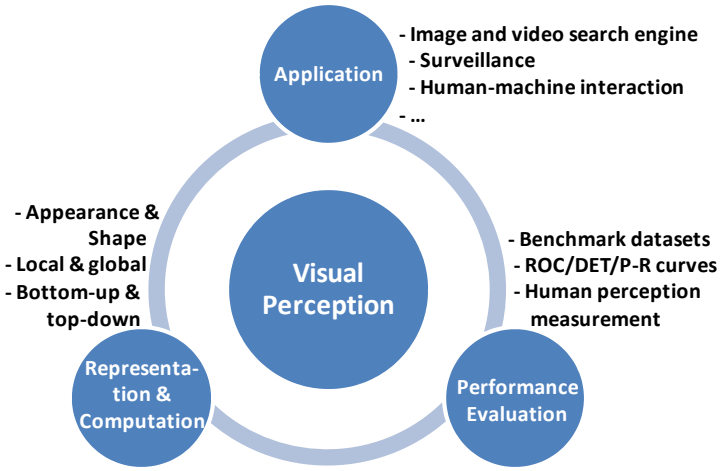

Figure 2-6 The relationship among the unsolved issues and future trends.

2.4.1 Local, Global Representation and Beyond

Local vs. Global. The first question is about the relative superiority between local and global representations for object recognition. Perceptually, it is clear that some objects are better to recognized as a whole like an egg or a bottle, while some others can be recognized using only several distinctive parts of them like a face or a bike. Further more, the importance of local and global information may vary in different recognition tasks, for example global information may be more important for categorization while the local one should be more suitable for within-category classification. Therefore, we cannot conclude with which one is better. Currently, in computer vision, local features have

dominated the representation due to their simplicity, flexibility, stability and robustness, but it does not mean that they are fundamentally superior to global ones. There are reasons to believe that abstracted global representations of objects exist in human vision system which are closer to the recognition results than local representations and they are robust to variations of the object appearance and shape. It has been confirmed by many psychophysical experiments that global perception is not a sum of the local perceptions [102], which suggests that the global representation cannot be replaced by the local ones. The two representations have their own advantages and disadvantages. Global representation is more unambiguous than the local ones, but it is not detailed enough to deal with local deformations and occlusions which can be easily handled by the local ones. Therefore, in general it is better to properly integrate both of them.

Integration. As can be seen from the literature, we are still lack of powerful tools to extract invariant and robust global features from cluttered images which need perceptual grouping and abstraction. For example currently we cannot explicitly extract a robust global representation for humans in images. Instead, we can describe the global representation by its properties, such as the pictorial structure of the human body, and use them as the global constraints for arranging the local parts. For computational issues, these constraints are usually decomposed into pair-wise relationships, which is exactly what the part-based models are trying to do. As a plus, global object templates can also be used as global representations as in Felzenszwalb et al.'s work on object detection [77]. We can generalize the idea by treating objects as the local ones and the whole scene as the global representation, then the integration becomes the popular context modeling problem for object recognition [103] which is a currently a hot topic and there exist a lot of modeling and learning methods [104–106].

Precedence. Computationally, local parts have to be constrained by the global representation while the later depends on the localization of the local ones, which is an chicken and egg problem. Perceptually, it is an open question about which one is first captured by the humans. In neural science or physiology, it is of little doubt that the processing of the visual data is from local stimulation to global structure and concept, but the order in which processing is initiated may not be nearly as relevant for determining perceptual experience as the order in which it is completed [107]. Though it was widely believed that the neural processing of visual recognition is in a local-to-global order, psychologist David Navon [108]and Lin Chen [109] both demonstrated that object perception can be global precedent. Up to now, the dispute on the precedence of them still continues. In computer vision, these two representations are usually modeled and optimized together, such as the latent SVM algorithm for simultaneous object parts localization and the global and local appearance model learning [77]. Practically, when they have to be optimized iteratively, one can choose either of them to start with.

Note that local and global are relative concepts, and the representation does not need to be only two layers. One can build a hierarchical representation from very local

generic primitives to part components, to semantic parts and to objects, and the local and global representations can be iteratively propagated between each two layers.

2.4.2 Modeling and Computation: Bottom-up and Top-down

A long lasting debate about object recognition and parsing is whether it should be a bottom-up process or a top-down process. The bottom-up process means gradually constructing higher level representations/abstractions by unsupervised perceptual grouping of the image data until reaching the semantic object-level/part-level representation. The top-down process is the opposite, i.e., going down from high-level semantic representations (e.g. a trained model or a task-specific prior) to low level representations for interpreting the data and the decision is made by evaluating the matching between the model and the data. There are numerous models and methods on either of these two strategies (e.g. BoW models are bottom-up ones while part-based models are top-down ones), and there are also many efforts trying to combine them.

In neural science or psychophysics, the bottom-up process and top-down process have been extensively researched for revealing the characteristics of visual attention which is closely related to the recognition and parsing of objects in cluttered images. Timothy and Miller [110] have recorded the neural activities of these two processes of monkeys, which indicate that top-down and bottom-up signals arise from the frontal and sensory cortex, respectively, and different modes of attention (top-down or bottom-up) may emphasize synchrony of the cortex at different frequencies. Such findings suggest the independence of these two processes. Chikkerur et al. [111] used the independence to build a Bayesian inference model for visual attention, and demonstrated the importance and complementariness of these two processes for object recognition. In their work, top-down attention is about the context for object localization which is also called spatial attention, while bottom-up attention is determined by local feature representation of the object therefore it is referred to as feature attention.

The findings in visual attention coincide well with the discussion on local and global representations where local representation is about the object/part features while global representation is about the contextual prior. The bottom-up and top-down processes can be viewed as ways for extracting these representations. Instead of treating them separately like extracting bottom-up local representation first and then perform the recognition or parsing along with the top-down global priors, we propose to have a deeper integration of them using the close-loop **Hypothesis-Verification** approach [102]. Briefly speaking, it generates object and/or part hypotheses on the bottom-up local representation and then verifies them base on the global prior; if some of the hypotheses do not satisfy the global constraints, then the locations of these hypothesized objects or parts will be updated (based on the global prior and the locations of other objects or parts) and new bottom-up local representation with corresponding new hypotheses

will be generated from the data. The Hypothesis-Verification progress continues until it converges. The kernel advantage of this approach is that after each round of verification it goes back to the data and regenerate local representations.

Recently, a lot of examples can be found on combining the bottom-up process and the top-down process for the tasks of combined segmentation and recognition[92, 112, 113], simultaneous localization and recognition[106, 114–117], and whole image parsing (segmentation, annotation/detection and classification)[118, 119]. Specifically, Yang et al.[120] have presented a method to quantitatively evaluate information contributions of individual bottom-up and top-down computing processes (two bottom-up processes from the object itself and its parts respectively and one top-down process from the context of the object) for object recognition. Since such a combination is still an open challenge, the modeling of which varies a lot and so do the learning and inference methods, and one can refer to each of them for details.

2.4.3 Performance Evaluation and Benchmark Datasets

Building a good benchmark dataset with proper performance evaluation methods is rather critical as it can motivate novel ideas on advancing the research and also evaluate different strategies and algorithms.

There are dozens of publicly available datasets on the topic of object recognition, which can be roughly categorized into four different groups according to the tasks and annotations of them as follows.

- **Presence vs. Absence Image Classification**. This task is to tell whether an image contains object instances of a specific category or not, which is usually evaluated by the *precision-recall (P-R) curves*. The most popular benchmark is the PASCAL VOC (visual object classes) database [71], which has 20 classes with thousands of images. Besides that, there are also some others focusing on hierarchical image classification based on the ontology of WordNet[121], such as the TinyImage [11] dataset with 80 million tiny images (32×32 pixels each), the ESP [122] dataset which utilizes an online computer game to label millions of images via word matching, and the ImageNet [12] database of 3.2 million (expected to be 50 million) full resolution images with relatively clean labels. These large-scale hierarchical datasets collected from the Internet may promote the research on content-based image retrieval which takes the advantage of large amounts of data and various machine learning techniques, but they are not suitable for modeling the objects themselves due to the influence of backgrounds and the large intra-class variations.

- **Object Detection and Localization**. Object detection is to detect and spatially localize object instances of certain category in images that may contain any number of them (including zero), while object localization is usually referred to the specific

problem where each test image has one and only one object instance in it[①]. For detection, usually *P-R curves* or *ROC (Receiver Operating Characteristic) curves* and its variant *DET (Detection Error Tradeoff) curves* are used as performance measurement, while for localization a soft overlap ratio can be used [7].

Object detection is one of the most popular tasks of object recognition and there are many databases on it, such as the CMU-MIT frontal face images [123], the UIUC car detection database [124], the MIT pedestrian dataset [125] along with many other pedestrian detection datasets from small ones with hundreds of images to large ones with about 1 million video frames [14, 126–130], and also some general purpose detection databases which contain multiple categories like the Caltech dataset [131] and the PASCAL VOC Challenge [71]. The PASCAL VOC Challenge [71] is currently the most unbiased detection dataset which has become a popular benchmark, however, it may not be good for guiding the promotion of current research, as too many aspects are involved[26].

- **Object Categorization and Scene Parsing**. The object categorization here is restricted to the problem of categorizing images or image subwindows with only one single object instance in each of them and usually the object occupies the major space of the image or the image subwindow, this is to differentiate it from image classification. The evaluation for them is usually *the confusion matrix or the average recognition rate*. Typical datasets include COIL-100 [40], Caltech-101 [132] and Caltech-256 [133]. Besides categorizing images with single objects, there is another valuable effort on categorizing regions in scene images where multiple object appears simultaneously. This is commonly referred to as scene/image parsing or scene understanding. Recently, a lot of databases have been built towards this goal, such as LabelMe [134], CBCL StreetScenes [135], the Microsoft Research Cambridge (MSRC) database [136], the CamVid database [13], and the Lotus Hill Database [137]. There is a new trend on object categorization using object attributes which can be shared by different object categories, and several new datasets have been built, such as the Animals with Attributions database [138], aPascal and aYahoo datasets [139]. The efforts on using attributes go deeper into the objects than traditional categorization methods by exploring the middle-level semantics, which to some extent coincides with the human vision system.

- **Within-category Object Classification and Identification**. Unlike category-level object recognition, within-category classification aims at separating objects belonging to a specific category into finer groups, while identification can be viewed as the extreme case of within-category classification in which each object instance is a separate class. *The confusion matrix and the average recognition rate are proper*

[①] However, different people may have different understandings, and some of them may treat them as the same thing.

measurements for them respectively.

For within-category object classification, a few researches have been proposed recently, one example is the multiplicative kernels [140] which has been used to classify the view angles of cars in a dataset collected from LabelMe database [134], and another work directly targeting at within-object classification [141] represents results on three different databases: a new face database for gender classification, the Baum lab RNAi cell phenotype database and a new pedestrian database collected by the authors for pose estimation. Object identification, which is an important type of object recognition, has been widely used in biometric identification, for example face recognition which has many publicly available databases [142, 143]. Besides that, the identification of some other objects like cars [144] and license plates [145] have also been researched.

Though many object recognition datasets have been built in the last few years and we can compare our algorithms on them, usually it is hard to say whether these datasets reflect the strengths and weaknesses of these algorithms. It's not that the database collections are not large and representative enough, it's that they are not systematically parameterized so that our algorithms' failure modes can be clearly identified [26]. Therefore, we need a benchmark dataset that clearly isolate the conditions which may influence the recognition performance, so that we can test the algorithms on different conditions and always be aware of the process we have made and the shortcomings we need to improve. Furthermore, the semantic annotations of current datasets are usually either too simple (e.g. only category labels) or too detailed (e.g. the parsing results and attributes) disregarding the state of the data, which is unlike the flexible human recognition results. Designing a proper annotation strategy for a deeper learning of the recognition and parsing ability of humans will be a challenging but valuable future work.

2.4.4 Scalability to Large Amounts of Visual Data

As the cameras get cheaper and cheaper, and the storage and computation resources become much more affordable than before, people pay more and more attention to taking pictures and videos, for recording their live experiences or just for fan. The high speed Internet connection makes sharing pictures and videos with other people all around the world possible and fast. In the recent few years, larger and larger object recognition datasets have been built and the internet-based applications demand even more on the scalability of the algorithms. Therefore, more and more researchers start to work on the challenging issue of making their recognition algorithms scalable to large amounts of data.

Three different ways have been tried to improve the scalability of the recognition model and method:

- **Incremental learning**. When the learning set is large or even huge, traditional

batch learning is infeasible. Therefore, an incremental learning algorithm is needed. Fei-Fei et al. [146] and Opelt et al.[147] have shown some pioneering work on it.

- **Efficient search and inference**. The branch-and-bound search strategy [148] and the hashing based indexing strategies (including the Locality Sensitive Hashing)[149] are two representatives. Saliency based approaches can also accelerate the search for objects in images when the saliency map can be fast computed.
- **Shareable structure**. When there are too many object categories, a sharable structure like a class hierarchy or a common codebook will not only make the learning of a new category efficient but also accelerates the inference process. Torralba et al.'s work on sharing features [150] and Lin et al.'s AND-OR graphs [151] are two of the many representatives.

This is currently a very active topic, and there is still a long way to go for making the recognition algorithms both effective and scalable.

2.5 Conclusion

The research focus of object recognition and parsing has clearly shifted from pure geometric modeling of several human-designed rigid objects with clean backgrounds to appearance-based statistical learning of the huge amount of real objects with dramatic variations and complex backgrounds. During this evolution, significant progress has been made on designing generic local features and proposing effective computational models such as BoW models and part-based models, which enable the machine to do some real-life object recognition and parsing tasks without specific constraints. Even though, the performance on generic object recognition is still far from being well enough for real applications, while many core issues of recognition and parsing remains unsolved, such as the integrating of local and global representations and the bridging of bottom-up and top-down processes.

There are mainly two future research trends: one is *application-oriented*, i.e. collecting larger and larger datasets with more and more real-life visual difficulties and trying to incorporate new machine learning techniques to handle them and make the best use of them; while the other is *problem-oriented*, i.e. trying to reveal the intrinsic mechanisms of recognition and generate new solutions for these fundamental problems. Clearly, most people are working on the former one which is relatively easier to follow, but the later is more important in the long run. A possible way to pursue on the later trend is to build new datasets that can better represent the human recognition properties and try to find or design computer vision algorithms for achieving the same results. This thesis is an attempt along this way.

CHAPTER 3
A Psychophysically Annotated Dataset with Deep and Layered Semantics for Object Recognition

Building a good dataset is critical for steering the research and encouraging novel ideas towards solving valuable problems. Unlike the other object recognition datasets which are designed for either category-level recognition or within-category classification, we introduce a novel dataset called "IAIR-CarPed" with not only categorial labels, but also deep and layered within-category semantics gathered from 20 subjects via a strict psychophysical experiment. To the best of our knowledge, it reveals the flexibility and the confusion statistics of human vision on object recognition in a psychophysical way for the first time, which can be used to design recognition algorithms mimicking such properties of the human vision system.

This dataset focuses on two object categories "car" and "pedestrian", which are representative for doing research and also very important for real applications. It involves 3132 images collected from pictures taken under various conditions and 8567 objects carefully annotated by all the 20 subjects. Besides deep and layered semantics, five types of detailed visual difficulties of these objects are also provided, which can be adopted for evaluating the representation and generalization ability of the recognition systems against individual difficulties. We present here the details of building this dataset, its statistics and properties, and discuss possible applications of it on object recognition, in which a novel application called *deep and layered object recognition* is specially recommended for exploring this dataset. We hope that IAIR-CarPed can motivate research on solving the proposed deep and layered object recognition problem towards the ultimate goal of deep scene understanding.

3.1 How About Being Deep and Layered?

There are several trends in the progress of building new object recognition databases as can be seen from the above review:

1. **Larger** (Collecting more images). As the computational power of machines increases and the visual data on the Internet grows, people are trying to collect larger and larger databases for object recognition tasks. For presence vs. absence image classification, the currently widely used PASCAL VOC Challenges [71] which has only 20 object categories with thousands of images starts facing competitive large-scale databases such as ESP [122], 80 million Tiny Images [11] and ImageNet [12] which have millions of images for thousands of object categories. In object

detection, the MIT Pedestrian database [125] and the INRIA Person database [126] which have only hundreds or thousands of positive examples are being replaced by much larger databases like the Daimler Pedestrian Detection Benchmark [129] and the Caltech Pedestrian Detection Benchmark [14] which have tens of thousands or even hundreds of thousands of annotated pedestrians. In object categorization and scene parsing, Caltech-101 [132] gives way to Caltech-256 [133], while CBCL StreetScenes [135] and the Microsoft Research Cambridge (MSRC) database [136] are being challenged by LabelMe [134], CamVid [13] and the Lotus Hill Database [137].

2. **Harder** (Involving more variations). Performances of the newly proposed recognition algorithms on early simple databases (such as the Caltech-101 [132] dataset, the MIT Pedestrian database [125] and the FERET face recognition database [142]) are becoming saturated, therefore, much harder databases have recently been proposed. Many of the newly proposed databases are either trying to use the images on the Internet directly which have great unbiased intra-class variations (like PASCAL VOC Challenges [71], LabelMe [134] and ImageNet [12]), or targeting at object recognition in real applications like the Caltech Pedestrian Detection Benchmark [14] which is captured in typical urban streets using a camera mounted on a vehicle for the purpose of automotive safety. These databases are much harder than those specific ones captured in controlled environments as they involve many visual difficulties, such as the variations of view, pose, illumination, occlusion, backgrounds, etc. Though a good recognition algorithm is expected to be robust to all these difficulties, it is hard to find such an algorithm. Therefore a good benchmark database is expected to be able to isolate the various difficulties that we need to conquer, so that we can tell the exact differences between different algorithms [26].

3. **Deeper** (Providing non-categorial semantics). The annotation of the object recognition databases is going deeper and deeper to reveal more details of how humans perceive the objects, therefore it may motivate new object recognition models. Spatially, more databases provide pixel-wise segmentation ground truths instead of polygons and bounding boxes for the annotated objects, while semantically, people start to provide within-category semantics like gender and pose [140, 141] and cross-category middle-level semantics like attributes [138, 139]. Deep semantics are important elements for understanding the highly abstracted category-level semantics while themselves are useful for real applications.

4. **More layered** (Constructing hierarchical semantics). There are tremendous objects in the world and the number of semantics attached to them is also huge. After a long time of evolution, we humans have developed efficient structures to organize them and therefore we are able to quickly recognize different things and distinguish them from other very similar ones. The taxonomy is one of such structures, and

many recent databases [11, 12] have used the WordNet [121] to collect images from the Internet. Such a layered tree structure is proved to be very helpful for object recognition [12].

Within these four trends, the first two are natural choices especially for Internet-based applications which have high demands on the scalability and robustness, but setting higher goals may not necessarily results in better understanding of the intrinsic mechanisms of object recognition in human vision, while the other two go deeper into the box and try to borrow more from the human vision system. However, currently the two later trends are still separate, deep non-categorial semantics are forced to be labeled disregarding the layer of confidence, while layered semantics are still limiting themselves to categories which are all nouns but no adjectives.

In our daily lives, however, we are neither interpreting all the objects appear in our sights as deep as possible (from categories to attributes and other within-category descriptions) nor just categorizing them into a few abstracted categories. Instead, object recognition in our vision systems seems to be both deep and layered, i.e., some of the objects are deeply classified (into subcategories), while the others are recognized at a coarser level. Though visual attention plays an important role in layered recognition, it still holds when the effect of visual attention is excluded. Closer objects tend to be deeply interpreted while farther ones are more likely to be coarsely treated as the visual information is weaker. The left part of Figure 3-1 shows a concrete example of the deep and layered human recognition results of different object instances within a single image, while the middle part illustrates the currently focused tasks based on the existing datasets. The problem is: can we build up a dataset with deep and layered semantics like the right part of Figure 3-1 which is close to the human recognition results?

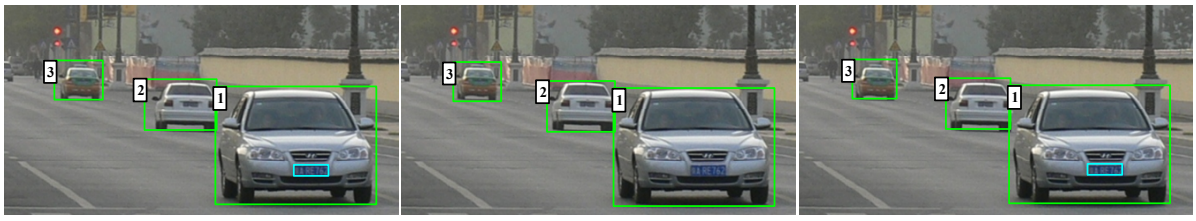

Figure 3-1 Motivation of building the IAIR-CarPed dataset.

We present here the IAIR-CarPed dataset, which is as far as we are aware the first object recognition dataset with carefully collected deep and layered semantic labels. It consists of 3132 images collected from pictures taken under various conditions, in which 8567 objects of two representative categories ("car" and "pedestrian") have been labeled

by 20 subjects through a strict psychophysical experiment. We focus on the layered orientational information of the object (from totally unspecific to surely specific) and the clearness of its key part (the license plate of a car or the face of a pedestrian) if visible. The annotations for each object include the object bounding box, the category of the object, 20 semantic labels from the subjects, the difficulty labels indicating the presence of 5 different types of visual difficulties, 20 votes for the clearness of the key part if the object has certain orientations, and the bounding box of the key part if it is clear.

This dataset has the following four contributions for advancing the research on object recognition:

- Deep and layered semantic labels are obtained from 20 subjects for all the annotated objects in the dataset, which can be used as a benchmark for evaluating algorithms targeting at deep and layered object recognition.
- Votes to different semantic labels reveal the confusions of the human vision among the semantics, and they may be valuable for guiding the learning of human-like recognition algorithms.
- The bounding boxes of the objects are suitable for researches on object detection. Due to the large within-class variations, it is a challenging and valuable dataset for this traditional task.
- Difficulty labels of the objects are applicable for evaluating the representation and generalization ability of different features and/or learning algorithms.

The rest of this chapter is organized as follows. Section 3.2 shows how this dataset was constructed, from choosing the categories and the semantic structure to the collection and cleaning of the data. Section 3.3 presents the details for annotation via a psychophysical experiment. The voting integration strategy for the final semantic label, the computation of human confusions and some other statistics of the dataset are stated in section 3.4. We discuss possible applications of the proposed dataset in section 3.5. Finally, discussions and conclusions are given in section 3.6.

3.2 Dataset Construction

We present here the details for constructing the IAIR-CarPed dataset, including the choosing of object categories, designing their semantic structures for annotation, and the process of collecting and cleaning the images.

Instead of trying to collect a large-scale dataset which contains a huge number of object categories, we choose to collect data on only two representative ones: car and pedestrian, believing in that it is more important to research on the intrinsic mechanism of deep and layered recognition itself than to explore its scalability. Cars and

pedestrians are within the most common objects in our daily lives and very important to computer vision based applications like automotive safety and surveillance. Moreover, they cover both rigid and deformable objects which expose different challenges in real scenes. Therefore, choosing these two categories for building a deep and layered recognition dataset is promising. We are also trying to make the dataset best represent the real world by involving sufficient difficulties, but unlike other existing datasets, we separate the difficulties by labeling them independently so that the dataset can be used to measure the robustness of algorithms against each of them.

3.2.1 Deep and Layered Semantics

For each category we are interested in, we would like to do deep within-category recognition with layered semantics, and the ground truth label of each annotated object in the IAIR-CarPed dataset will depend on the human visual perception result of its appearance. Based on our visual experiences, we find the orientation is an important property of the interested objects from these two categories in static images, and its value can be layered depending on the appearance of the objects. We choose to use the uncertainty of the orientation as the measurement for designing the layers as shown in Figure 3-2. In the two tree structures, objects belonging to the root nodes ("unspecific") are the most uncertain ones, i.e., we almost have no feelings of the concrete value of their orientations. The second layer of the trees indicates that the orientation can be narrowed down into smaller ranges but still cannot go down to very specific values. The labels in the third layer are those specific orientations. To make things easier, we choose to use semantically intuitive orientation labels (such as front, back/rear, left and right) for the third level semantics, instead of trying to tell the concrete degree of the orientation which is visually hard. Therefore semantic nodes in the second layer are defined as "front/back(rear)" and "left/right" which are more ambiguous than the third-level semantics, but more certain than the root nodes. For those objects whose orientations stay between two specific orientations, we treat them as orientation unspecific ones, i.e., put them into the root nodes.

In addition to the orientations of the objects, some object parts are also important and may be of great interest to humans, such as the license plate of a car and the face of a pedestrian. Therefore, we treat the object with a clear key part as a deeper interpretation than global orientation classification. To make it easier, we only consider the clearness of the key part when the orientation is specific, i.e., the frontal and rear cars and the frontal pedestrians. In these cases, the key parts (license plate and face) are more likely to be reliably detected than the others.

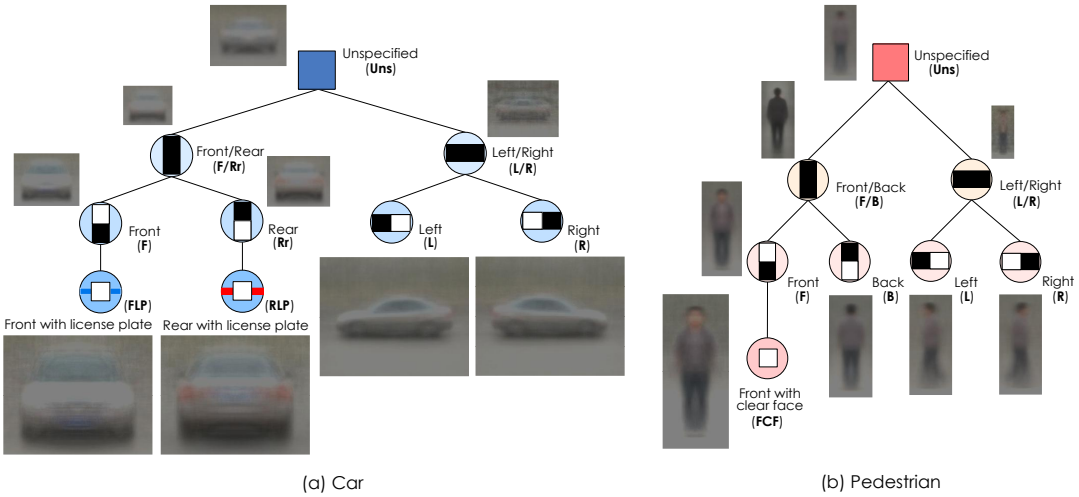

Figure 3-2 The tree structure of the deep and layered semantics for each category. The top three layers are about the orientation of the objects, while the fourth layer is on the clearness of the key parts. Going down along the trees, the semantics get more and more certain and specific. The shapes of the nodes are designed to illustrate the semantic meanings associated to them. Beside the nodes are the average images of the annotated objects with the corresponding semantic labels (excluding the truncated objects). We also choose proper abbreviations for the semantic labels so that they can be indicated easily. For convenience, they may be used later without specific declaration.

3.2.2 Data Collection and Preprocessing

We were trying to make the dataset as representative as possible when we collected the data. There are two key points which have guided our collection: a) to make sure there are sufficient examples for each output state, the data itself should have enough diversity in term of the aspects determining the variability of the outputs, namely *sufficient between-class variation*, and b) to make the dataset as generic and natural as possible, we should also involve as many variances of object appearance as possible even when these objects have the same output values, namely *significant within-class variation*.

Details about how this dataset was constructed are as follows.

Taking/Collecting Pictures Most pictures are taken from natural scenes in a city with original image size of 2048×1536 or 640×480. These natural scenes include campus, park, street, mountain, and rural areas. To make the dataset most generic, we also gathered some other images taken under unusual weather conditions (like thick fog, snow and sand storm) or special shooting angles (like sunrise/sunset and backlight) from the Internet or taken by us in controlled environment. The diversity of pictures ensures the enrichment of object orientations, including ambiguous orientations caused by the environment, e.g. front/back ambiguity under backlight condition. We haven't considered frames taken from videos, but they can be our future candidates.

Selecting Pictures There are some rules which have guided our selection of the taken pictures or those from the Internet. For better understanding of the IAIR-CarPed

dataset, we list them as follows.

- Very low quality pictures are rejected. Cases are those highly blurred ones and very low contrast ones.
- Pictures with only unusual cars and other vehicles are not selected. Trucks, buses, and other types of non-car motors are of this type. Racing cars or irregular concept cars are also out of consideration.
- Pedestrians are restricted to be upright people (standing, walking, running, lifting, backpacking, etc.). Upright people with very unusual poses are also rejected.
- Avoid extreme occlusion. We do not considered highly occluded cases, like the objects with more than 2 out of 3 of them occluded.
- The objects shouldn't be truncated too much. Less than 50 percent is acceptable.
- The pitch angle of the camera should be normal (say held by a person with reasonable distance to the object), neither too high nor too low.
- Statistically, the amount of samples with different semantics in the predefined output space should be as balanced as possible.

Note that some of the demands are on objects but not images. There are cases that some objects in an image satisfy the demands while the others in the same image do not, we might still keep it for further processing.

Cropping and Resizing During the selection process, we got some pictures which partially satisfy the demands. For example, some pedestrians are highly blurred due to motion and low illumination while the others are in good conditions. In order to make the best out of these pictures, we tried to crop subimages from them and used these cropped images instead, ensuring that in each image most objects are acceptable. To make the distribution of object size reasonable, we also downsized some images before cropping. Been cropped or not, all the images were finally resized to a fixed size of 512×384 for consistency and convenience. Whenever the resize operation was used, only downsizing was allowed to ensure the quality of the data, and the aspect ratio of the objects were kept as they had been to avoid distortion.

3.3 Annotation

There are totally 3132 images adopted for annotation, each of which contains at least one instance of the interested objects. For each image, we label all the object instances (cars and/or pedestrians) within it which satisfy our demands: visually tellable, with less than two thirds' occlusion, truncated less than one half, type(for car) or pose(for pedestrian) satisfying the demands in 3.2.2, and above minimum size (car 30-pixel wide and pedestrian 45-pixel high).

Our annotations are mainly two typed: semantics and localizations. The semantics are the nodes shown in Figure 3-2 while the localizations are the bounding boxes of the objects and their key parts (if clear). Figure 3-3 shows some examples of the annotated images with an integrated semantic ground truth for each object in the image. The way to annotate the objects is introduced in this section, while the method we have used for integrating the semantic labels will be presented in 3.4.1.

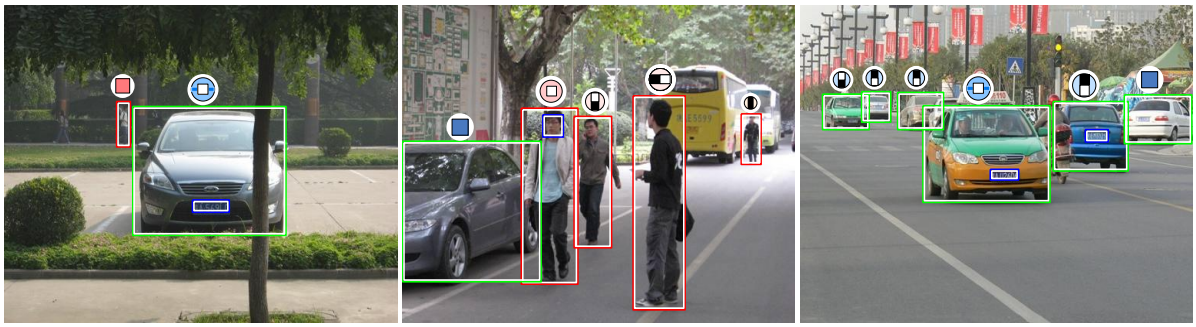

Figure 3-3 Example images and their annotations. Green and red bounding boxes represent annotated cars and pedestrians respectively, while the sign above each object illustrates its semantic label.

3.3.1 Visual Difficulties and Object Localization

As stated in 3.2.2, the dataset is made to be as generic and representative as possible, therefore it has great variations in the data, including significant visual difficulties. We follow the suggestion of Sven Dickinson [26] to isolate the visual difficulties by categorizing them into 5 different types as list below, hoping that it can aid the analysis of the robustness of different recognition algorithms and thus indicating improving directions. Since there are examples with mixed difficulties, we check the 5 difficulties one by one for each object to see whether the object has it or not, resulting in a 5-dimensional difficulty label vector. Each element of the vector is a binary value, for which 1 indicates that the corresponding difficulty exists while 0 means the opposite.

The five difficulty types are:

- **Occlusion**. Occluding less than 2/3 of the object area is considered, while larger occlusion or very little occlusion (less than 1/5) is ignored.
- **Truncation**. 1/5 to 1/2 of the object is truncated, based on the area of the bounding box.
- **Nonuniform illumination**. Caused by shadows of other objects like trees, or uneven lightings at night.
- **Low contrast**. Under conditions of low illumination (e.g. in the rain), or motion blur.

- **Unfrequent shape**. We want the 3-D shapes of the cars and pedestrians to vary as little as possible. Therefore, car types besides sedan are treated as unfrequent ones, and pedestrians with unusual pose or little children are of this type. Actually, in the IAIR-CarPed dataset, they are unfrequent.

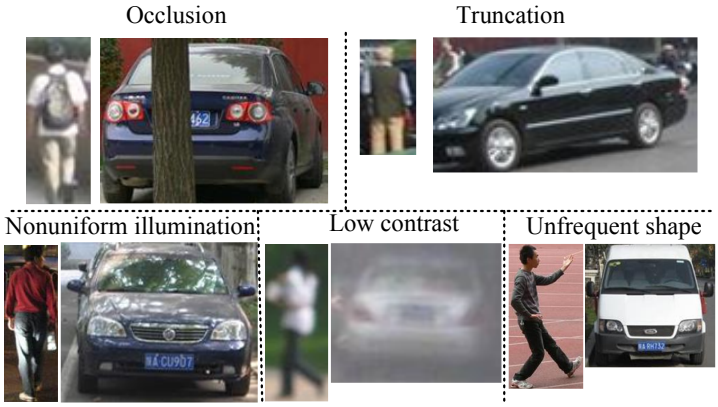

Figure 3-4 Visual difficulties and their actual examples in IAIR-CarPed dataset.

Figure 3-4 shows some annotated objects with one of the five visual difficulties. Though most of the time occlusion and truncation are treated as the same, we would like to differentiate these two concepts in this work so that they can refer to different scenarios which can be handled in different ways. The way we differentiate them is based on the bounding box of the object. We label the bounding box of a object as tight as possible to just bound the visible parts of the object without objective estimation of its missing parts. Then we define "occlusion" and "truncation" as: **occlusion** – some part(s) of the object in the labeled bounding box is/are occluded by other objects, and **truncation** – the labeled bounding box is considered to be smaller than the actual bounding box if the object is fully visible. Note that our definition of these two concepts are not mutually-exclusive, so that one object may have both of these two difficulties. By separating these two concepts, it is expected that pure occlusion without truncation is relatively easier to handle as the extension of the object is still tellable, while the cases in which truncation is involved are very challenging as the real bounding boxes are hidden. Therefore, in our setting, truncation is the most challenging difficulty, and it needs to be handled using latent part based models with part presence reasoning which is an even harder optimization problem than the part-based deformable model [77]. We leave it as an open challenge for future research.

Since the difficulty labels and the bounding boxes are deterministic, we had them labeled by one single person and checked by another.

3.3.2 Orientation

The orientation of the objects is the main cue used by us to design deep and layered semantics as shown in Figure 3-2. It can be seen that these semantic labels cannot be easily separated for some ambiguous cases and different people may have different opinions on them. Therefore, we decided to use the strategy of having a group of people (usually several dozens) to label them independently and then integrate the results, as widely used in psychophysical experiments.

However, we immediately found that human vision system is so powerful that by careful examination it can use very minor details to tell the exact orientations even though the objects are very small or look ambiguous at a glance. In our daily lives, however, we usually do not try to recognize every single object as deep in the semantic hierarchy as possible no matter how far it is away from us. Closer objects tend to be more informative so that they can be more easily categorized into deeper layers. Such a phenomenon shows the high efficiency of the human vision system based on the so called rapid recognition, which has been researched a lot by neuroscientists in the past few years [5]. In order to ensure the consistency and the layered property of the labeling results, we followed the rapid recognition idea, and extended it from binary presence vs. non-presence image classification to our deep and layered recognition. Unlike the assembling of a sequence of binary labeling processes of ImageNet database as stated in [12], a one-shot multiclass assignment is used for the deep and layered labeling of IAIR-CarPed. Our experiments show that humans perform pretty well on this task without much effort, which proves our observation and proposition.

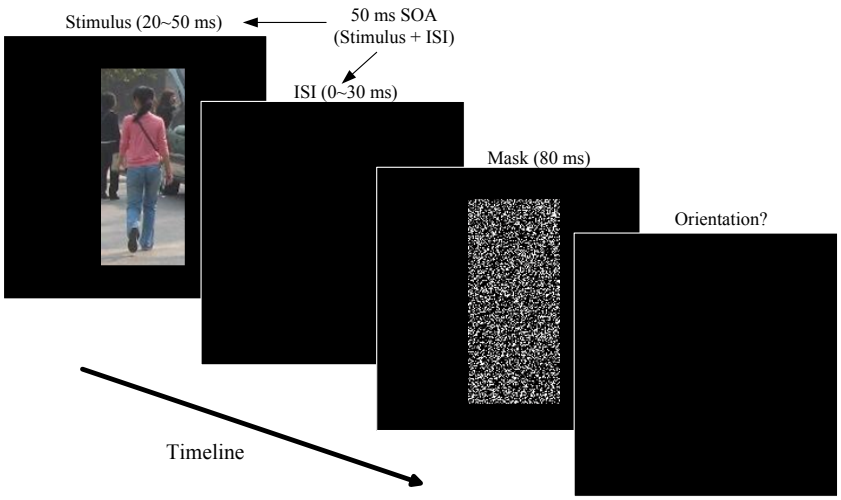

Figure 3-5 Deep and layered semantic annotation through rapid recognition. Please refer to the text for details.

We hired as many as 20 undergraduate and graduate students to participate in the psychophysical experiment which is designed according to the one presented in [5]. As shown in Figure 3-5, our experiment operates as follows. At each time, the machine

randomly picks an unlabeled object instance cropped from the original image using its bounding box with some margin, and shows it in the middle of the screen against a black background for a stimulus time ranging from 20ms to 50ms, which is in inverse proportion to the scale of the object. When the stimulus time is less than 50ms, an interstimulus interval (ISI) is shown with the black background only until the stimulus onset asynchrony (SOA), which is the stimulus plus the interstimulus interval, lasts as long as 50ms. After that, a mask with random noises which has the same size as the image shown before is displayed in the middle of the screen against the black background for 80ms. This is to block possible top-town back-projection. Then the mask disappears, leaving only the black background. The subject who is participating in the experiment is asked to tell the orientation of the object by assigning one of 7 labels in the first three layers of the semantic tree to this object. This is done by pressing a predefined key and usually it takes less than 1s if the subject sees the object. Once the label is saved, the system transfers to the next one, and a new round begins.

The reason for varying the stimulus time is to compensate the response time of the eyes for adapting to the downsizing of the object scale. The smaller the object is, the longer the stimulus lasts. Though we haven't found the psychophysical or biological prove for the exact relationship between the scale of the object and the stimulus time needed for rapid recognition, experimentally we found that such a simple strategy generates reasonably good results.

Our labeling tool is designed to be considerate and user-friendly. Late response exceeding 1s will be treated as a sign indicating that the subject is not sure about the orientation of the object, therefore his/her choice will be considered to be unreliable, and the labeling tool will automatically choose the parent label of the label chosen by the subject to increase the reliability. If the subject needs a break, he/she can just leave the tool without response and come back to press a resume key to continue the labeling. If one realizes that a mistake has been made, then the tool can roll back to the former object and let the subject reassign a label.

Note that for objects with at least one of the three visual difficulties of occlusion, truncation, and low contrast, rapid recognition is very hard to generate reliable results. These objects usually need selective visual attention or even visual reasoning which require feedback circles in the cortex. Therefore, we designed a separate experiment for labeling these objects by extending the stimulus time to as long as 500ms.

3.3.3 Key Part Clearness

In addition to the orientation, we also labeled the clearness of the key parts for objects with certain orientations: frontal/rear cars and frontal pedestrians. This is so far the deepest layer of our semantic structure which goes beyond the object itself to its key part. It is motivated by the visual experiences that we humans are very interested in

the key parts (especially the faces of pedestrians) when they are clearly visible, and such an extension naturally fits the layered human recognition of these two types of objects.

However, it is impossible to put the key part clearness labeling directly into the rapid recognition framework, because the key part recognition is usually competing with the recognition of the global object. When the object is shown very rapidly, humans can usually focus on either the orientation of the object or the key part of it, but not both as there is no time for saccade. In this situation, saliency and visual attention will play an important role on the final result. If the object itself is more salient than its key part, then it's more likely that humans will get its orientation while disregarding the clearness of its key part. On the opposite, if the key part is more salient, then it may attract humans' attention, making them forget to tell the orientation of the object. Therefore, we chose to label the orientation first by forcing the subjects to focus on the global appearance of the object, and after getting the final orientation of the objects through the annotation integration as to be represented in the next section, we asked the subject to label the key part clearness of the frontal/rear cars and frontal pedestrians by just looking at the local areas where the key parts may appear.

Note that the clearness of the key parts is not easy to be properly defined, as we do not know the exact perceptual threshold of it and how this threshold can be used for consistent labeling by many subjects. After plenty of observation, we defined it as: for license plate the clearness means that most of the characters on it are readable, and for face it means that the facial features (especially the eyes) of it are tellable. We observed that when the characters on the license plate are perceptually separable, they are usually also tellable, therefore we chose the separability as our measurement instead of the recognition results of characters for its simplicity and efficiency. The clearness check seems to be more complex than the simple presence classification, rapid response may be unreliable. Therefore, enough stimulus time (1s) is given for this task.

3.4 Statistics

We present some informative statistics of the dataset based on the annotations. The most important statistic is the integrated semantic label for each object from the original 20 labels, which is presented in 3.4.1. Another critical information for understanding the intrinsic mechanism of object recognition in human vision is the confusion matrix, which may guide the learning of a human-like recognition model and can also be used for evaluation. We show the details of it in 3.4.2. Other statistics such as the distributions of the semantic labels, the object scale distributions for each semantic label are presented in 3.4.3.

3.4.1 Voting for Annotation Integration

As presented in section 3.3, the semantic labels were gathered from the subjects through two separate stages: one for orientation labeling, and the other for key part clearness labeling. Since the later is a binary classification problem, a simple majority voting is used to integrate the labels, i.e., the clearness is confirmed if and only if more than half of the labels are "yes".

For object orientation labeling which is a multiclass classification problem, though normal voting is a simple and fair way to integrate the labels, it is not suitable for our problem. If an object has very diverse votes, for example, 5 to "front or back", 6 to "front", 5 to "back", and 4 to the others, then it will be dangerous by choosing "front" as its ground truth, as most people do not agree with that. However, if we choose "front and back" instead, then the 11 people who have chosen "front" or "back" would partially support that, and maybe change their ideas to "front or back" to get a common agreement. This is plausible because the diverse votes indicate that the example is hard to tell whether it is "front" or "back". Actually, such a case is normal in our rapid recognition setting. When an object's orientation is ambiguous, people may choose the right semantic label in the two upper layers of the tree, but may also choose either one of its children (i.e., a more specific orientation) if they think they can somehow weakly tell its orientation. These risky judgements result in the diversity of votes for the visually ambiguous object. From the actual labeling results we found that the subjects hired by us tend to make such risky decisions instead of conservative ones.

To integrate such labeling results, we propose a transferred super-majority voting strategy as shown in algorithm 1.

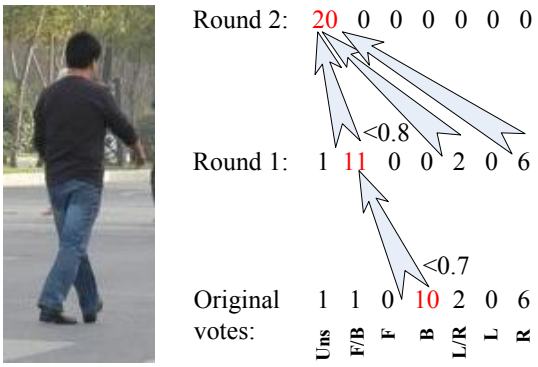

Figure 3-6 A real example of the transferred super-majority voting for label integration. In this example, though the majority of the votes are "back", they cannot go beyond the threshold 0.7. After two rounds of label transferring and subtree integration, the final label is "unspecific", which is a suitable assignment with respect to the original votes.

The goal of the transferred super-majority voting algorithm is to find the deepest label for the object which has enough supports from all the original votes with respect to predefined thresholds. In our case, the thresholds for the nodes in the same level of

Algorithm 1 TRANSFERRED SUPER-MAJORITY VOTING:

Input:

The semantic tree structure $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{N_1, \ldots, N_n\}$ are the nodes, and $\mathcal{E}$ are the edges (if $(N_i, N_j) \in \mathcal{E}$, then N_i is the parent of N_j);

The original semantic vote results $\mathbf{v} = (v_1, \ldots, v_n)^T$ of the input object to the tree nodes;

A group of predefined winning thresholds $\mathbf{t} = (t_1, \ldots, t_n)^T$.

Output:

The final label index of the input object $l \in \{1, \ldots, n\}$.

1: Normal voting: $l = \arg\max_i v_i$. Note that if l has multiple initial values due to equal votes, then go through the following steps until termination with each one of them, and pick the one lies deepest in the tree $\mathcal{T}$ as the final label index l from all the results.

2: Thresholding and decision making: if $v_l / \sum_{i=1}^{n} v_i \geq t_l$, then terminate and return l.

3: Subtree integration: $\forall N_i \in \mathcal{V}$, if $(N_l, N_i) \in \mathcal{E}$, then $v_l = v_l + v_i, v_i = 0$.

4: Termination or label transferring: if $v_l / \sum_{i=1}^{n} v_i < t_l$ and $\exists i, (N_i, N_l) \in \mathcal{E}$, then $l = i$ and go back to step 3, else terminate and return l.

the tree are set to be the same, and the ones for the three different semantic levels are set to be 0, 0.8 and 0.7 respectively from top to bottom. We found that such a setting results in semantically plausible final labels for the annotated objects. An example of the transferred super-majority vote is shown in Figure 3-6.

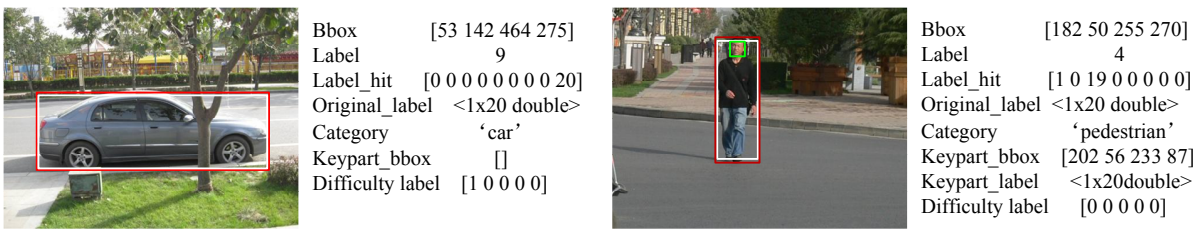

Figure 3-7 Examples of the final annotation information provided for each object. The "Label_hit" stands for the transferred votes for each semantic label in the tree structure in a depth first order including the key part clearness label, while the "Label" is the index of the final label after annotation integration. "Original_label" and "Keypart_label" are the original labels from the 20 subjects.

We provide both the original labels from all the subjects and the final voted label in the annotation of each object, so that one can choose to use our proposed label as the ground truth or generate his/her own result from the original labels based on any other vote integration strategies when needed. Figure 3-7 presents two typical examples of the final annotation information for the annotated objects in the IAIR-CarPed dataset.

3.4.2 Semantic Confusions of Humans

The original votes not only provide elements for final label assignment, but also reveal the recognition ability of humans on distinguishing different semantic subsets. We treat the final voted label as the ground truth of the object, and the votes to the other labels as recognition mistakes. By computing the average percentage of the number of votes to the mistaken node N_j while its voted ground truth is N_i, we can see how likely humans may mistake the semantic label associated with N_i for the one associated with N_j. We believe that the semantic confusions of humans are valuable for guiding the learning of a recognition model to generate reasonable results as close to those of the humans as possible. If humans seldom make certain mistakes, the machine is better to avoid making them. And if humans usually get confused on some semantic labels, the same types of mistakes made by the machines should be relatively more acceptable.

Since we have two stages of labeling and voting, a confusion matrix involving all the semantic labels in our output space as shown in Figure 3-2 needs to fuse the voting records from these two stages. Though it's very easy to do so, we discuss here only the confusions between different orientational semantics, as they can represent the human confusions more clearly and intuitively. When working on all the semantics including the clearness of the key parts, one only needs to separate the votes to the third layer and the fourth layer by the key part clearness label of these objects, then an augmented confusion matrix can be got.

After getting the final labels from the voting, we can compute the confusion matrices using the original votes, which are referred to as **original confusions**. To see how people make mistakes on different data, we compute the confusion matrices on two different subsets of the two categories (car and pedestrian) respectively: set "S" contains objects without any visual difficulties, i.e., only simple examples, while set "D" is the complementary subset, containing objects with at least one special difficulty. Note that set "S" is labeled though rapid recognition only, while most of the examples in set "D" are not labeled by rapid recognition due to their difficulties. The results are shown in the upper row of Figure 3-8. It can be seen that there are very tiny differences between the two different subsets of both car and pedestrian, which demonstrates that our rapid recognition strategy is reliable, as humans can almost eliminate all the visual difficulties on set "D" when enough time is given and therefore the confusions on that set are considered to be the real representations of how people may confuse these semantics. Compared to the confusions on set "D", those on set "S" from "Uns", "F/Rr", "F/B" and "L/R" to their child semantics are slightly larger, which indicates that in rapid recognition people tend to make more mistakes on ambiguous examples. This is reasonable because in rapid recognition humans do not have enough time to carefully check the details if they are not clear.

Though the original votes directly represent the recognition results of humans,

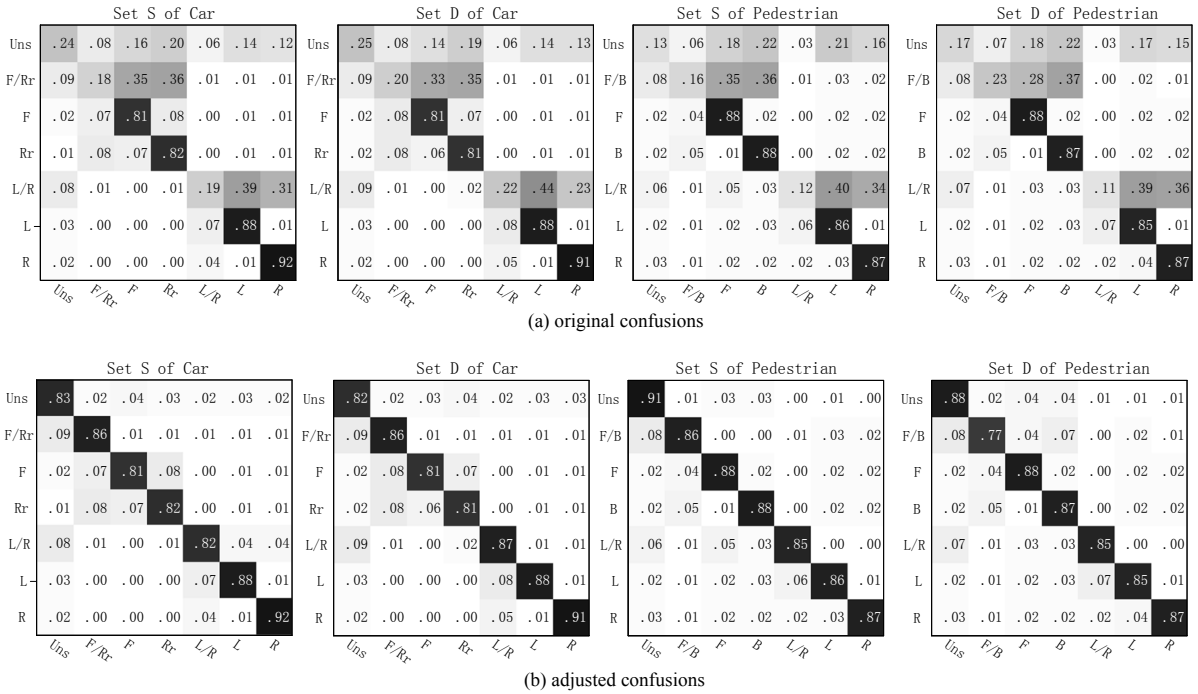

Figure 3-8 The human confusions on the object orientations through rapid recognition. The upper row shows the human confusions computed using the original labels which are aggressive, while the lower row presents the adjusted confusions which look much more conservative. The adjusted confusions are computed using the transferred labels which significantly prevent predicting ambiguous objects with deeper labels than they should have.

votes of semantically ambiguous objects are relatively diverse. Therefore, confusions computed on these semantic labels (such as "Uns", "F/B(Rr)" and "L/R") are very large, especially those confusions to their child nodes. Following this kind of confusions may end up with aggressive predictions which might be risky in some applications. Therefore, we propose to use the conservative confusions computed using the adjusted votes after the transferred super-majority voting, namely, after each subtree integration, the votes to the nodes in the subtree are all redirected to the root node of the subtree. By doing so, the original confusion matrices shown in the first row of Figure 3-8 are mapped into the ones shown in the second row of the same figure, which are referred to as **adjusted confusions**. Such an adjustment has significant effects on reducing the confusions from ambiguous semantics to their child semantics.

Note that the confusions shown in the figure are from all the applicable examples in the dataset. When the confusion matrix is used for model training, it should be computed from the training examples only.

3.4.3 Data Distributions

Scale distributions Intuitively, it can be inferred that objects belonging to the higher layers tend to have smaller scales. To verify this and show the actual distributions of our data, we compute an object scale histogram for each semantic subset as shown in Figure 3-9 excluding the objects with truncations whose actual size are unknown. Undoubtedly, semantically ambiguous subsets have more examples in smaller scales. And the scale distributions present some statistical properties of the data: a) there are more cars in the parallel directions than in the vertical directions and the majority of the cars are far way especially in the parallel directions, which is reasonable when we capture the pictures along the streets; b) few of the pedestrians come close or very close to the camera, and numbers of them in the other two groups are very close; c) unlike the cars, pedestrians in the four orientation specific subsets have very similar scale distribution though the parallel directions have relatively more examples.

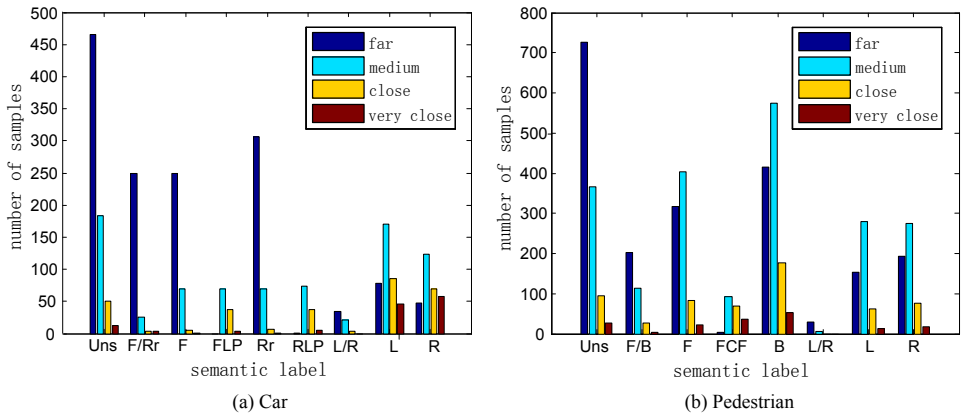

Figure 3-9 The scale distributions of the objects on the semantic subsets. We briefly quantize the scales into four groups (far, medium, close, and very close) according to the width of the car and the height of the pedestrian respectively. More specific, for cars, the four groups are [30, 128], (128, 256], (256, 384] and (384, 512], while for pedestrians they are [45, 96], (96, 192], (192, 288], and (288, 384].

Difficulty distributions The detailed labeling of visual difficulties is an important and distinguishing property of IAIR-CarPed which enables the robustness analysis of the features and/or the recognition algorithms. Since we labeled the presence of the five difficulties (occlusion, truncation, non-uniform illumination, low contrast, and unfrequent shape) independently, there are totally 2^5 combinational states. We list 8 of them with their distributions on the semantic subsets in Table 3-1. They include the simple set "S" without any difficulties, the five sets with only one single type of difficulty (from "D1" to "D5"), the whole dataset excluding only the truncated examples ("SD-D2"), and the whole dataset itself ("SD"). It can be seen that there are totally 3292 cars and 5275 pedestrians in the dataset, in which about half of them are difficult ones with at lease one visual difficulty. For both car and pedestrian, "L/R" contains the smallest

number of examples, which indicates that for these two categories, left and right are relatively easier to tell from each other. As mentioned before, due to the fact that we have only labeled the tight bounding boxes for truncated objects, it is not easy to these examples as the actual extents of them need to be estimated during training. Therefore, the set of "D2" and "SD" in the table are usually not used for experiments, while the other six are recommended. Briefly speaking, for these six sets, the distributions of the number of examples belonging to different semantic subsets are similar, though the total number of examples varies.

Table 3-1: Distributions of the number of objects in subsets with various difficulties. The capital 'S' and 'D' in the second column stand for "simple" and "difficult" respectively. The abbreviations are set notations, while the numbers in the parentheses are the corresponding visual difficulty labels. '1' means the difficulty presents while '0' means not, and 'x' means '0' or '1'.

Category	Set	Uns	F/Rr(B)	F	FLP (FCF)	Rr (B)	RLP	L/R	L	R	All
Car	S (00000)	388	167	233	84	265	100	20	253	190	1700
	D1 (10000)	136	24	34	4	36	3	25	57	74	393
	D2 (01000)	67	14	16	5	36	2	5	40	30	215
	D3 (00100)	24	14	12	11	21	4	1	6	2	95
	D4 (00010)	41	27	7	0	11	0	3	10	7	106
	D5 (00001)	28	14	14	7	27	6	0	17	5	118
	SD-D2 (x0xxx)	711	283	326	109	384	118	60	380	299	2670
	SD (xxxxx)	953	320	362	116	454	123	106	476	382	3292
Pedestrian	S (00000)	591	130	569	158	760	0	15	316	336	2895
	D1 (10000)	198	24	89	27	276	0	3	84	114	815
	D2 (01000)	41	3	50	12	43	0	2	23	18	192
	D3 (00100)	6	1	7	3	9	0	1	0	2	29
	D4 (00010)	265	164	91	1	75	0	5	62	51	714
	D5 (00001)	46	9	31	7	36	0	5	31	36	201
	SD-D2 (x0xxx)	1215	348	824	201	1219	0	35	507	562	4911
	SD (xxxxx)	1309	355	910	216	1302	0	37	544	602	5275

Training and testing data To make the IAIR-CarPed dataset ready for experiments and performance comparison, we split the whole image set (containing 3132 images) into two equally sized subsets for training and testing by random sampling. Therefore, each one of them includes 1566 images, and the numbers of objects for training and testing are roughly the same for both car and pedestrian. The indices of the images for training and testing are distributed along with the dataset. Note that all the subsets listed in Table 3-1 contain both training and testing examples.

3.5 Applications

The IAIR-CarPed dataset provides new opportunities for advancing the research on object recognition. One of them is object detection using only the bounding boxes and the category labels, and another is the opposite, i.e., within-category object classification using these deep and layered semantic labels assuming the bounding boxes of the objects are known, which is called *deep and layered object classification* in this thesis. We present some primary experimental results on these two applications with discussions on the inspirations they bring to us. Further more, we propose a new challenge on this dataset,

which is a combination of the first two problems, i.e., simultaneous detection and layered within-category classification. Due to its complexity, we discuss here only some possible issues towards solving it, leaving the solution itself for further exploration.

3.5.1 Object Detection

There are a lot of publicly available datasets for car and pedestrian detection, such as the widely used UIUC side-view car detection dataset [124], some multiview car detection datasets [48, 140] collected from other general purpose databases, the small scale pedestrian detection datasets collected in different scenarios [126–128], the recently proposed large scale pedestrian detection datasets captured in street scenes only [14, 129, 130], and the challenging unconstrained car and people data in the PASCAL VOC challenges [71]. Even though, the IAIR-CarPed dataset is still a valuable benchmark for car and pedestrian detection which cannot be replaced by the others, as it has three advantages: a) moderate size (thousands of objects in thousands of images) which is computationally desirable for most algorithms; b) sufficient but constrained intra-class variations (various environmental conditions, but relatively constrained poses); and c) separated visual difficulty labels which can be used to measure the performances in different ways.

Unlike the other datasets, the IAIR-CarPed dataset contains no specific negative images for training and testing which are considered by us to be unnecessary. Instead, one can use the sliding windows which have less than 50% overlap with the labeled ground truths as negative examples. Such a strategy not only directly differentiates the objects and their related backgrounds, but also takes into account the possible false positives due to misalignments, like the region within two pedestrians.

We propose a baseline algorithm using the two-layer HOG features (please refer to 4.7.2 for details) and the linear SVM classifier for object detection on the IAIR-CarPed dataset. For comparison, the state-of-the-art part-based deformable model [77], which has winning performances on the PASCAL VOC 2008 challenge dataset, has been adopted to test on it. We trained our baseline algorithm on set "SD-D2" which contains all the applicable object instances, and tested the trained model on six different subsets: "S", "D1", "D3", "D4", "D5" and "SD-D2". For the part-based deformable model, we used the pre-trained model on the PASCAL VOC 2008 challenge dataset to directly test on these six different subsets. By doing so, we can have a rough sense of the complexity of the IAIR-CarPed dataset compared to the INRIA person dataset [126] which is the closest competitor on pedestrian detection. Since the minimum size of the annotated objects in the IAIR-CarPed dataset is about twice smaller than the size of HOG templates, we upscaled the images by two times for training and testing.

The experimental results are shown in Figure 3-10, which is measured by the per-image Detection Error Tradeoff (DET) curves on a log-log scale, i.e, miss rate versus FPPI, as proposed in [14]. Such a per-image measurement is proved to be more reliable

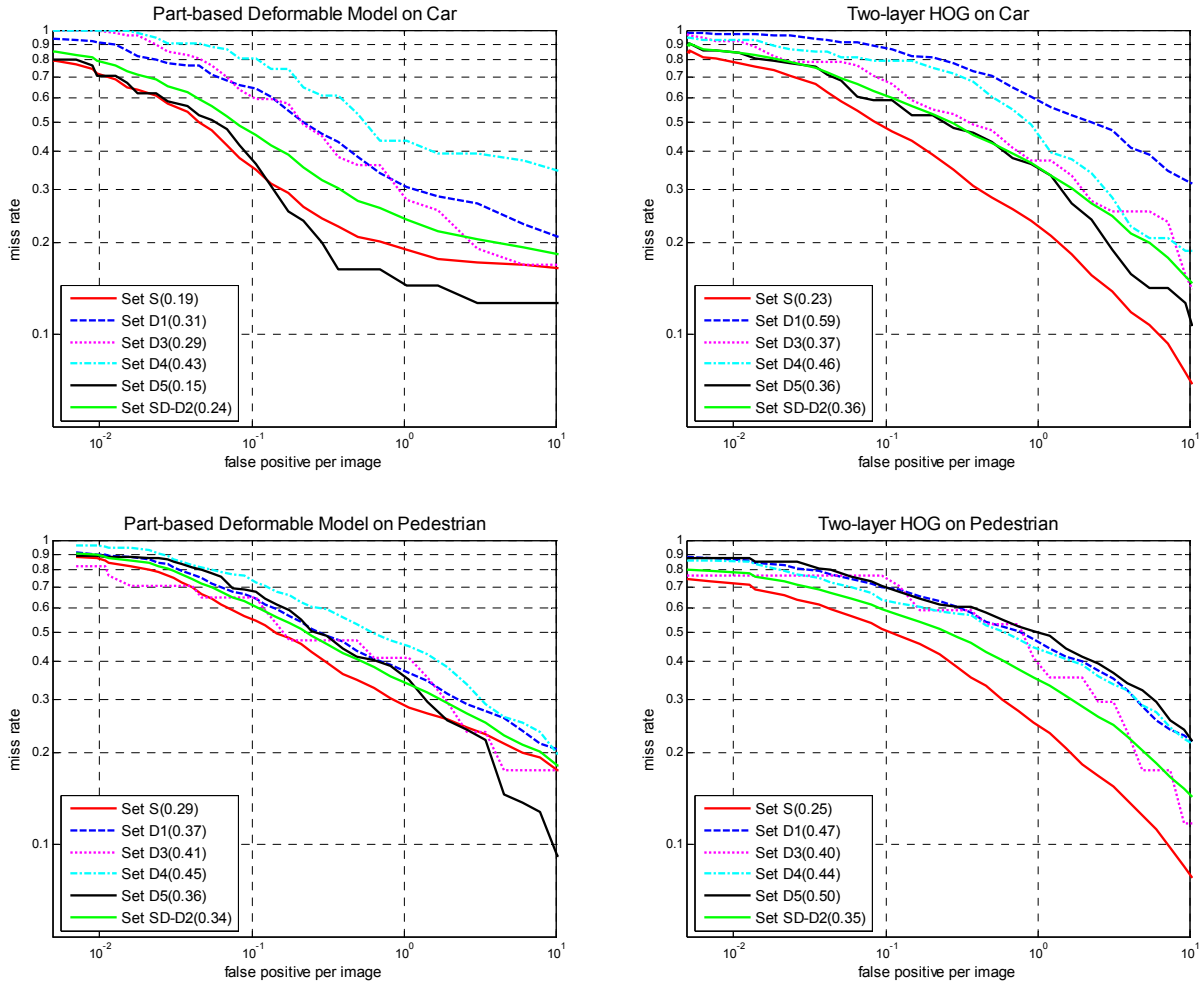

Figure 3-10 Object detection results measured by per-image DET curves.

and fair than the traditional per-window measurement. Besides the per-image DET curve, the Precision-Recall (PR) curves are also good for measuring the detection performance, which has been used in PASCAL VOC challenges [71]. However, as we want to compare the performance on different subsets which has different numbers of positive examples, the precision measurement will bias on larger subsets, and such a bias can results in significant disorder of the curves as shown in Figure 3-11. So that for the purpose of comparing detection performances of one algorithm on different positive subsets of the dataset, one should look into the per-image DET curves but not the PR curves. For the purpose of comparing different algorithms on the same dataset, however, both of these two measurements are applicable.

From the results in Figure 3-10, we can see that comparing to the baseline algorithm, part-based deformable model generalizes better to objects with unfrequent shape and occlusions, proving the benefit of using the latent part-based deformable model. Since both of these two algorithms are based on HOG features, they all suffer from the

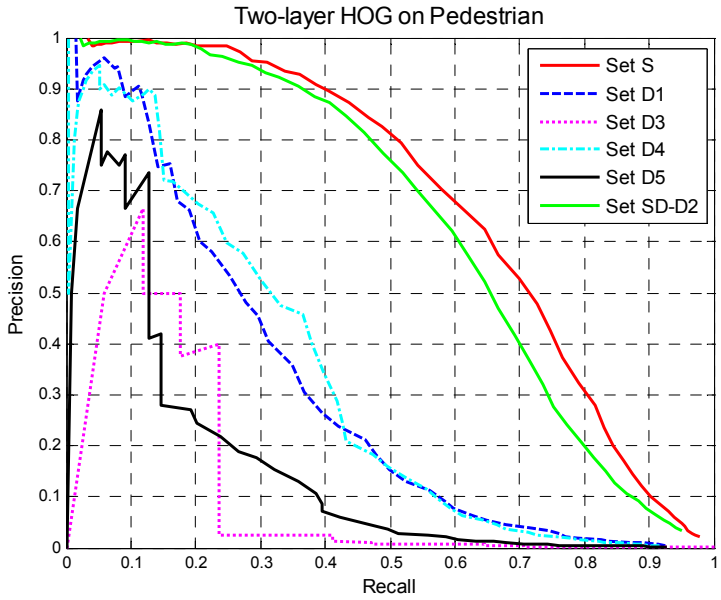

Figure 3-11 The disorder of the PR curves dues to different numbers of positive examples in the listed testing subsets. Since the precision is computed by the true positives vs. true positives plus the false positives, for a given FPPI of the model, it's more likely that those subsets with more positives have more true positives (i.e., higher precision).

low contrast difficulty which weakens the gradients. Though these two algorithms are not completely comparable as they were trained on different data, by comparing their results on car and pedestrian we can still get the inspiration that a divide-and-conqueror strategy can promote the performance of the model. The superiority of the part-based deformable model is more significant on cars than on pedestrians because the two components of the car model trained on PASCAL are all applicable to the IAIR-CarPed dataset while only one component of the two-component person model can be used to detect the full body pedestrians. Therefore, if a detection model can take the advantage of the deep and layered semantics provided by the IAIR-CarPed dataset, it may significantly boost the performance. This is one of the motivations for simultaneous localization and classification which is exactly the new challenge we are going to propose.

The pedestrian detection results of the part-based deformable model on the IAIR-CarPed dataset also demonstrate that it is a more challenging benchmark for pedestrian detection comparing to the INRIA person dataset. As presented in [14], the miss rate of the part-based deformable model at 1 FPPI on the INRIA person dataset is 0.21, while for subset "S" and subset "SD-D2" of the IAIR-CarPed dataset, it has a value of 0.29 and 0.34 respectively.

3.5.2 Deep and Layered Object Classification

This is a within-category multiclass classification problem, but it has an important characteristic compared to other similar problems: the classes have layered semantic

relationships between them. To model these structured relationships, we can use the maximum-margin based discriminative structured learning algorithms like SVMstruct [23]. We choose to use a relatively more efficient learning algorithm named SOnline (see 4.7.3) to do our experiments. It is an online learning algorithm which can also been used in a batch mode by repeating the online learning process several times until it converges or reaches the maximum number of iterations. For our experiments, a 3-epoch batch learning is adopted and it can generate reasonably good results. Since the number of training examples for this application is not very large, no training set updating is needed when using SOnline for learning.

About the loss function in the structured learning problem for classification, a detailed discussion can be found in 4.5.1. Both the zero-one loss $l_{0/1}$ and the confusion loss l_{Conf} are used in our experiments for comparison, in which the parameter η controlling the diversity of the confusion loss is set to 0.2. As discussed before, the adjusted confusions are better for learning a conservative model which is semantically more plausible. Therefore, we use the adjusted confusions on the training data for computing the confusion loss.

Usually for multiclass classification, the performance is measured by the error rate which is computed based on the zero-one loss. However, it may not be a good measurement for structured prediction problems like the problem we are working on, as different mistakes are not equally bad. Using the confusion loss instead is a plausible choice, as it represents the way in which humans tolerate the mistakes. To compare these two measurements, we computed both of them in our experiments.

Note that the dataset provides a hybrid semantic annotation which contains both orientational information and key part clearness information as shown in Figure 3-2. Though semantically correlated, these two types of information need to be represented by different features. The key parts we have chosen (license plates and faces) are very specific patterns, so their clearness (or equivalently presences) are better to be described by corresponding detectors, while the orientation of the object is better to be represented by other object-level features like global shape features. To make things easier and clearer, we choose to do some primary experiments on the orientational semantics (i.e., the first three layers) only. However, they can be easily extended to deal with all the semantics as long as the features and the loss function are computed.

We choose the well-known HOG features [126] as our data descriptor for this application, which has been proved to be good for representing cars and pedestrians. However, unlike the detection problem, the within-category classification needs to explore the shape subtleties between the semantics. Therefore, we use a dense HOG with 12×16 cells for cars and 24×8 cells for pedestrians without margins (as the margins may be useless for the within-category classification problem). For both efficiency and effectiveness, we use the PCA-HOG [77] instead of the original HOG in our experiments.

Baseline results By training and also testing on set "S" (using their corresponding

Table 3-2: Baseline deep and layered object classification results using different loss functions for training and measured by different testing losses. The columns are different losses for training while the rows are losses for performance evaluation.

	Car		Pedestrian	
Loss function	$l_{0/1}$	l_{Conf}	$l_{0/1}$	l_{Conf}
$l_{0/1}$	0.1957	0.2372	0.3663	0.3822
l_{Conf}	0.1680	0.1659	0.3217	0.2905

examples of it), we got the results as listed in Table 3-2. It can be seen that for both cars and pedestrians, training with l_{Conf} results in larger zero-one loss but smaller confusion loss in testing than training with $l_{0/1}$. Figure 3-12 shows the detailed confusions on the test data after training with these two different loss functions. Clearly, because the confusion loss biases on conservative recognition, the results of training with l_{Conf} have fewer aggressive mistakes but more conservative mistakes than the results using $l_{0/1}$. The experimental results indicate that by setting a proper loss function, the algorithm can generate recognition results with desired properties.

Note that cars have much smaller test errors (losses) than pedestrians, which means that cars have greater within-category shape variations (captured by HOG) due to orientation changes than pedestrians, which is visually provable.

Robustness to visual difficulties To use the difficulty labels for robustness analysis, we propose to train the model on set "S" and test it on the difficult subsets with only one visual difficulty in them, such as "D1", "D3", "D4" and "D5" ("D2" is not used due to its special ground truth and performance measurement), in comparison to the test performance on set "S".

Table 3-3: Robustness of the baseline deep and layered classification algorithm against individual visual difficulties. The model is trained using l_{Conf} on set "S". Performance on set "S" is presented for comparison

	Car					Pedestrian				
Test set	S	D1	D3	D4	D5	S	D1	D3	D4	D5
$l_{0/1}$	0.2372	0.4592	0.3469	0.3962	0.3061	0.3822	0.4296	0.2941	0.4971	0.6147
l_{Conf}	0.1659	0.3146	0.1938	0.3430	0.1357	0.2905	0.3150	0.1748	0.4086	0.4598

Table 3-3 shows the results using the dense HOG features and training with the confusion loss. The results inform us that the robustness to certain difficulty not only depends on the feature representation, but also depends on the object category, i.e., the data itself. For the four different difficulties we are interested in, low contrast (in set "D4") consistently reduces the performance of HOG features as it weaken the gradients, while occlusion (in set "D1") has much larger influence on cars than pedestrians which

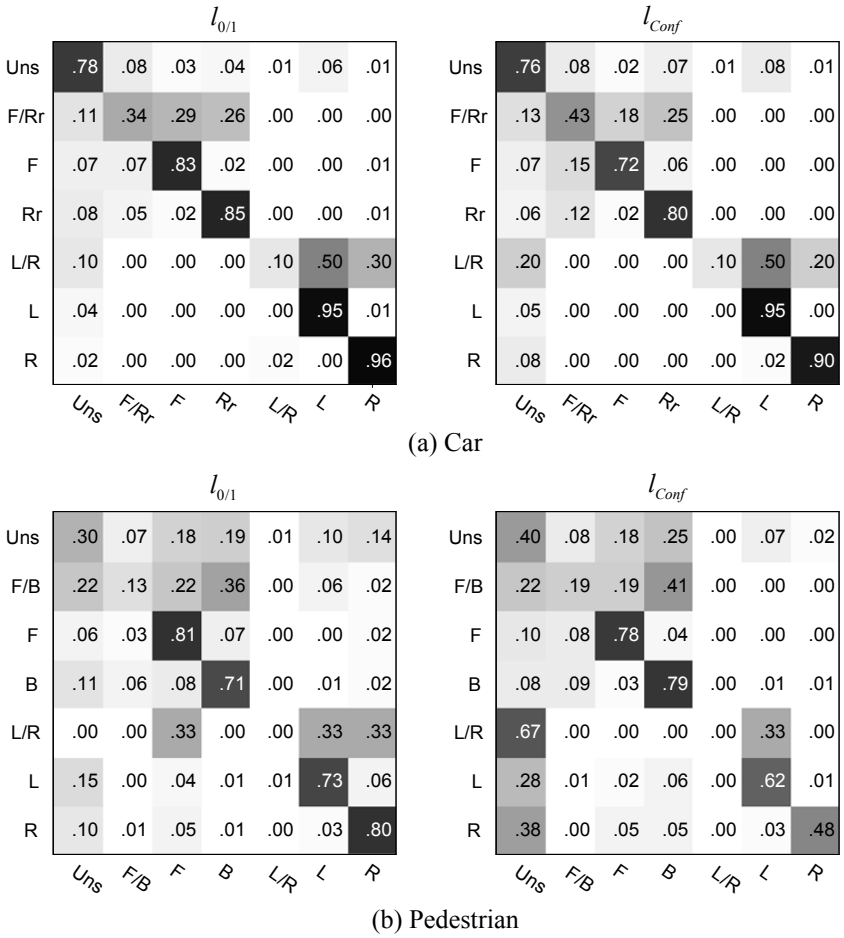

Figure 3-12 The confusions in the test results using the two different loss functions. It shows that comparing to the flat multiclass classification, structured learning using the relatively more conservative confusion loss can lead to more conservative predictions.

may suggest that the normal occlusions of these two categories are not equally serious. Note that the occlusion defined by us does not contain truncation, so that cars are usually occluded by pedestrians or trees which may influence the orientation classification a lot, while pedestrians are more often to be occluded by bags which has little impact on judging the orientation. The other two visual difficulties have opposite impacts on these two categories: nonuniform illumination (in set "D3") hurts cars a lot but little to pedestrians, while unfrequent shape (in set "D5") is a big problem for pedestrians but not as disturbing for cars. Such contrasting results mainly due to the properties of the data but not the features and the algorithms. Nonuniform illumination has so few examples for pedestrians (as shown in Table 3-1) that its results are not statistically meaningful, while the unfrequent shape for cars is not challenging as it stands for those vehicles having very similar shape to sedans.

This is just a demo on how these difficult subsets can be used, it will be more meaningful if different features and/or different algorithms are compared on them to reveal the differences between them, which is open to the public.

How much can feature representation improve? From the results presented above, we can see that this deep and layered recognition is really challenging. As we have only explored the dense HOG features, one may doubt that tuning the parameter of HOG or using a hierarchical feature representation may improve the performance. But how much can it improve? We present here more experimental results using different HOG features as shown in Table 3-4, in which "Coarse HOG" has only 6×8 cells for cars and 12×4 cells for pedestrians which is very close to the usual settings for detection [77] while "Fine HOG" stands for the dense HOG used before, and "Two-layer HOG" means using both of them. It can be seen that using finer HOG can only reduce the error rate by no more than 1 percent while combining both coarse and fine HOG can get another 1 percept improvement (showing that they are complementary). Therefore, to get a significantly performance improvement, one has to introduce other features which is complementary to HOG features. On the other hand, designing better learning algorithm may also help.

Table 3-4: Results of deep and layered classification using various HOG features. Since l_{Conf} is a better performance measurement than $l_{0/1}$, the results are compared using l_{Conf} only.

	Car			Pedestrian		
Feature	Coarse HOG	Fine HOG	Two-layer HOG	Coarse HOG	Fine HOG	Two-layer HOG
l_{Conf}	0.1677	0.1659	0.1530	0.3009	0.2905	0.2806

3.5.3 A New Challenge: Deep and Layered Object Recognition

Though the two applications presented above are valuable research topics for object recognition and the dataset can serve as a good benchmark for them, we are interested in a uniform recognition framework whose desired outputs are exactly what have be annotated, i.e. simultaneous object detection and deep and layered object classification. We call this uniform object recognition problem **deep and layered object recognition**, which contains both category-level recognition and layered within-category recognition. We believe that such a problem with both these two components is more promising than problems with either one of them, because these two are highly correlated and they may help each other.

For solving the brand new object recognition problem, the feature representation has to be powerful enough to distinguish the objects from their backgrounds while at the same time differentiate the within-category subsets with nonidentical semantics. The learning algorithm should be able to make use of the relationships among these semantics, and train the model in an efficient way since the task is more complex than the others. Moreover, the relationship between objects and their key parts should be properly modeled and the features for representing them should be somehow combined.

Note that there can be any number of interested objects in a single test image from zero to many, and the performance should depend on both the classification and localization results.

Though the category-level labels and layered within-category labels of the dataset are collected through separate annotation processes for practical considerations, the human vision system seems to be able to output these two types of labels simultaneously. Therefore, designing a computer algorithm which can do the same task will be very useful.

3.6 Conclusion and Discussion

The ultimate goal of object recognition is to *understand* the colorful objects within the visual world. To achieve this goal, the recognition should not be limited to localization, categorization and within-category classification, but better to be modeled for adaptive interpretation which can choose the fittest semantic label for each object from a layered semantic set which contains both categorial and non-categorial semantics. However, currently there is no object recognition dataset designed for such a purpose. Therefore, we introduce the IAIR-CarPed dataset for deep and layered object recognition with carefully collected human annotations on two representative categories: car and pedestrian.

The IAIR-CarPed dataset distinguishes itself from the others by two major properties: a) deep and layered human object recognition results with inspiring confusions of human vision, and b) detailed difficulty labels which can be used to evaluate the robustness of the algorithm/system against individual factors. The first property makes the dataset a benchmark for learning and evaluating deep and layered recognition algorithms, while the second one fractionizes the dataset for in-depth study of the generalization ability of the algorithms.

Three typical applications of this dataset have been discussed with primary experimental results on the first two of them. The first application on object detection is a byproduct of the dataset, but it is contributive to researchers working on detecting cars and pedestrians as it provides a moderate dataset with some new properties that other detection datasets do not have. Experiments on this application show that this dataset is more challenging than the well-known INRIA person dataset, and within-category classification may help detection. The second application on deep and layered object classification is a simplified version of the third application on deep and layered object recognition. Even though, we got some valuable results from the primary experiments on it: a) the loss function can significantly drive the recognition results, and the confusion loss based on human confusions is more plausible than the zero-one loss for both the training and the evaluation of deep and layered classification; b) the difficulty labels of the dataset can reveal the robustness of visual features to different variations. Though

we haven't presented experimental results on the third application, hints on designing algorithms for solving it have been given. Exploring effective and efficient algorithms for such an application will be our future work.

Due to the semantics we have chosen, the annotation based on human rapid recognition is informative but also expensive. IAIR-CarPed took about 200 person hours for the semantic labeling in controlled environment. For generalizing the idea to many other object categories, we can design new less demanding semantics for them and put the annotation tool on the Internet for collecting the labels from the public. Possible solutions may be making the annotation a game like the ESP dataset [122], or using the service of Amazon Mechanical Turk (AMT) like the ImageNet database [12]. Anyway, for both research purpose and real applications, IAIR-CarPed is a proper starting point, and we hope that it will open a door to the new exciting research field on deep and layered recognition.

To summarize, the IAIR-CarPed dataset is a proper benchmark for the research on both the newly proposed deep and layered object recognition (including deep and layered classification) and traditional object detection, and it is suitable for the robustness evaluation of algorithms/systems. The limitation is two-fold: it contains only two representative object categories: car and pedestrian, and the layered semantics are restricted to the orientation of the object and the clearness of the key parts.

CHAPTER 4
Deep and Layered Object Recognition

In this chapter we present a generic definition of Deep and Layered Object Recognition (DLR), followed by detailed discussions on practical issues of modeling, learning, evaluating and implementing it in computer vision. A case study on recognizing cars and pedestrians in the IAIR-CarPed dataset with deep and layered semantics shows the effectiveness of the proposed model and demonstrates its superiorities against traditional object recognition models.

4.1 Motivation and Contribution

When talking about object recognition, we are no strangers to the amazing functionality of animals including ourselves, and the challenging problem we are working on for decades in both computational neural science and computer vision. A picture is worth a thousand words. There is much information embedded in even a single still image and it holds for many common objects like cars and pedestrians. Cars and pedestrians may have various appearance and they may appear at different places under changeable illumination conditions. However, we have no problems living in the world with tremendous amounts of such objects. How can we do this? What information do we get from different instances of the same object category when we see them in the real world?

To make things easier, neural scientists paid special attention to the so called feedforward rapid recognition in primate visual cortex [5, 152], which is thought to be a good simplified model for understanding human recognition, and one example of simulating it in computer vision is proposed in [4]. Though most existing cognition experiments on rapid recognition are about object presence classification and category-level recognition, our new experiment presented in chapter 3 shows that a 20-50 ms rapid stimulation period can result in deeper and more layered recognition results. It seems natural for our humans to get the deep and layered interpretations of the visual objects rapidly and effortlessly, and such a recognition manner works well in our daily life.

However, the existing problems about object recognition we are trying to solve in computer vision are not like that. We may work on one of several concrete recognition tasks for understanding an image. One can categorize the whole image into predefined groups, like the recent work presented in [64] and [51], which are two of many papers on such a task. One can localize or detect the specific object instances (like cars) in images [77, 148, 153]. One can go further to do multi-class classification by assigning quantized direction labels to the objects, which in substance is the same as multi-class

object categorization problems [154, 155]. Beyond that, one can also identify every object instance [156, 157]. Besides the work on these separate recognition tasks, some others have dug into the combined problems like recently proposed concurrent object localization and classification or state estimation [106, 114–117]. Compared to the flexible deep and layered recognition results of humans, the outputs of current object recognition techniques look much more stiff and pallid.

The common character of the above efforts is that we are trying to eliminate the variances of object instances and assign category-level semantics to them (except the identification problem), more theoretically, to model the constancy of recognition. It is reproachless to do so and a good recognition system should have the constancy property. We can also ask the subjects to output the category label of the objects directly, and we will find that human beings do have very good recognition constancy. But, do they only see the category information? Or do they extract invariant and discriminative features for a specific category and put them into a classifier to make the decision directly? If so, we should be able to describe such features as the reasons for making such a decision, and then we have already solved the recognition problem. The fact is, usually we cannot figure out any features robust to arbitrary object variations in the real world. Instead, we may argue that we make the decision because we see this feature and it happens only when the object instances of this category present in such a state as shown in this picture and no others. The underlying magic is that we use the "divide and conquer" principle. That's why we can easily perform recognition but very hard to find an invariant feature. As argued in [158], low in-class variability leads to better detection performance for a classifier. If we ask a classifier to handle very high intra-class variability (like the class of "person" under all conditions), the performance won't be very good no matter how hard we try it, see the latest results in PASCAL VOC Challenge [159].

To learn more from human vision about recognition, we have to go deeper to interpret the visual differences of objects, and make the right tradeoff between inter-class discrimination and intra-class representation. We find that the key is to make the recognition deep and layered. Different object instances may have different appearance which expose different information to us. If you can clearly tell the key part of the object (like the license plate of a car) and be sure to its direction, you shouldn't just say that you only got its category label. As the opposite, if you can hardly tell the details of the object, concrete intra-class categorization will be risky. It's better to recognize it as it is without promotion (trying to extract subtle features from small, noisy, cluttered, and weakly-illuminated image regions where the interested objects present) and degradation (trying to ignore salient details from large, clear, clean, and well-illuminated visual objects). Then the results may be more reliable and useful. This is much closer to the way human beings perceive and recognize objects, and we call it ***deep and layered recognition*** in this thesis, where "*deep*" means detailed within-category interpretation to distinguish it from traditional categorization, and "*layered*" indicates that the inter-

pretation is semantically layered, not biasing on the deepest within-category semantics.

The contributions of this work include the following three:

- proposing a general computational model for DLR based on structured learning and prediction;
- presenting two feature representation strategies along with concrete examples for DLR;
- providing an efficient structured online learning algorithm (SOnline) to train the proposed DLR model.

In the following sections, we first discuss how others' work inspired us and helped us with the implementation details of DLR, then we present our definition of DLR and some brief ideas towards solving it. The major issues of modeling, learning, evaluating and implementing DLR are discussed thereafter, followed by a concrete case study on recognizing cars and pedestrians with the implementation details. Experiments on the IAIR-CarPed dataset demonstrate two typical advantages of DLR against traditional category-level recognition models:

- *Rich functionalities and adaptive interpretation.* Instead of predicting only the category-level labels, DLR can output much richer recognition results such as subcategory labels, part information, object attributes and so on, and it is adaptive. This is done by making proper use of different information in the original image and systematically manipulate the relationship between the input feature representation and the output labeling structure.
- *Improved performance on category-level recognition.* By making a proper trade-off between intra-class differentiation and inter-class collaboration of object instances belonging to the same category, DLR can make better use of the training examples and achieve better performance on category-level recognition than former recognition approaches.

4.2 Related Work

We are aiming at humanizing recognition of objects based on their visual appearance. In general, the output interpretation can be category labels, subcategory labels, locations, identities, and any attributes or descriptions of the objects, including the semantics for their parts. The key is that the recognition should show its respect to the data, assigning proper labels or states to various object instances even when they present in the same scene or image. As far as we are aware, there are no sufficiently overlapped ideas or work proposed. However, there is a lot of work which has ever inspired us and given us much help on figuring out some of the details.

Deep within-category semantics have recently been investigated by several research groups. Specially, attributes have been proven to be helpful for category-level recognition, as shown by Lampert et al.[138] who looked into the problem of between-class attribute transfer for recognizing new object classes with no training examples, while at the same time Farhardi et al.[139] shifted the goal of recognition from naming to describing, which significantly improves the categorization performance and also enables learning new categories. Besides attributes-based approaches, there are some other prior researches on within-category classification/description. The work on multiplicative kernels [140] manages to classify the view angles of cars in a dataset collected from LabelMe database [134], and another work directly targeting at within-object classification [141] represents results on three different databases. Though our DLR also involves describing the attributes of objects and classifying the objects into subcategories, we are neither trying to use the attributes as the middle-level representation for categorization nor aiming at attribute inference or within-category classification. Instead, our recognition results are layered, which may and also may not be sub-categorial semantics with attributes.

Meanwhile, semantic hierarchies have also been widely researched in the past few years on visual recognition. Marszalek and Schmid[160] proposed a semantic hierarchical classifier that uses the semantics of image labels to promote multiclass object detection performance. Binder et al[161] tried to classify images into a given, pre-determined taxonomy using a structured learning framework. Their empirical results on Caltech256 and PASCAL VOC2006 data show that the algorithm exploiting the structure of the taxonomy outperforms multi-class classification approaches. Zweig and Weinshall [162] demonstrated that training a single classifier using object hierarchies obtained from publicly available datasets can significantly improve multi-class object recognition results. Kapoor et al[163] combined kernels designed at each level of a object class hierarchy to benefit both computational efficiency and performance in Caltech 101 multi-class categorization. However, almost all these existing papers are limiting themselves to the object category and the levels above, but not for exploring object subcategories and attributes to better describe an object instance. There are also some work on exploring both the object-level information and the information of its parts, such as the research on the combination of bottom-up and top-down processes proposed by Yang et al. [120]. Though they have combined the information from three different levels, their goal is to interpret the middle level only. Chen et al[164] aimed at simultaneous object detection, segmentation and part parsing using a rapid inference algorithm on a specifically designed AND/OR graph, however, the three types of outputs are forced to be predicted.

In the learning part, we are facing the same problem of outputting multilabel or structural labels as the above referred work and many other structure learning applications. Specifically, we are interested in a principle structure learning framework which can model the problem in a compact way and solve it efficiently. Tsochantaridis et al

[165] investigated large margin methods of doing so. Bordes et al[24] proposed an efficient online learning algorithm for multiclass classification problems in a maximum margin discriminative learning framework. Recently, Wu et al[166] extended this framework to handle more general structural output prediction problems and showed experimental results on digits recognition. Lampert and Blaschko[167] proposed to use the joint kernel directly for structured prediction by an alternative model named Joint Kernel Support Estimation (JKSE).

On the side of data representation, we would like to follow the two-layer HOG representation of objects in [77], but without latent part localization for efficiency. Using histograms of different scales in a fixed form is also effective, as discussed in [87]. Beside that, colors for special parts (like faces) can also be informative for recognizing humans and cars, and we got the inspiration of using it from [168] and [87]. The widely used SIFT features [8] are very good at representing rigid objects with view changes, therefore we use it for representing the license plate, which is the key part of the car.

4.3 Definition

In general, Deep and Layered Recognition (DLR) is properly interpreting the visual data based on its actual appearance by assigning deep and layered semantics and geometric label(s) to it. The semantic label space should contain some within-class categorization labels representing meaningful and case-dependent details, and the interpreted labels vary in both level and amount, depending on the data.

Mathematically, given data $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is the input data space, DLR is to find the labeling $\mathbf{y} \in \mathcal{Y}$ that best interpret the data $\mathbf{x}$. Suppose $g : \mathcal{X} \mapsto \mathcal{Y}$ denotes the mapping of DLR. The differences between DLR and traditional object recognition strategies are two-fold:

- **Semantic part of $\mathbf{y}$**. In traditional recognition, the semantic part of $\mathbf{y}$ usually represents either category or subcategory information of the object, while in DLR it represents both category and subcategory information, and may also include semantic information of the object parts.

- **Flexibility of $\mathbf{y}$**. Unlike traditional recognition strategies which demands all the information represented by $\mathbf{y}$ to be assigned (e.g. a concrete pose of the object), DLR may assign only some of the desired information, indicating the rest is unsure. By doing so, it is not forcing the data to fit the outputs, but exploring and interpreting the data to assign case-dependent labels. It is trying to bring the model closer to the data than before. The recognition result may depend on the scale, view, illumination, pose, spatial occupation and other aspects of the object and its context.

Note that in general the input data $\mathbf{x}$ can be an object region, an image, or a video sequence, depending on the visual task. While at the same time, the output label vector $\mathbf{y}$ can be designed accordingly.

Though by definition DLR can be only the recognition of the objects themselves, we are interested in simultaneously interpreting objects and their parts. Furthermore, we care about not only the data-dependent semantics of the objects and their parts, but also the geometry information of them, which can be treated as the category-level recognition component. Undoubtedly, the gap between the input data and our desired output interpretation is very large. We are putting efforts on three different aspects that are considered to be critical for bridging the gap as shown in Figure 4-1, and we will show concrete ways on realizing them hereinafter. These three aspects are as follows:

- **Output space**. Compared with traditional object recognition problems, DLR has richer output states which need to be properly designed so that the learning can be done efficiently.
- **The prior**. It represents the structure of the output space, which can be modeled by the losses of confusing different output states. The prior directly influences the performance of the classifier.
- **Feature representation**. Compared with traditional object recognition problems, DLR has richer output states, so how to make the feature representation both effective and efficient is rather important. When the output space is structured, the feature representation has to take into account of the sharing and discrimination abilities of the features simultaneously, making a good trade-off between them.

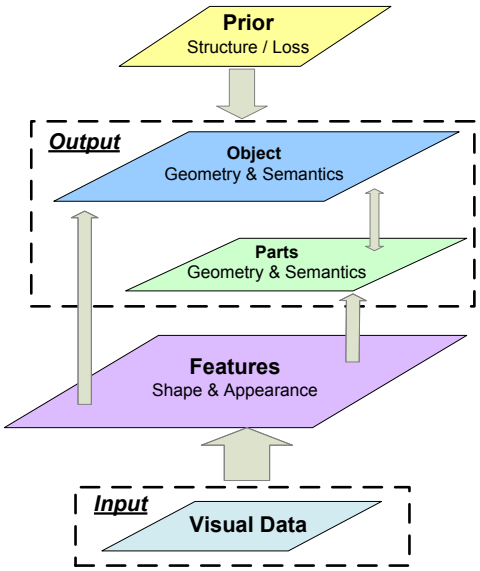

Figure 4-1 A brief framework of solving DLR.

4.4 Modeling and Learning Framework

4.4.1 Modeling

Suppose for each object, there are at most n parts of it need to be recognized, then we can define the output vector as $\mathbf{y} = (\mathbf{y}_O; \mathbf{y}_{P_1}; \ldots; \mathbf{y}_{P_n})$, where $\mathbf{y}_O, \mathbf{y}_{P_1}, \ldots, \mathbf{y}_{P_n} \in \mathcal{Y}_{entity}$ are the output vectors for the object and its n parts respectively. Since they are all entities that may have similar recognition demands (e.g. semantic categorization and geometric localization), their outputs are briefly denoted as belonging to the same type of output space $\mathcal{Y}_{entity}$ for convenience and consistency. However, in practice, their concrete content and domains may be different. Let $\mathbf{y}_{entity} \in \mathcal{Y}_{entity}$ be an arbitrary entity (the object itself or one of its parts), we define two types of outputs on it: $\mathbf{y}_{entity} = (\mathbf{y}^C_{entity}; \mathbf{y}^L_{entity})$, where $\mathbf{y}^C_{entity} = (y^C_1, \ldots, y^C_m)^T$ denotes the categorization information and $\mathbf{y}^L_{entity} = (t, l, b, r)^T$ denotes the localization information (top, left, bottom and right of the bounding box). Note that $m \geq 1$, and each y^C_j is a microlabel of $\mathbf{y}^C_{entity}$. We let y^C_1 stand for the presence of the entity (i.e., $y^C_1 = 1$ denotes the presence of the entity and $y^C_1 = 0$ the absence) and the others as within-class semantic microlabels. In general, these semantic categorization microlabels can be organized in a structure. An example of such semantic structures is shown in Figure 4-2.

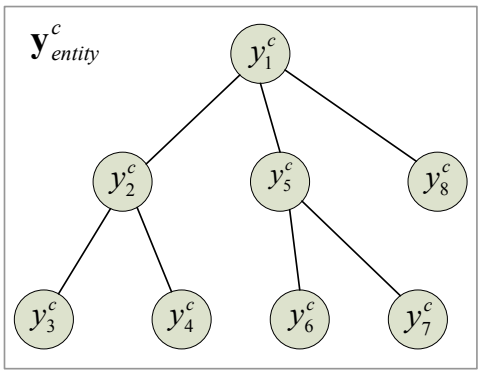

Figure 4-2 A typical hierarchical semantic tree for categorization.

For an arbitrary input $\mathbf{x} \in \mathcal{X}$, the mapping of DLR can be defined as:

$$g(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \tag{4-1}$$

where $f(\mathbf{x}, \mathbf{y})$ is a compatibility function between the input and the output. A common definition of the compatibility function is the inner product between the model parameter vector $\mathbf{w}$ and the joint input-output feature $\Phi(\mathbf{x}, \mathbf{y})$:

$$f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle . \tag{4-2}$$

And one can also formulate it as a conditional probability function:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{w}, \mathbf{x})} \exp\{\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle\}, \tag{4-3}$$

then the mapping $g : \mathcal{X} \mapsto \mathcal{Y}$ becomes a maximum a posteriori (MAP) estimation. As argued in [169] and [166], the later one usually requires an extra computation of the normalization term $Z(\mathbf{w}, \mathbf{x})$ which might be computationally expensive or even unbounded. No matter which one is chosen, the joint feature $\Phi(\mathbf{x}, \mathbf{y})$ is the key.

Since the optimization of the equation 4-1 requires searching the global optima in the output space $\mathcal{Y}$, the feasibility and efficiency of the learning and prediction algorithm will depend much on the structure of it. When the output contains both the object itself and its semantic parts, and each of them has both the structured categorization and localization information to be predicted, then it will be a big challenge. Note that this is unlike methods that divide the whole problem into several pieces and optimize them in a sequential procedure, it is about global optimization taking into account of the contextual relationships among the semantic entities (the objects and their parts).

To make things easier, we can narrow down the problem a little bit by decoupling the contextual relationships among object parts and use a star model instead to capture the pictorial structure of object (i.e., the relationships between the object itself and each of its parts). This is inspired by [95] and [77]. Formally, we decompose the joint feature as:

$$\Phi(\mathbf{x}, \mathbf{y}) = (\Phi_O(\mathbf{x}, \mathbf{y}_O); \Phi_{P_1}(\mathbf{x}, \mathbf{y}_{P_1}, \mathbf{y}_O); \ldots; \Phi_{P_n}(\mathbf{x}, \mathbf{y}_{P_n}, \mathbf{y}_O)). \tag{4-4}$$

In which, the global object-level joint feature is

$$\Phi_O(\mathbf{x}, \mathbf{y}_O) = \Phi_O(\mathbf{x}, \mathbf{y}_O^C, \mathbf{y}_O^L), \tag{4-5}$$

which means that it depends on both the semantic categorization structure and the object's position in the data. For each of the object part P_i, the joint feature can be further decomposed into two separate terms:

$$\begin{aligned} &\Phi_{P_i}(\mathbf{x}, \mathbf{y}_{P_i}, \mathbf{y}_O) = \Phi_{P_i}(\mathbf{x}, \mathbf{y}_{P_i}^C, \mathbf{y}_{P_i}^L, \mathbf{y}_O^C, \mathbf{y}_O^L) \\ &= (\Phi_{P_i}^C(\mathbf{x}, \mathbf{y}_{P_i}^C, \mathbf{y}_{P_i}^L, \mathbf{y}_O^C); \Phi_{P_i}^L(\mathbf{y}_{P_i}^L, \mathbf{y}_{P_i}^C, \mathbf{y}_O^L, \mathbf{y}_O^C)) \end{aligned} \tag{4-6}$$

where $\Phi_{P_i}^C(\mathbf{x}, \mathbf{y}_{P_i}^C, \mathbf{y}_{P_i}^L, \mathbf{y}_O^C)$ is the categorization feature, while the later $\Phi_{P_i}^L(\mathbf{y}_{P_i}^L, \mathbf{y}_{P_i}^C, \mathbf{y}_O^L, \mathbf{y}_O^C)$ indicates the localization feature for the contextual relationship between the object and its part P_i, which not only depends on the localizations of these two, but also depends on their categorization information, saying that different categorization combinations may have different contextual relationships. This formulation has the same flavor as the latent model introduced in [170] and [77], in which the model can be viewed as a simplified instantiation of the above joint feature representation. Their object categorization is a binary object/non-object label while the parts are forced to appear without further categorization since they are not semantically meaningful. Nevertheless, the latent SVM algorithm for simultaneous optimizing object recognition and its parts' localization is very impressive.

4.4.2 Learning

The two semantic outputs $\mathbf{y}^C$ and $\mathbf{y}^L$ of our model form a complex but usually structured space for both learning and prediction. Therefore, structured learning algorithms [6] should be adopted to solve this problem efficiently. As argued in [169], though structured output prediction starts from multiclass classification problems, they have significant differences.

In multiclass classification, every possible output state corresponds to one class and the learning algorithm trains separate parameters for it, while in structured prediction the joint feature map may take advantage of the output structure by decomposing the combinatorial space into much fewer substructures, resulting in fewer parameters and better generalization ability. For example, a tree-like structure can represent a semantic hierarchy or a taxonomy, which can be decomposed into pair-wise edges for efficient learning and inference as presented by [171]. There can also be some loose structures like the partial order of localization coordinates of objects and/or object parts. A clever branch-and-bound algorithm is introduced in [148] to solve it efficiently when the feature representation is additive with respect to spatial extension.

Besides efficiency and generalization ability, we would like to argue that structured prediction may take into account of the intrinsic structure of the data (represented by the input-output relationships) and therefore result in semantically more plausible predictions and better recognition performances. This is also one of the main reasons for choosing structured prediction algorithms for DLR.

The modeling of DLR in 4.4.1 already refers to the two main types of structured prediction methods: discriminative methods based on equation 4-1 and 4-2, and generative methods induced by equation 4-1 and 4-3. Both of them can solve this problem using various optimization methods, however, in this paper we focus on the discriminative ones.

Given the training set $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, and the definition of joint features in equation 4-4, we can construct a structured max-margin optimization objective function constrained by partial ranking:

$$\begin{aligned} &\min_{\mathbf{w}} \left\{ \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i \right\} \\ &s.t. \quad \begin{cases} \langle \mathbf{w}, \Delta\Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq l(\mathbf{y}_i, \mathbf{y}) - \xi_i \quad , \quad \forall i,\ \mathbf{y} \neq \mathbf{y}_i \\ \xi_i \geq 0 \ , \quad \forall i \end{cases} \end{aligned} \tag{4-7}$$

where $\Delta\Phi(\mathbf{x}_i, \mathbf{y})$ stands for $\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \mathbf{y})$, and $l(\mathbf{y}_i, \mathbf{y})$ is the overall loss function. Different values of the loss function (depending on the two outputs) indicate different demands on the margins of partial ranking. This is a general formulation for max-margin based structured prediction. The learning is to find a weight vector $\mathbf{w}$ that when the joint features of each training example project on to it, the one with the correct output is larger than any other output by a margin represented by the mistaking loss defined on

these two outputs. Specifically, this formulation is called margin-rescaling, and it's said to be simpler for optimization than its slightly different alternative the slack rescaling formulation (see [169]). Their differences are out of the scope of this paper, therefore the slack rescaling formulation is not presented here.

For our DLR problem, which has specific definitions on the joint feature, the partial ranking of the above formulation becomes:

$$\begin{aligned} &\langle \mathbf{w}, \Delta\left(\Phi_O(\mathbf{x}_i, \mathbf{y}_O); \Phi_{P_1}(\mathbf{x}_i, \mathbf{y}_{P_1}, \mathbf{y}_O); \ldots; \Phi_{P_n}(\mathbf{x}_i, \mathbf{y}_{P_n}, \mathbf{y}_O)\right)\rangle \\ &\geq l\left(\mathbf{y}_i, \mathbf{y}\right) - \xi_i \quad , \quad \forall i,\ \mathbf{y} \neq \mathbf{y}_i \ . \end{aligned} \tag{4-8}$$

Decompose the weight vector into that of the object and its parts, i.e. $\mathbf{w} = (\mathbf{w}_O; \mathbf{w}_{P_1}; \ldots; \mathbf{w}_{P_n})$, then the above nonequal constraints become a decomposed version:

$$\begin{aligned} &\langle \mathbf{w}_O, \Delta\Phi_O(\mathbf{x}_i, \mathbf{y}_O)\rangle + \textstyle\sum_{j=1}^{n} \left\langle \mathbf{w}_{P_j}, \Delta\Phi_{P_j}(\mathbf{x}_i, \mathbf{y}_{P_j}, \mathbf{y}_O)\right\rangle \\ &\geq l\left(\mathbf{y}_i, \mathbf{y}\right) - \xi_i \quad , \quad \forall i,\ \mathbf{y} \neq \mathbf{y}_i \ . \end{aligned} \tag{4-9}$$

When the loss can also be decomposed into that of the object and its parts, the constraints are:

$$\begin{aligned} &\langle \mathbf{w}_O, \Delta\Phi_O(\mathbf{x}_i, \mathbf{y}_O)\rangle + \textstyle\sum_{j=1}^{n} \left\langle \mathbf{w}_{P_j}, \Delta\Phi_{P_j}(\mathbf{x}_i, \mathbf{y}_{P_j}, \mathbf{y}_O)\right\rangle \\ &\geq l_O\left(\mathbf{y}_{Oi}, \mathbf{y}_O\right) + \textstyle\sum_{j=1}^{n} l_{P_j}\left(\mathbf{y}_{P_j i}, \mathbf{y}_{P_j}\right) - \xi_i \quad , \quad \forall i,\ \mathbf{y} \neq \mathbf{y}_i \ . \end{aligned} \tag{4-10}$$

Put equation 4-5 and equation 4-6 into the above nonequal constraints, we can get further decomposed ones by separating categorization and localization. Due to the space limit, the unfolded expression is omitted.

In general, the above formulation is a latent version of the structured output prediction because of the embedded localization of the object and its parts. In this case, usually an iterative method is needed to alternate between the optimization of the feature weights $\mathbf{w}$ and that of the locations $\mathbf{y}^L$. As presented in [7] and [77], a clever formulate is needed to make these two optimizations computationally feasible and efficient. [7] and [148] propose an efficient branch-and-bound searching technique on bag-of-words features, while [77] made the part localization features quadratic to ensure the semi-convexity of the optimization objective. Since both our categorization and localization (object and its parts) are designed to be much more complex than theirs, the overall optimization will be accordingly much harder. Therefore, in this work we would like to simplify the localization but emphasize the adaptive categorization to make the idea of DLR start to be realized. Concretely, we use the traditional sliding window approach to localize the object and use separate part detectors to localize object parts, details of which is presented in 4.7.1. However, as the way we formulate the DLR problem, we are very interested in a direct and efficient global optimization, which is open to the community and will also be our future work.

4.5 Evaluation

To evaluate the performance of DLR, the predicted results of test samples need to be compared to their ground truths. Usually the overall recognition performance is a function of the individual predictions. From the learning point of view, it is best to directly optimize the recognition performance during the training process. Therefore, the evaluation criterion needs to be designed together with model learning. A common choice is to use a loss function $l(\mathbf{y}_i, \mathbf{y})$ to guide the learning of training samples $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, N\}$as in structured learning algorithms. Then the performance measurement is designed accordingly, for example a simple sum of predicted losses.

Since the output $\mathbf{y}$ may contain two different semantic information: the categorization $\mathbf{y}^C$ and the localization $\mathbf{y}^L$, we discuss different ways of designing the proper loss functions for them in the followings.

4.5.1 Categorization Loss

As discussed in 4.4.1, the categorization of an object or an object part may be a vector $\mathbf{y}^C = (y_1^C, \ldots, y_m^C)^T$ indicating the values of different semantic microlabels. Therefore, the loss of mistaking two different categorization outputs (e.g. from $\mathbf{y}_i^C$ to $\mathbf{y}^C$) are usually defined on these microlabels. We also introduce here a new loss function which goes beyond such a tradition by focusing directly on the probability of semantic mistakes. Details of the publicly available losses and the new one to be introduced are discussed in the following.

- **Zero-one loss**:

$$l_{0/1}^C(\mathbf{y}_i^C, \mathbf{y}^C) = \left[\mathbf{y}^C \neq \mathbf{y}_i^C\right] \tag{4-11}$$

  The square brackets act as a binary function which equals 1 when the condition within it satisfies and 0 otherwise. This is the simplest loss, which penalizes all mistakes equally. Despite its simpleness, it is widely used in object detection and categorization, where other information retrieval statistics, such as precision (P), recall(R) and F score, can be computed based on it. This binary judgement seems to be too crucial when the output states have some uneven relationships, i.e., the mistakes of confusing two outputs are not equal. For example, mistaking all components of the vector $\mathbf{y}_i^C$ might be worse than just having one single component incorrectly predicted.

- **Hamming loss**:

$$l_{Ham}^C(\mathbf{y}_i^C, \mathbf{y}^C) = \sum_j \left[y_j^C \neq y_{i_j}^C\right] \tag{4-12}$$

It counts the number of mistaken components of the output vector, which is actually the hamming distance between the predicted output and the ground truth, so we call it Hamming loss. It may be called something else like the "symmetric difference loss" in [171]. The loss increases monotonically as the number of mistaken components increases. It is a good measurement when all the components of $\mathbf{y}^C$ are equally important.

- **Hierarchical loss**:

  When the output space is structured, especially a hierarchical tree structure, then it will be better to let the loss reflect the structure. There are many types of hierarchical loss in the literature, especially those on hierarchical text classification. A common way to reflect the output hierarchy is to weight each node in the hierarchy differently, i.e. the deeper the node lies, the lighter it weights. A general form of the hierarchical loss is:

$$l^C_{Hier}(\mathbf{y}^C_i, \mathbf{y}^C) = \sum_j c_j \left[y^C_j \neq y^C_{i_j} \right] \tag{4-13}$$

  where $0 \leq c_j \leq 1$ is the weight of the node j, which is also the jth component of the vector $\mathbf{y}^C$.

  There are different ways to set the node weights, such as dividing the upper level weight by the number of its children or assigning weights propositional to the volume of the subtree, as in [171]. In their work, they proposed to penalize only the first mistake along a path from the root to a node, which is arguable.

- **Confusion loss**:

  Even though hierarchical loss reflects the differences of the output components, the hierarchical structure itself may not well represent the actual relationships among its nodes. Taxonomies are designed for hierarchical classification, but they do not guarantee that the concepts associated to all the siblings of a node are equally distinguishable, or in another word the recognition difficulties may not be distributed in the same way as we design the hierarchical loss.

  Besides of hierarchical structure, there are some other structures or general relationships that cannot be characterized by trees, like those of the handwritten digits presented in [166]. Instead of trying hard to find a weird structure that is hard to explain and optimize, they proposed to use the average confusion matrix reported in the literature to derive a loss function. The mapping from confusion matrix to the loss matrix is as follows:

$$l^C_{Conf}(\mathbf{y}^C_i, \mathbf{y}^C) = e^{-\rho \cdot \mathbf{conf}(\mathbf{y}^C_i, \mathbf{y}^C)} \tag{4-14}$$

where $\mathbf{conf}(\mathbf{y}_i^C, \mathbf{y}^C)$ is the confusion probability of the pair $(\mathbf{y}_i^C, \mathbf{y}^C)$ in the confusion matrix, and ρ is a parameter controlling the variation of the loss, which is derived from a simpler ratio parameter

$$\eta = \frac{\min_{i,j,i\neq j} l_{Conf}^C(\mathbf{y}_i^C, \mathbf{y}_j^C)}{\max_{i,j,i\neq j} l_{Conf}^C(\mathbf{y}_i^C, \mathbf{y}_j^C)}. \tag{4-15}$$

Since this loss function is based on the confusion matrix, we name it "confusion loss" here.

The confusion loss directly models the recognition difficulty of distinguishing two outputs, so it is more reasonable and general than the other losses once the confusion matrix is set properly. Under the circumstance, confusion matrix is the steerer, controlling the model learning and performance evaluation results. In other words, confusion matrix is the very result we are trying to make the recognition algorithm learn to perform. Therefore, it is critical to get it well-designed. For semantical concepts, since they are generated by our human beings after hundreds of thousands of years' living experiences, it is best for ourselves to generate the confusion matrix for object recognition, but not the way proposed in [166], where it is an averaging over the results of other recognition algorithms. By setting the human recognition results as the goal, the computer vision algorithms will try to mimic the recognition performance of human beings.

Though it's straightforward that the confusion matrix should represent the ability of human beings on distinguishing two semantic concepts, it is not an easy task to concretely measure that ability. For example, it might be easy for us to tell cats from dogs, but not as easy for distinguishing wolves and dogs. By building up taxonomies, we are trying to formulate the relationships of concepts into hierarchies, but they are just rough classification rules, not exact measurements of human recognition abilities. As far as we are aware, existing annotations of object recognition datasets haven't gone deep into this problem, leaving it a challenge for further research.

4.5.2 Localization Loss

When the output contains localization information of objects and/or object parts, there should be a metric for evaluating the localization performance and guide the learning. We introduce here two localization losses from the others and a new one proposed by ourselves as follows.

- **Hard loss**:

  A commonly used performance evaluation metric for object detection (where localization is involved) is that an object is counted as correctly detected if and only if

the area overlap ratio of predicted object bounding box and its ground truth exceeds 50%, see PASCAL VOC challenges introduced in [71]. This is an empirical threshold for differentiating object and non-object samples, and it is sharp. However, as mentioned in [7], in standard sliding window based object detection algorithms, the training set usually only contains examples that either exactly cover the labeled object or completely uncover it. In the form of loss function, it is just as follows:

$$l^L_{Hard}(\mathbf{y}^L_i, \mathbf{y}^L, \mathbf{y}^C_i, \mathbf{y}^C) = \begin{cases} 0, & if\ \frac{Area(\mathbf{y}^L_i \cap \mathbf{y}^L)}{Area(\mathbf{y}^L_i \cup \mathbf{y}^L)} = 1; \\ 1, & if\ \frac{Area(\mathbf{y}^L_i \cap \mathbf{y}^L)}{Area(\mathbf{y}^L_i \cup \mathbf{y}^L)} = 0. \end{cases}$$

where $\mathbf{y}^L_i \cap \mathbf{y}$ denotes the intersection of the two bounding boxes represented by $\mathbf{y}^L_i$ and $\mathbf{y}^L$, while $\mathbf{y}^L_i \cup \mathbf{y}$ is the union. Since there are no partial overlapping training samples, the loss is unknown. This is undesirable because the evaluation metric does not coincide with the loss function used for training, and when it comes up a partial overlapping window in the test image, the output of the algorithm is unpredictable. If the area overlap ratio is below 50% and the object recognition algorithm treats it as a positive sample, then it will be a false positive.

- **Soft loss**:

  To overcome this shortage and make better use of the training data, a soft loss function has been proposed in [7], which scales smoothly with the degree of the overlap ratio and takes into account all the windows with partial objects in them for training. The loss is defined as:

$$l^L_{Soft}(\mathbf{y}^L_i, \mathbf{y}^L, \mathbf{y}^C_i, \mathbf{y}^C) = \begin{cases} 1 - \frac{Area(\mathbf{y}^L_i \cap \mathbf{y}^L)}{Area(\mathbf{y}^L_i \cup \mathbf{y}^L)}, & if\ y^C_{i_1} = y^C_1 = 1; \\ 1 - (y^C_{i_1} - 1)(y^C_1 - 1), & if\ y^C_{i_1} y^C_1 = 0. \end{cases}$$

  It works well with the object localization algorithm using structured output regression as presented in [7]. They built up a visual codebook using local features, and used the bag-of-visual-words histograms with a linear kernel for training. In their case, the maximization step in either training or testing stage can easily get optimized using the efficient branch-and-bound optimization strategy introduced in [148], where the bounding function can be divided into two monotonic additive functions respect to the bounding box. The additivity and monotonicity coincide well with the softness of the localization loss on the manner of changing smoothly with the degree of the overlap ratio. More concretely, the feature representation (especially the bag-of-words model) in their work changes gradually as the object bounding box shifts. Therefore, such a soft localization loss is a good choice for similar settings.

- **Semi-hard loss**:

  When the features are sensitive to the precision of localization or alignment, this soft loss will be improper because it may confuse the model learning: the response of the model may be hard to decrease monotonically as the loss goes up. And the soft loss function does not directly optimize the evaluation metric which is thresholded. In this case, we propose to use a modified version of the common localization loss as follows:

$$l^L_{Semi-hard}(\mathbf{y}^L_i, \mathbf{y}^L, \mathbf{y}^C_i, \mathbf{y}^C) = \begin{cases} 0, & if\ \frac{Area(\mathbf{y}^L_i \cap \mathbf{y}^L)}{Area(\mathbf{y}^L_i \cup \mathbf{y}^L)} = 1; \\ 1, & if\ \frac{Area(\mathbf{y}^L_i \cap \mathbf{y}^L)}{Area(\mathbf{y}^L_i \cup \mathbf{y}^L)} \leq 0.5. \end{cases} \tag{4-16}$$

  It treats the low overlap ratio bounding boxes ($\leq 50\%$) as non-object samples as the evaluation metric does, while leaving out the high ratio partially overlapped ones ($> 50\%$). Since after non-maximum suppression (NMS) only one bounding box is desired for the final localization result, using only the exactly matched one as positive object sample is enough. The loss function has the same flavor with the evaluation metric while at the same time ensures the precision of localization as it prefers exact localization instead of treating it and those partial detections equally.

Note that for DLR which may demand simultaneous localization and categorization, the loss function should be a mixture of the semantic loss and the localization loss. Besides fitness, the feasibility and efficiency of optimization is also a very important issue when choosing the proper loss function and evaluation method.

4.6 Feature Representation

In former category-level object recognition tasks like classification and detection, the goal is to make the recognition system discriminative between object classes while at the same time invariant to instance changes within object classes. Therefore, features are usually designed to be as invariant as possible to intra-class variations such as changes of scale, view, pose, illumination, deformation and so on. Methods of eliminating visual information referring to these properties and facts are mostly adopted. For example, to train an object detector of a specific category, a common way is aligning training object examples by rescaling, rotating or even stretching the original ones, as in [77]. This is reasonable and effective when we want to represent various object instances using a single model, though the stretching operation distorts the data.

However, even for detection only, trying to unify all the object instances is sometimes very hard due to dramatic appearance changes like those caused by view changes. [77] propose to use mixture models which are in principle much like mixture of gaussians for modeling distributions. Though features representing the split data for learning mixture models are more informative and stable than those defined on all the data together,

the splitting strategy is usually chosen heuristically, and the split subsets are usually not semantically meaningful subclasses.

In the case of deep and layered recognition, the goal itself is recognizing different object instances as examples of subclasses with different semantic meanings. Therefore, the features for a specific object category have to represent the differences among these subclasses for differentiating them. Instead of splitting the data of the same category heuristically, meaningful labels can be used to group the examples into semantic subsets. The difficulty is how to make the features discriminative among object classes and subclasses while at the same time as compact as possible. We need to design feature descriptors for capturing the subtleties of different object instances while keeping certain robustness of the same object category to ensure interclass determinativeness. We discuss two key issues towards solving this problem: one is the so called output-sensitive features, and the other is the trade-off between discrimination and sharing.

4.6.1 Output-sensitive Features

Features for representing visual objects are designed to be both informative and robust for recognition. Dealing with the deep and layered recognition, we cannot eliminate the intra-class variations as we do for category-level recognition. Instead, we have to keep and enhance those intra-class variations helpful for differentiating subclasses. For example, if object instances of different scales correspond to different subclasses, we should let the features represent the information of scale changes, i.e. making the features scale-sensitive, but not scale-invariant. Similar demands may be needed for other aspects such as view, pose, illumination, and so on. Generally, we call these kinds of features "output-sensitive features", which means that they are sensitive to various outputs. In the case of traditional category-level recognition, features only need to be sensitive to category labels, where we do not specifically call them output-sensitive. While in the case of DLR, between-category discrimination is a must, and within-category discrimination really shows the power of adaptation of the features to output variations (usually multimodal). This is different to traditional feature representation, therefore we emphasize its capability of output sensitivity.

There can be numerous ways of designing output-sensitive features and the specification of features depends on the instantiation of deep and layered recognition tasks, namely, the data itself and its corresponding output space. Figure 4-3 presents one group of features that can be sensitive to scale changes. In this case, features are histograms with fixed number of cells while their sizes are adaptive to the actual data. Data is kept as what it is, and histogram grads are rescaled or even stretched (changing the aspect ratio) to fit it. It may result in decimal sizes of cells (subpixel level description), so interpolation is needed to do the binning. Usually, bilinear interpolation is a cheep and effective way. Since objects at different scales may be perceptually different (larger

scales tend to show more details), this type of scale-sensitive features implicitly represent the scale information. Besides that, one can also explicitly include scale as a feature, which may be useful. Global illumination and image quality measurements can be used as output-sensitive features as well.

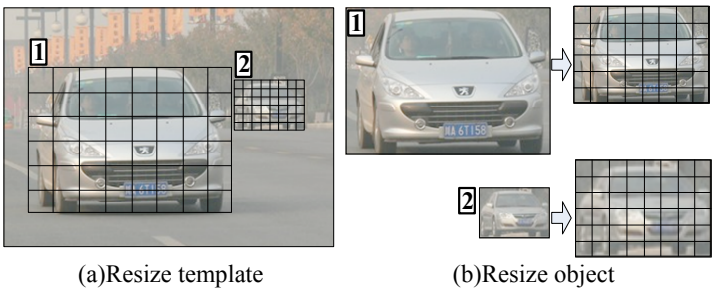

Figure 4-3 Scale-sensitive subpixel features. Instead of resizing the object to fit a fixed feature template like (b), we resize the template to fit the data as shown in (a). The subpixel-level feature descriptor represents the subtle differences of objects with different scales.

4.6.2 Trade-off Between Discrimination and Sharing

Though features for DLR need to be output-sensitive, we should not design features for each output state (object classes and subclasses) separately, disregarding possible relationships among output states and the compactness of features. This is rather critical when there are exponentially many number of output states induced by the combination of output components. Sharing of features among object classes and especially subclasses should be explored. Since both discrimination and sharing need to be considered when we design features for DLR, there should be a trade-off between them. A simple way of doing so is to include both effective category-level features and carefully designed subcategory-level features based on observations or prior knowledge, while at the same time trying to make the dimension of features as small as possible. Dimension reduction methods and feature selection methods can be used if they are efficient and effective. Though it is possible to keep the discrimination and sharing demands in mind when designing and choosing features by hand, a principle and easy to operate strategy is usually hard to find. It seems better to just make the features as informative and compact as possible before learning (i.e. a rough trade-off between discrimination and sharing), and leave the fine tuning of the trade-off to learning algorithms.

4.7 An Instantiation on Car and Pedestrian Recognition

As discussed in section 4.3, deep and layered recognition is a general problem and it can have different instantiations in different scenarios. Here we look into the case of recognizing cars and pedestrians in static images, and use the IAIR-CarPed dataset to

instantiate the model and do the experiments. Details of the implementation is presented in the following.

4.7.1 Output Structure and Loss Function

We would like to predict all the outputs provided by the IAIR-CarPed dataset, namely, the semantic hierarchy which contains a three-level semantic categorization of the object and a simple presence categorization for the key part as shown in Figure 3-2, along with the localization of both of them.

Mathematically, for each category (car or pedestrian), the output vector is $\mathbf{y} = (\mathbf{y}_O^C; \mathbf{y}_O^L; y_{KP}^C; \mathbf{y}_{KP}^L)$, where y_{KP}^C is a binary scalar indicating only the clearness of the key part.

Compact representation of categorization Since the output space of the object categorization is limited and we only care about the key part's clearness when the object is categorized into certain states, we can compress the semantic output space of the object and its key part to as small as the number of nodes of each semantic tree in Figure 3-2. By doing so, $\mathbf{y}_O^C$ and y_{KP}^C can be compressed into one single multi-valued scalar y^C, which indicates the categorization state of the whole object and its key part. Specifically, $y^C = \{0, 1, 2, \ldots, 9\}$ for car, and $y^C = \{0, 1, 2, \ldots, 8\}$ for pedestrian, in which $y^C = 0$ means the object is absent and the other values are the indices of the semantic concepts on the tree in a depth-first order. Then the whole output vector becomes $\mathbf{y} = (y^C; \mathbf{y}_O^L; \mathbf{y}_{KP}^L)$.

Categorization loss As the confusion matrix of human recognition can be derived from the votes of the 20 subjects who have labeled the IAIR-CarPed dataset, we would like to use the confusion loss l_{Conf}^C (defined in equation 4-14) as our categorization loss, and we believe that it is a good loss function and performance evaluation criterion for DLR. The underlying principle is that the more people confuse on two concepts, the smaller the loss of mixing them should be. Details about it have been stated in 4.5.1 and experiments thereafter will prove its reasonableness. In our experiments, the parameter η for the confusion losses is set to be 0.2.

Intrinsic structural semantics Though the categorization notation we have chosen doesn't represent the tree structure of the semantic hierarchy, the loss function represents the intrinsic relationships among these semantic concepts, and it is proved by human beings to be more reasonable than a heuristically defined tree structure. Therefore, the categorization is not a traditional multiclass classification, but a structured prediction whose structural information is implicitly included in the loss function. The differences between this specific DLR and traditional multiclass classification will be discussed in the experiments section.

Localization loss Generally, it's better to optimize the locations of objects and their parts together with their semantic categorizations, which need the localization loss to

be involved in the inequality constraints. In order to do this, an iterative optimization of the weights **w** and the locations should be adopted, and the loss function need to be carefully designed for easy optimization, like the soft localization loss l^L_{Soft} proposed by [7]. However, the soft localization loss together with the branch-and-bound search algorithm has the demand that the features need to be additive, so it will limit the types of features for use.

In this paper, we choose to use the sliding window strategy, which is a simple approximation for localization and it uncouples the localization and the categorization in a brute-force way. By doing so, the localization loss is usually a simple bi-valued function like the hard loss l^L_{Hard} and the semi-hard loss $l^L_{Semi-hard}$ defined in 4.5.2. And it can be combined with the semantic categorization loss to form a joint loss by treating the unsuccessfully localized hypophyses as belonging to the non-object category (i.e. $y^C = 0$), and the localization loss acts just like the categorization loss. Actually it is commonly used in object detection methods. However, we found that most of them use l^L_{Hard} in the training while using $l^L_{Semi-hard}$ for evaluation without realizing it. A typical phenomenon is that negative examples (in binary detection) are usually sampled in separate negative images which can never overlap a little bit with the positive training examples. In this paper, we would like to make a change, to use $l^L_{Semi-hard}$ for both learning and evaluation.

About the localization of the key part, we used a simple filtering classifier in a sliding window manner to find the best scoring candidate within a feasible search region relative to the whole object. This is a trade-off between the performance and the efficiency, as further discussed in the following subsection. By doing so, we do not need to put the key part localization loss into the global optimization constraints. Therefore, the final loss is the joint loss of the confusion loss for categorization and the semi-hard loss for object localization, which coincides well with the output semantics y^C. The loss function actually used for training is illustrated in Figure 4-7.

4.7.2 Feature Representation for the Specific Instantiation

The general discussion on feature representation for DLR has been given in subsection 4.6, and we present here a concrete example on selecting proper features for the case of car and pedestrian recognition.

The features we have chosen belong to four different types described in the following.

a. Shape descriptor (two-layer subpixel HOG) HOG feature and its variations have been widely used after its first introduction by [9], and it has been proved to be very powerful for representing many objects, for example pedestrians and cars. The power comes from its proper balance on selective dense gradient representation and invariant histogram binning with local normalization. [77] found that using a two-layer

HOG for representing the whole object and its deformable parts respectively can result in a much better detection performance on the challenging PASCAL dataset. Besides the benefits of the deformation and alignment of the part templates, the two-layer HOG (coarse scale and fine scale) should have provided useful complementary shape information for recognition. It inspired us to use similar representation for the adaptive data-dependent recognition of cars and pedestrians, because different categorization results seem to ask for different precisions of the shape representation (direction ambiguous subcategories prefer coarse-level HOG while direction specific subcategories tend to rely more on fine-level HOG). We believe that HOG is a good descriptor for cars and pedestrians because instances of them may vary in different appearance (color, texture, etc.) but the global shape information seems to be the most informative and robust.

We used the same strategies for efficient computation of the two-layer HOG features as presented in [77]. Specifically, there are three of them: 1) cell-based computation, i.e., each cell has four copies normalized within 4 different neighborhoods for convenient weight template filtering during testing; 2) analytic dimensionality reduction for compactness and computational efficiency; and 3) a feature pyramid for maximizing the reusability of features during multi-scale sliding window search. Beside of these, we pre-computed a gradient binning lookup table for fast binning of the pixels since there are only 511×511 different combinations for any pair of two directional gradients (dx and dy) and the binning results of each of them can be computed in advance. Since this table is independent of the size and the content of the image, once it is computed, it can be reused thereafter with only the complexity of indexing, but not that of computing the anti-trigonometric functions.

As demonstrated in Figure 4-3, we use the scale-sensitive subpixel binning method to compute the two-layer HOG features, therefore there is no rescaling or stretching of the data, and the feature can implicitly represent the scale information which is very important for the DLR strategy. Similar to the original HOG presented in [9], we add a one-cell margin to the coarse-scale HOG to involve the contextual information around the object, while letting the fine-scale HOG to just cover the object itself with the number of cells doubled. Concretely, the two-layer HOG for car has the amount of 14×6 coarse cells and 24×8 fine cells respectively, while that for pedestrian is 8×10 coarse cells and 12×16 fine cells. An example of the actual HOG features we got for a pedestrian is illustrated in Figure 4-4.

b. Appearance descriptor (local color histograms) Though the global appearances of cars and pedestrians are unstable, there are some local regions whose appearances are statistically stable within certain subcategories while at the same time sensitive to the semantic output variations. Specifically, the color of the car lights is usually informative for front or rear views (front: red, and rear: white), and the spatial distribution of the color of the skin (face and neck) and the hair is helpful for discrimi-

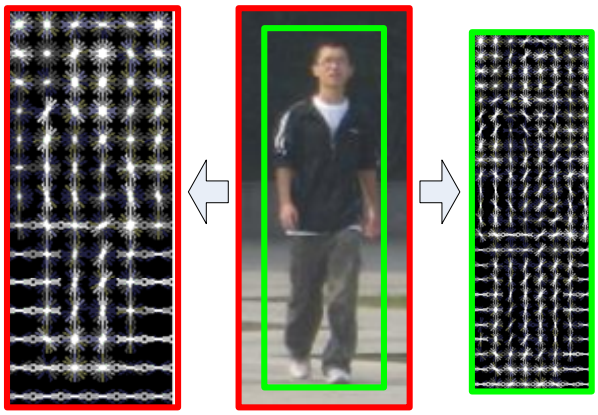

Figure 4-4 Two-layer HOG features. The coarse scale feature captures big structures of the objects and takes the close context into account, while the fine scale one focuses more on the details of object parts and shape subtleties of the objects themselves.

nating the directions of the pedestrians.

As suggested by many other people, we use the HSV color space which is considered to be intuitive and informative for general purpose and quantize it into a few bins for building color histograms. For cars, three regions, two for the left and right lights in front or rear view and the third one for direction unspecific (e.g. oblique views) cases are chosen for computing the color histograms, as shown in Figure 4-5. We quantized the whole HSV space into 18 bins, in which "Hue" has 3 bins ([0, 30°] & [330°,360°), (30°, 75°] and (75°, 330°)), "Saturation" has 2 ([0, 0.5] and (0.5, 1]), and "Value" has 3 ([0, 0.2), [0.2, 0.4), [0.4,1]). The 3 different hues are designed for representing different car lights, i.e., red, yellow and the others. We noticed that usually the tail lights are red (lighted or not) and the front lights are yellow/white when lighted and transparent if not. For pedestrians, according to [172], the "Saturation" of the skin color spreads too wide and therefore it is not discriminative, while in "Hue" the skin color is well clustered, mainly ranges in [0, 0.128] or equivalently [0, 45°]. Therefore, we only used 6 bins to quantize the space for skin color: 2 for "Hue" ([0, 0.128] and (0.128,1]), and 3 for "Value" ([0, 0.2), [0.2, 0.4), [0.4,1]). The three local regions for computing the color histograms are also shown in Figure 4-5. Note that the binning for "Value" is done heuristically without further tuning, and it can be improved in the future.

Since the dimensions of the local color histograms are not high (18×3 for car and 6×3 for pedestrian), they are efficient complementary features of the high-dimensional two-layer HOG features. As usual, integral images can be used to make the computation of the color histograms efficient.

c. Global content-independent features (object scale) As mentioned in 4.6.1, in the scenario of deep and layered recognition, some content-independent features may also be useful as long as they are output-sensitive. The scale of the object is one

of them. From the average images shown in Figure 3-2 and the distributions of the object scale presented in Figure 3-9, we can see that the scale is able to distinguish some semantic subcategories from the others. Therefore, this one dimensional global feature was involved in our final features. There might be some other features that are also helpful, but we haven't looked into them.

d. Key part features (presence score) Unlike the cropped body parts in [77], our semantic key parts (face and license plate) are visually very different patterns from the whole objects they are in, so we couldn't use the same types of features to describe the objects and their parts. Though HOG features are considered to be effective for representing the shape information of many objects, they are not suitable for describing small objects or object parts with tiny details like faces and license plates. Instead, some other features have been proved to be effective for handling these specific tasks, such as the haar-like features for face detection in [47] and the statistics of gradients for license plate in [173]. Though faces and license plates are relatively rigid and specific patterns, to build a robust detector for either of them is not an easy task, and usually a lot of weak features are needed to boost the performance as in [47]. Obviously, putting a large amount of raw features together with the features of the objects for global model parameter (i.e. $\mathbf{w}$) learning is undesirable in DLR as it has a relatively large output space and therefore needs a large number of training examples. To make things easier, we chose to use these raw features to train a separate key part detector, and use the classification score as the feature of the key part. By doing so, the dimension of the key part features is mostly reduced while the power of the raw features for representing the key part is kept. The disadvantage is that this separate key part detector proposes only a local minimum of the global optimization problem. Even though, we believe that a reasonable result can be got, and it can serve as a benchmark for further research.

To build a reliable license plate detector, we used four types of features as below:

1. **Statistical gradient features**.

   Inspired by the two global statistical features (gradient density and density variance) introduced in [173], we used the following three statistical gradient features which are more informative and robust than those two:

   - *Gradient density contrast.* As demonstrated in Figure 4-6, instead of just computing the average gradient (i.e. gradient density) in the license plate region, we compute the contrast of the gradient density between the license plate region and its surrounding background (i.e. the body of the car). When this feature is large, it not only means that the gradient density in the license plate region is large, but also indicates that the gradient density of the surrounding background is relatively very small. The shape of the license plate is implicitly taken into account. This is thought to be more robust than the gradient density

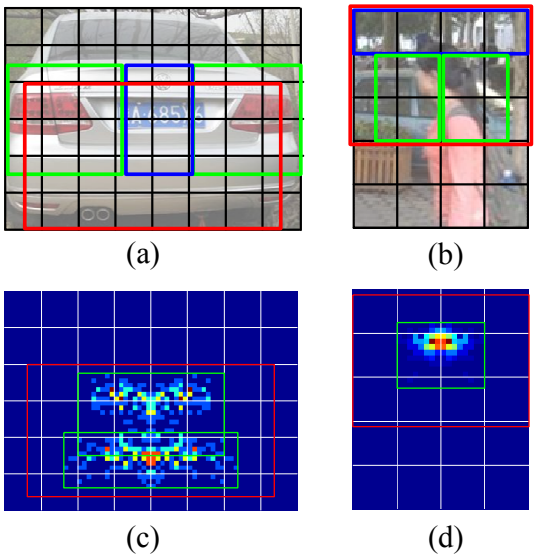

Figure 4-5 Spatial occupations of the features. The green and blue bounding boxes in (a) and (b) show the regions for computing color histograms. Subgraph (c) and (d) illustrate the statistic distributions of the license plate and the face respectively. The colored spots show the spatial histogram of the centers of this two key parts, while the green rectangles bound the distributions of the centers. The red rectangles in (c) and (d) bounds the annotated key part bounding boxes, while the red ones in (a) and (b) are the search regions actually been used in our implementation.

Figure 4-6 Features for training the license plate detector: a) regions for extracting statistical gradient features and b) an example of the extracted SIFT features.

proposed in [173] which may have confusions on other gradient-rich areas like car lights and the radiator cowling. To make the feature robust to misalignments caused by sliding window search, we computed the gradient densities of two candidate regions of the license plate (the upper one in green and the lower one in red) and chose the one with larger density as the representative of the license plate. This simple trick can make sure the gradient density is always computed within the license plate when the step of the sliding window is 1/3 of the height of the license plate.

- *Spatial gradient density variance.* It is the same as the density variance feature proposed in [173]. Instead of dividing the license plate region into 12 cells, we divided representative license plate region (the winner on gradient density) into 2×7 cells and compute the gradient density variance on it.

- *Orientation entropy of gradients*. This new feature is designed to distinguish the license plate with varying curved characters from other gradient-rich areas like the radiator cowling whose gradients are more directional. Like the HOG descriptor, we divided the orientations into 18 bins (20°each), but we only computed the orientational histogram of gradients without spatial binning. The histogram was normalized using $L1$ norm, Then the orientation entropy is compute based on the normalized histogram $\mathbf{h}^g = (h_1^g,$ $\ldots, h_{n_{ori}}^g)$:

$$I_{LP} = -\frac{1}{\log n_{ori}} \sum_{i=1}^{n_{ori}} h_i^g \log h_i^g.$$

  where $n_{ori} = 18$ is the number of orientation bins.

2. **Color feature**.

   Visually it can be easily found that license plates in one country usually have unified colors, and this characteristic can be used for detection. Statistical analysis shows that the license plates in the IAIR-CarPed dataset have well-clustered "Hue" values (mainly range in (0.5, 0.7)), but other channels are not well-clustered due to diverse illuminations. Therefore, we just computed a two-bin histogram on the "Hue" value of the license plate region, and normalize it using $L1$ norm. Due to the linear correlation of these two bins, using only a single value is enough, we use the contrast value as the final color feature, i.e., the percentage of pixels whose "Hue" values are within (0.5, 0.7) minus the percentage of those outside the range. We linearly mapped this contrast value to be within [0, 1].

3. **Content-independent feature (scale)**. Scale is an important factor that may influence the presence (or more precisely clearness in our setting) of the license plate. In most cases, license plates with larger scales tend to be clearer. Therefore, this simple content-independent feature is adopted.

4. **Bag of SIFT features**.

   The above three types of features are low-dimensional statistical features, which are informative but not discriminative enough to train a good license plate detector. The distinctiveness of license plates is not limited to these statistics, but more importantly the regular characters and digits printed on it. The content of a license plate happens to be a combination of very few possible characters and digits, and they are printed in the same font and size at fixed positions. This characteristic makes it possible to compute SIFT features at fixed positions to represent the

characters and digits, and then use a bag-of-words (BoW) model to make the representation spatial invariant, i.e., to coincide with the randomness of the characters and digits.

Figure 4-6 shows how these SIFT features are arranged. In the IAIR-CarPed dataset, each license plate has exactly 7 characters and digits, and each of them has a aspect ratio close to 2 (height vs. width). Therefore, we computed 14 SIFT features for each license plate as demonstrated in the figure, with each of them capturing one half of a character or digit. Then these SIFT features were clustered into a codebook with the size of 200 using k-means. The size of the codebook was chosen heuristically, considering possible misalignments and the appearances of negative examples. Once the codebook is built, the 14 SIFT features of each license plate can be represented as a 200-dimensional histogram on this codebook, which is our bag of SIFT features.

Using all of these features (totally 205-dimensional), we can train a good license plate detector using linear SVM with the standard bootstrapping technique. Note that we only trained the detector on object windows whose semantic categorization labels are "front with/without clear license plate" or "rear with/without clear license plate", because only these windows are possible for the presence of the license plate as desired. A well-trained detector using this kind of data may generate false positives on cars with other directions or in non-car regions, there is no need to worry about it because our DLR classifier makes decisions based on both the object features and the key part presence feature. A false positive license plate detection is almost impossible to change the direction of a car or generate a false positive car detection with a very deep semantic label: front or rear with a clear license plate.

About the face detector, we used the off-the-shelf detector from OpenCV 1.0 which is a nice implementation of the algorithm proposed by [47]. Unlike the license plate detector which was trained using our own training data, the face detector we used is the already trained one as it is in OpenCV 1.0. We did so not just for convenience, but also for the robustness of the detector. There are very few clear frontal faces (around 100) in our training dataset , so they are not sufficient to train a good detector. By choosing the off-the-shelf detector, we just treated it as a feature extraction tool that can extract a good face presence feature for us.

Once the key part detector was trained or obtained, we used it as a filtering classifier to convolute with both the training and test images. The filter response map is a score map for each possible locations of the key part. Together with the pre-computed object feature maps, we can construct all the features for each object candidate window. Since the key part is much smaller than the object, and its relative position to the object may vary a bit, the step of the sliding window for the key part will be much smaller than that for the object itself. For example, the license plate's sliding step is 5 times

smaller than that of the car. Therefore, for each object location, there are many location candidates for the key part which is consistent with the object. To find the exact key part locations relative to the object location, we did a statistical analysis on the annotations of the key part locations with respect to the object bounding boxes. The lower part of Figure 4-5 shows the distributions of the key parts' localization centers along with the maximum extensions of their bounding boxes. Using these extension boundaries, we can define search regions for the key parts as shown by the red rectangles in the upper part of Figure 4-5. Within the search region of the object key part, we propose its best location by a non-maximum suppression (NMS) strategy, i.e., the location with the largest presence score wins.

4.7.3 SOnline: An Efficient Structured Online Learning Algorithm

As mentioned in 4.4.2, in general a discriminative structured output prediction problem can be formulated as:

$$\begin{aligned} &\min_{\mathbf{w}} \left\{ \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i \right\} \\ &s.t. \quad \begin{cases} \langle \mathbf{w}, \Delta\Phi\left(\mathbf{x}_i, \mathbf{y}\right)\rangle \geq l\left(\mathbf{y}_i, \mathbf{y}\right) - \xi_i \quad , \quad \forall i,\ \mathbf{y} \neq \mathbf{y}_i \\ \xi_i \geq 0 \ , \quad \forall i \end{cases} \end{aligned} \tag{4-17}$$

where $\Delta\Phi\left(\mathbf{x}_i, \mathbf{y}\right) = \Phi\left(\mathbf{x}_i, \mathbf{y}_i\right) - \Phi\left(\mathbf{x}_i, \mathbf{y}\right)$.

Though the above formulation looks simple, in practise there might be some computational difficulties which need to be seriously treated: a) the high dimensionality of $\mathbf{x}$, and b) the existence of large amounts of training patterns. Absolutely batch learning algorithms cannot be used to solve it directly. A hint from the object detection problem is to use the data mining (or namely bootstrapping) strategy to iteratively sample an acceptable number of representative non-object examples for training when the number of non-object samples is very large. Usually, the mined non-object examples are added to the training set, resulting in the increasing of its size. [77] proposed a new data-mining algorithm with the removal of non-supporting patterns which can control the size of the training set while ensuring its quality. Though the algorithm they proposed is proved to be optimal, its convergence may be rather slow when the data is hard for the recognition task.

As far as we are aware, after each iteration of the data-mining in object detection, most algorithms train a new model from scratch on the new training set. This is undesirable because the formerly trained model which might be close to the optimal is totally abandoned during the retraining. When each round of the training is time-consuming (as in many structured output prediction problems) and the number of data-mining iterations is large, it will be a disaster. Therefore, we present here an efficient online learning algorithm for solving the structured output prediction problem when the training set

is large or changing over time. It not only trains the model online when the training examples are given, but also updates the trained model online when new examples are added in the training set or adopted to replace some of the existing examples.

The dual problem of formula 4-17 can be derived as:

$$\begin{aligned}&\max_{\alpha} \sum_{i,\mathbf{y}\neq\mathbf{y}_i} \alpha_i(\mathbf{y})\, l(\mathbf{y}_i,\mathbf{y}) \;- \\ &\quad \tfrac{1}{2}\sum_{i,\mathbf{y}\neq\mathbf{y}_i}\sum_{j,\bar{\mathbf{y}}\neq\mathbf{y}_j} \alpha_i(\mathbf{y})\,\alpha_j(\bar{\mathbf{y}})\,\langle \Delta\Phi(\mathbf{x}_i,\mathbf{y}), \Delta\Phi(\mathbf{x}_j,\bar{\mathbf{y}})\rangle \\ &s.t. \quad \begin{cases} \alpha_i(\mathbf{y}) \geq 0\ , & \forall i, \mathbf{y}\neq\mathbf{y}_i \\ \sum\limits_{\mathbf{y}\neq\mathbf{y}_i} \alpha_i(\mathbf{y}) \leq C\ , & \forall i \end{cases}\end{aligned}$$

By replacing α with an augmented variable γ

$$\gamma_i(\mathbf{y}) = \begin{cases} -\alpha_i(\mathbf{y}), & \mathbf{y}\neq\mathbf{y}_i \\ \sum\limits_{\mathbf{y}\neq\mathbf{y}_i} \alpha_i(\mathbf{y}), & \mathbf{y}=\mathbf{y}_i \end{cases}$$

and factorizing the joint kernel as

$$\begin{aligned}&K(\mathbf{x}_i,\mathbf{y},\mathbf{x}_j,\bar{\mathbf{y}}) = \langle \Phi(\mathbf{x}_i,\mathbf{y}), \Phi(\mathbf{x}_j,\bar{\mathbf{y}})\rangle \\ &= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle \langle \varphi(\mathbf{y}), \varphi(\bar{\mathbf{y}})\rangle = K_{\mathbf{x}}(\mathbf{x}_i,\mathbf{x}_j) K_{\mathbf{y}}(\mathbf{y},\bar{\mathbf{y}})\end{aligned}$$

with the simplest choice of the output kernel

$$K_{\mathbf{y}}(\mathbf{y},\bar{\mathbf{y}}) = \begin{cases} 1, & \mathbf{y}=\bar{\mathbf{y}} \\ 0, & \mathbf{y}\neq\bar{\mathbf{y}} \end{cases},$$

the optimization problem can be transformed into such a formula:

$$\begin{aligned}&\max_{\gamma} \left\{ -\sum_{i,\mathbf{y}} \gamma_i(\mathbf{y})\, l(\mathbf{y}_i,\mathbf{y}) \; -\tfrac{1}{2}\sum_{i,j}\sum_{\mathbf{y}} \gamma_i(\mathbf{y})\,\gamma_j(\mathbf{y})\, K_{\mathbf{x}}(\mathbf{x}_i,\mathbf{x}_j) \right\} \\ &s.t \quad \begin{cases} \gamma_i(\mathbf{y}) \leq 0, & \forall i, \mathbf{y}\neq\mathbf{y}_i \\ \gamma_i(\mathbf{y}_i) \leq C\ , & \forall i \\ \sum\limits_{\mathbf{y}} \gamma_i(\mathbf{y}) = 0\ , & \forall i \end{cases}\end{aligned} \tag{4-18}$$

After learning the optimal γ^*, the output prediction function is as simple as

$$f(\mathbf{x}) = \arg\max_{\mathbf{y}} \sum_i \gamma_i^*(\mathbf{y})\, K_{\mathbf{x}}(\mathbf{x}_i,\mathbf{x}). \tag{4-19}$$

The proposed `SOnline` algorithm for solving the dual problem (4-18) is summarized in algorithm 2 which can handle changeable training examples. When a model has been trained on the initial training set and there are new updates on the training examples, the `SOnline` algorithm can update the model using the new training set along with the indices of the updated examples. Note that *niter_train* is the number of training

Algorithm 2 SOnline:

Input:

Training examples $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ are the training patterns and $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathcal{Y}$ are their corresponding semantic labels;
Loss function $l(\mathbf{y}_i, \mathbf{y}_j)$;
Initial model parameters γ_0 (Optional);
Update example indices $\mathcal{U}_0$ (Optional).

Output:

Learned model parameters γ.

if a initial model γ_0 exists, **then**
  $\gamma = \gamma_0$; $\mathcal{U} = \mathcal{U}_0$;
else
  $\gamma = \mathbf{0}$; $\mathcal{U} = \{1, \ldots, n\}$;
end if
for $iter = 1$ to $niter\_train$ **do**
  if $iter > 1$ **then**
    $\mathcal{U} = \{1, \ldots, n\}$;
  end if
  for $i = 1$ to $|\mathcal{U}|$ **do**
    Let j be the ith member of $\mathcal{U}$;
    repeat
      `adaptive-operation-selection`(j);
      Suppose $(k, \mathbf{y}^+, \mathbf{y}^-)$ are the outputs of the selected operation, call algorithm SMO-SOnline$(k, \mathbf{y}^+, \mathbf{y}^-, \gamma)$;
    until The example $\mathbf{x}_j$ is processed.
  end for
end for

iterations, and when it is set to 1, the algorithm will be a pure online learning algorithm, which performs well on relatively easy datasets. When the data is hard for recognition, taking a few number of iterations (e.g. 3-5) is usually a promising choice.

The function `adaptive-operation-selection` refers to the adaptive operation selection strategy proposed in [24], which selects one of three different operations on choosing the most promising support pattern and its support vectors (i.e. a triple $(i, \mathbf{y}^+, \mathbf{y}^-)$) for online model learning. The selection strategy maintains a running estimate of the average ratio of the dual increase over the duration of each operation, and randomly selects an operation with a probability proportional to its estimate of the ratio. It has the same spirit as stochastic gradient decent, while the embedded perceptron-based algorithm uses the structural inference to select the two most promising parameters for optimization. A sequential minimal optimization (SMO) elementary step is adopted to

do the model updating as presented in algorithm 3.

Algorithm 3 SMO-SOnline:

Input:

The index of the pattern to be processed i;
The class label whose model parameter needs to be increased $\mathbf{y}^+$;
The class label whose model parameter needs to be decreased $\mathbf{y}^-$;
Existing model parameters γ.

Output:

Updated model parameters γ.

1: Retrieve or compute gradients:
$g_i(\mathbf{y}^+) = -l(\mathbf{y}_i, \mathbf{y}^+) - \sum_j \gamma_j(\mathbf{y}^+) K_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_j)$
$g_i(\mathbf{y}^-) = -l(\mathbf{y}_i, \mathbf{y}^-) - \sum_j \gamma_j(\mathbf{y}^-) K_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_j)$

2: Compute model updating step:
$\lambda^* = \frac{g_i(\mathbf{y}^+) - g_i(\mathbf{y}^-)}{2K_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_i)}$
Considering the constraints, set the updating step to be:
$\lambda = \max(0, \min(\lambda^*, C\delta(\mathbf{y}^+, \mathbf{y}_i) - \gamma_i(\mathbf{y}^+)))$.

3: Update the model:
$\gamma_i(\mathbf{y}^+) = \gamma_i(\mathbf{y}^+) + \lambda$
$\gamma_i(\mathbf{y}^-) = \gamma_i(\mathbf{y}^-) - \lambda$;

4: Update the support vector set $\mathbf{S}$ by removing zero values in γ;

5: Update gradients for fast retrieval in the future:
$g_j(\mathbf{y}^+) = g_j(\mathbf{y}^+) - \lambda K_{\mathbf{x}}(\mathbf{x}_j, \mathbf{x}_i), \quad \forall j, \quad (\mathbf{x}_j, \mathbf{y}^+) \in \mathbf{S}$
$g_j(\mathbf{y}^-) = g_j(\mathbf{y}^-) + \lambda K_{\mathbf{x}}(\mathbf{x}_j, \mathbf{x}_i), \quad \forall j, \quad (\mathbf{x}_j, \mathbf{y}^-) \in \mathbf{S}$.

4.7.4 Learning and Inference Using SOnline

As stated in 4.7.1, the localization of objects is transferred into object/non-object labeling of the subwindows in the sliding window based recognition, and the semi-hard localization loss can be represented by the semantic categorization loss between objects and non-objects. The key part localization is done by a separate non-maximum suppression in its relative search region to the object hypothesis. In this setting, the DLR problem can be simplified into a standard structured prediction problem, in which the input pattern $\mathbf{x} \in \mathcal{X}$ is the feature representation of an arbitrary subwindow, and $y^C \in \mathcal{Y}$ with $\mathcal{Y} = \{0, 1, \ldots, m\}$ is the only desired output of it. The loss function for this problem is defined on different states of y^C. Since the output to be predicted is only about categorization, we use y instead of y^C for simplicity.

Mathematically, the specific problem becomes as simple as:

$$\begin{aligned} &\min_{\mathbf{w}} \left\{ \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \xi_i \right\} \\ &s.t. \quad \begin{cases} \langle \mathbf{w}, \Delta\Phi(\mathbf{x}_i, y) \rangle \geq l(y_i, y) - \xi_i \quad , \quad \forall i,\ y \neq y_i \\ \xi_i \geq 0 \ , \quad \forall i \end{cases} \end{aligned} \tag{4-20}$$

where $\Delta\Phi(\mathbf{x}_i, y) = \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, y)$ and the joint feature vector has two parts: $\Phi(\mathbf{x}_i, y) = (\Phi_O(\mathbf{x}_i, y); \Phi_{KP}(\mathbf{x}_i, y))$. This is exactly the problem which the proposed SOnline algorithm can solve efficiently. As the output is a scalar which can take only a few different values, no special inference technique is needed and a brute-force search will work.

We present the implementation details for the instantiated DLR problem as follows.

1. **Preprocessing**

   Data splitting All the images were randomly permutated and split into two subsets: training set $\mathbf{I}_{tr} = \{I_1^{tr}, \ldots, I_{n_{tr}}^{tr}\}$ and test set $\mathbf{I}_{ts} = \{I_1^{ts}, \ldots, I_{n_{ts}}^{ts}\}$, where n_{tr} and n_{ts} are the numbers of images in these two subsets. Usually a fixed training-test ratio of $\frac{n_{tr}}{n_{ts}}$ is kept during the splitting (e.g. 1.0). If different trials need to be done for experiments or averaging, this step can be run several times to generate different splits of the data.

   Key part detector training Once the training set is fixed, a license plate detector can be trained from the object windows whose categorization labels are "front with/without clear license plate" or "rear with/without clear license plate" as mentioned before. When the training set changes, the key part detector needs to be retrained to keep consistent with the whole model training.

   Feature precomputation All the features can be precomputed except that for the license plate which depends on the training set indices. When the training set is fixed and the license plate detector is trained, the license plate presence feature can be computed and put into the whole feature pool. For the sliding window based applications when the feature depends on the subwindow of the object, this precomputation can usually save much time for both training and testing, especially when the kernel computation can be reused for extracting features at different locations and scales. In our implementation, the two-layer HOG features are decomposed into cells which can be computed independently, and as there are plenty of cells in each dimension of the feature descriptor (see 4.7.2), the stride of the sliding window can be set to be just one-cell height/width, therefore these precomputed cells (which naturally forms a pyramid) can be assembled into features of subwindows at different locations and scales. These raw cell features are stored instead of the HOG features, and the assembling is done on-the-fly during the usage of the subwindow features (in training and testing). We represent the precomputed cell features for

the whole training set as $\varphi_F(\mathbf{I}_{tr}) = \{\varphi_F(I_1^{tr}), \ldots, \varphi_F(I_{n_{tr}}^{tr})\}$, in which the function φ_F denotes the feature extraction process.

Localization map Using the localization loss function of the object (i.e. the semi-hard loss $l_{Semi-hard}^{L}$), and the annotated object bounding boxes, we can generate a object/non-object indication map for all the subwindows in an image extracted by the sliding window. We call this map "localization map" with the notation map_L. The localization maps of the training set $map_L(\mathbf{I}_{tr}) = \{map_L(I_1^{tr}), \ldots, map_L(I_{n_{tr}}^{tr})\}$ can be used for data mining non-object examples, while those of the test set are good for performance evaluation.

2. **Training**

Given the training set with annotations, the chosen loss function and the precomputed features and localization maps, the specific DLR problem of recognizing cars and pedestrians in static images can be solved by the proposed `SOnline` learning algorithm along with the data-mining strategy. The procedure of the training process is presented below:

Procedure of Training

Input:

Training image set $\mathbf{I}_{tr} = \{I_1^{tr}, \ldots, I_{n_{tr}}^{tr}\}$;
Precomputed features:
$\mathcal{X}^{obj} = \{\mathbf{x}_1^{obj}, \ldots, \mathbf{x}_{n_{obj}}^{obj}\}$ for objects,
$\varphi_F(\mathbf{I}_{tr}) = \{\varphi_F(I_1^{tr}), \ldots, \varphi_F(I_{n_{tr}}^{tr})\}$ for non-objects;
Object categorization labels $\mathcal{Y}^{obj} = \{y_1^{obj}, \ldots, y_{n_{obj}}^{obj}\}$;
Localization maps of the training set
$map_L(\mathbf{I}_{tr}) = \{map_L(I_1^{tr}), \ldots, map_L(I_{n_{tr}}^{tr})\}$;
Loss function l;
Acceptable number of non-object examples n_{non}.

Output:

Trained model parameters γ.

1: Initial sampling of non-object examples $\mathcal{X}^{non} = \{\mathbf{x}_1^{non}, \ldots, \mathbf{x}_{n_{non}}^{non}\}$,
whose categorization labels are $\mathcal{Y}^{non} = \{y_1^{non}, \ldots, y_{n_{non}}^{non}\}$,
$\forall i \in \{1, \cdots, n_{non}\}, y_i^{non} = 0$.
Set all the training examples to be $\mathcal{X} = \mathcal{X}^{obj} \cup \mathcal{X}^{non}$ with labels $\mathcal{Y} = \mathcal{Y}^{obj} \cup \mathcal{Y}^{non}$.
2: Train a initial model γ and get the non-supporting non-object examples $\mathcal{X}_{NS}^{non}$:
$[\gamma, \mathcal{X}_{NS}^{non}] = \text{SOnline}(\mathcal{X}, \mathcal{Y}, l)$.
3: Data-mining non-object examples on $\mathbf{I}_{tr}$:
4: **for** $iter = 1$ to $niter\_mining$ **do**

$\mathcal{X}_{rep}^{non} = \emptyset$;
for $i = 1$ to n_{tr} **do**
  $\mathcal{X}_{mine}^{non} = \texttt{data-mining}(\varphi_F(\mathbf{I}_{tr}), map_L(\mathbf{I}_{tr}), \gamma)$;
  $\mathcal{X}_{rep}^{non} = \mathcal{X}_{rep}^{non} \cup \mathcal{X}_{mine}^{non}$;
  if $\left|\mathcal{X}_{rep}^{non}\right| \geq |\mathcal{X}_{NS}^{non}|$ **then**
    break;
  end if
end for
if $\left|\mathcal{X}_{rep}^{non}\right| = 0$ **then**
  break;
end if
$[\mathcal{X}^{non}, \mathcal{U}] = \texttt{replace-example}(\mathcal{X}^{non}, \mathcal{X}_{NS}^{non}, \mathcal{X}_{rep}^{non})$;
$\mathcal{X} = \mathcal{X}^{obj} \cup \mathcal{X}^{non}$;
Update the model γ and get its non-supporting non-object examples $\mathcal{X}_{NS}^{non}$: $[\gamma, \mathcal{X}_{NS}^{non}] = \text{SOnline}(\mathcal{X}, \mathcal{Y}, l, \gamma, \mathcal{U})$;
if *fppi* $<$ *fppi_th* **then**
  break;
end if
end for

The function `data-mining` does the data-mining on the training set, while the function `replace-example` replaces the non-supporting examples with the newly mined non-object examples. Some of the details of this procedure are explained in the following.

Initial sampling of non-object examples Features of object instances were directly computed in the annotated bounding boxes, while those for non-object examples were precomputed in a batch manner as mentioned above. The definition of non-object examples is based on the semi-hard localization loss $l_{Semi-hard}^{L}$, i.e., when the overlap ratio between the subwindow and the bounding box of the object is lower than 50%, then the data within this subwindow is a non-object example. To make the initial set of the non-object examples as diverse as possible, they were sampled randomly from all the non-object candidates in the training images (cross image, scale and location). The number of initial non-object examples is decided by a predefined object non-object ratio $r_{tr} = n_{obj}/n_{non}$, where n_{obj} is the number of object instances for training and n_{non} is that of the non-object instances.

Data mining and model updating Since training with all the examples (object and non-object) simultaneously is infeasible, a data-mining (or bootstrapping) strategy is commonly used to address this problem. The main idea of it is to iteratively mine "hard" non-object examples from all the candidates and add them into

the training set to train a new model. The iteration ends when there is no more "hard" non-object examples can be found or the training error (e.g. false positives per image (fppi)) meets a predefined threshold, or even simpler the iteration meets the predefined maximum number of iterations *niter_mining*.

In practice, usually the "hard" means incorrectly classified, i.e., false alarms. However, as argued in [77], adding only false alarm examples into the training set is just an approximation to the problem of training on all the data, and when the "hard" means supporting patterns for maximum-margin classifiers, then this iterative mining will be sure to converge to the exact solution of the training problem using all the examples. It can be proved that it is true, not only for the traditional SVM or the latent SVM, but also for the maximum-margin based structural prediction algorithms. However, in our experiments, we found that the convergence may be too slow, especially in our proposed `SOnline` algorithm. Alternatively, using the false alarms as "hard" examples can make the mining process converge much faster. This is reasonable since incorrectly classified non-object examples influence the decision plane more than those in the margin.

When the dimension of the data is very large as in our case it is more than 8000 and the output space is also large which means there are many parameters to learn, training is usually very slow. Therefore, we adopted two more strategies to accelerate the data-mining process: a) pre-mining valuable non-object examples using a subset of features (e.g. excluding the fine-level HOG features) and b) mining from a subset of training images (e.g. the first 20%) to get a good enough model and then use it to mine new examples from the whole training set.

3. **Testing**

   The testing is just the inference problem given the model, which is identical to the data-mining operation in the training process. Equation 4-19 gives the exact function for testing and data mining. To make both of these two processes as fast as possible, we chose to use the linear kernel $K_{\mathbf{x}}(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x} \rangle$, which results in a much simpler prediction function:

$$f(\mathbf{x}) = \arg\max_{y} \langle \mathbf{w}^*(y), \mathbf{x} \rangle . \tag{4-21}$$

   where $\mathbf{w}^*(y) = \sum_i \gamma_i^*(y)\, \mathbf{x}_i$ denotes the linear weights for each output state. By doing so, predictions on subwindows in an image can be done in a batch processing way by convolution on the feature pyramids. Note that, the bias needs to be taken into account in the linear case, and it can be done by simply augmenting the feature vector with a constant "1".

4.8 Experiments and Results

Experiments were done on the two most representative subsets of the IAIR-CarPed dataset: set "S" and set "SD-D2". The first one is the simplest subset with no visual difficulties, while the second one is the largest subset with all applicable difficult samples (excluding the truncated cases). For convenience, we rename them as "**set A**" and "**set B**" respectively.

4.8.1 Superiority of DLR

To show the superiority of the proposed DLR, we would like to benchmark it against traditional *binary categorization* (i.e. object detection) and *multiclass classification*. We used the same input feature representation for all of them to exclude its influence on the final performance. Besides that, we adopted the same learning algorithm (`SOnline` with data-mining) to train their models, leaving the only difference among them to be the output structure which is the intrinsic one. For *binary categorization* (in short "BC"), all the object instances are treated as belonging to one single category without differentiation, therefore the output y takes only two values(0 and 1), and the `SOnline` algorithm is degraded into a normal binary SVM with online learning ability. For *multiclass classification*, we just use the zero-one loss $l_{0/1}$ for the output space instead of the confusion loss l_{Conf}. By doing so, the structural information of the output space represented by the loss function is lost and the problem is degraded into a multiclass classification problem though the possible output states remain unchanged. Note that it is not identical to the traditional definition of multiclass classification problem since the output classes are not within a single layer and the feature representation is already made to be output-sensitive. More precisely, the feature representation and the output space are designed for DLR, but the learning is the same as that for multiclass classification when y is a scalar. A proper view of it might be multiclass classification for solving DLR. Therefore, we just name it use the loss "$l_{0/1}$" and have DLR also named after its structured loss "l_{Conf}". The confusion loss of the training data is shown in Figure 4-7, in which the losses among different output states of the objects are derived from the semantic confusions of the human labels and those between the object and non-object categories are set to 1 which coincides with the semi-hard localization loss $l^L_{Semi-hard}$.

Since the proposed `SOnline` algorithm roots in the `LaRank` algorithm introduced in [24], we implemented it based on the publicly available code of `LaRank`. Note that there is a free parameter C in the learning when linear kernel is chosen. We tried several different values during the learning of the model, and found that $C = 1$ is a good choice for all the three recognition strategies and the training results are not very sensitive to its value.

Figure 4-8 shows some representative testing results of these three different strategies on set A. It can be seen that the *multiclass classification* simulated by DLR with loss

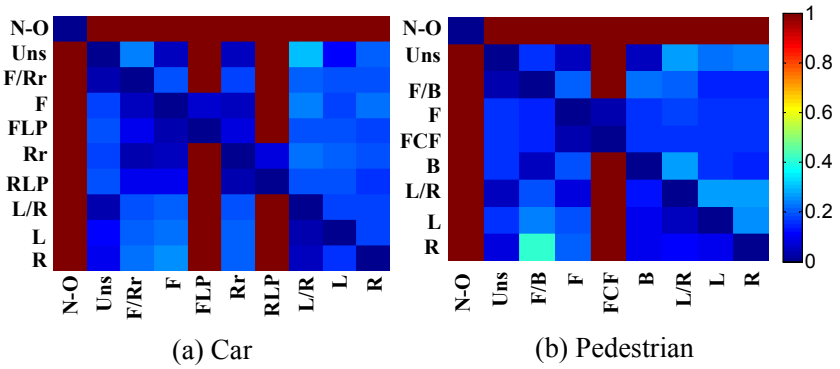

Figure 4-7 The joint loss matrices for the class of "pedestrian" and "car" respectively on the training data. The abbreviation of "N-O" denotes the non-object category.

"$l_{0/1}$" tends to have higher recall on object detection than *binary categorization* while at the same time provides deeper categorization of the detected objects, and our proposed DLR with loss "l_{Conf}" gets even better detection (category-level recognition) performance and mostly its detailed categorization of objects looks more reasonable than that of the *multiclass classification*. Though statistically it is true, there are some exceptions as presented in the last two images of Figure 4-8. "BC" sometimes gets higher recall than the two others while the robustness of them may vary in different images as shown in the second row ("BC" gets the right most car with little occlusion and "DLR with l_{Conf}" gets the other three). The depth of recognition may not coincide well with the object scale as closer objects may not necessarily have clearer appearances. In the last row, "DLR with $l_{0/1}$" predicts the right label (front with clear face) of the pedestrian while "DLR with l_{Conf}" fails, but the mistake is reasonable as its prediction is closest to the actual and their different is very small for the specific case.

Besides qualitative demonstration, we also provide quantitative measurements as follows.

Table 4-1: Statistics on the performance of category-level recognition. The first three measurements are calculated based on the final trained model, and the forth one named $F1_{max}$ denotes the maximum $F1$ value over different bias values of the non-object category.

		Car				Pedestrian			
		P	R	$F1$	$F1_{max}$	P	R	$F1$	$F1_{max}$
Set A	DLR with l_{Conf}	0.8088	0.7918	**0.8002**	**0.8086**	0.8264	0.7402	**0.7809**	**0.7809**
	DLR with $l_{0/1}$	0.7738	0.7404	0.7567	0.7791	0.8236	0.7097	0.7624	0.7624
	BC	0.6841	0.6281	0.6549	0.6597	0.7944	0.6918	0.7396	0.7397
	DTPM	0.8386	0.5328	0.6516	0.7081	0.4969	0.6724	0.5715	0.6243
Set B	DLR with l_{Conf}	0.576	0.6911	**0.6283**	0.6669	0.1659	0.8876	0.2795	**0.6948**
	DLR with $l_{0/1}$	0.4929	0.716	0.5839	0.6445	0.09	0.9099	0.1638	0.6679
	BC	0.3856	0.6032	0.4705	0.5239	0.2552	0.8161	0.3888	0.6623
	DTPM	0.8685	0.4376	0.5820	**0.6777**	0.6076	0.6164	**0.6120**	0.6289

(a) BC (b) DLR with $l_{0/1}$ (c) DLR with l_{Conf}

Figure 4-8 Some results of the three different recognition strategies. Green and red bounding boxes represent car and pedestrian hypotheses respectively, while blue bounding boxes show the missed ones (either car or pedestrian). The white box within each object bounding box is for contrasting. Though both the *multiclass classification* simulated by DLR with $l_{0/1}$ and our proposed DLR with l_{Conf} predict the direction of the objects, they are theoretically different. The former treats the different output states equally while the later differentiates them according to the semantic structure among them. We illustrate the predicted semantic outputs by the visual signs of them. Note that for DLR, the sign not only stands for itself, but also denotes the semantic level it stays. The first row shows the most common results, while the second and the third ones are two typical exceptions.

Category-level recognition performance. We calculate the precision, recall and F1 value of each strategy on the category-level recognition (i.e. detection) only, as presented in Table 4-1. By varying the bias value of the non-object category, we can get the precision-recall curves as shown in Figure 4-9, which represents more information about the trained models on the binary object/non-object categorization. Both of them clearly show that DLR with "l_{Conf}" performs superior to the degraded version with "$l_{0/1}$" (i.e. the *multiclass classification*), and the later outperforms the *binary categorization*. This

is reasonable because the *multiclass classification* acts like a multi-component model which represents the object category better than the simple *binary categorization*, while our proposed DLR makes better use of the collaboration among object subcategories than the flat *multiclass classification*.

We also run the state-of-the-art object detection algorithm [77] denoted by "DTPM" on the same test sets for comparison. Note that DTPM is trained on the PASCAL VOC 2008 dataset, so the testing results are not fully comparable. Due to the great variations of the image contents, the percentage of difficult training examples in the PASCAL VOC 2008 dataset is larger than that in the IAIR-CarPed dataset. Therefore compared to DLR, DTPM performs much better on set B than on set A. For cars which has relatively rigid shape and similar appearance, DTPM performs as well as DLR on set B which benefits from the good training data. As far as pedestrians and simple cars are concerned, in which such a benefit doesn't exist, DTPM performs significantly worse than the proposed DLR model.

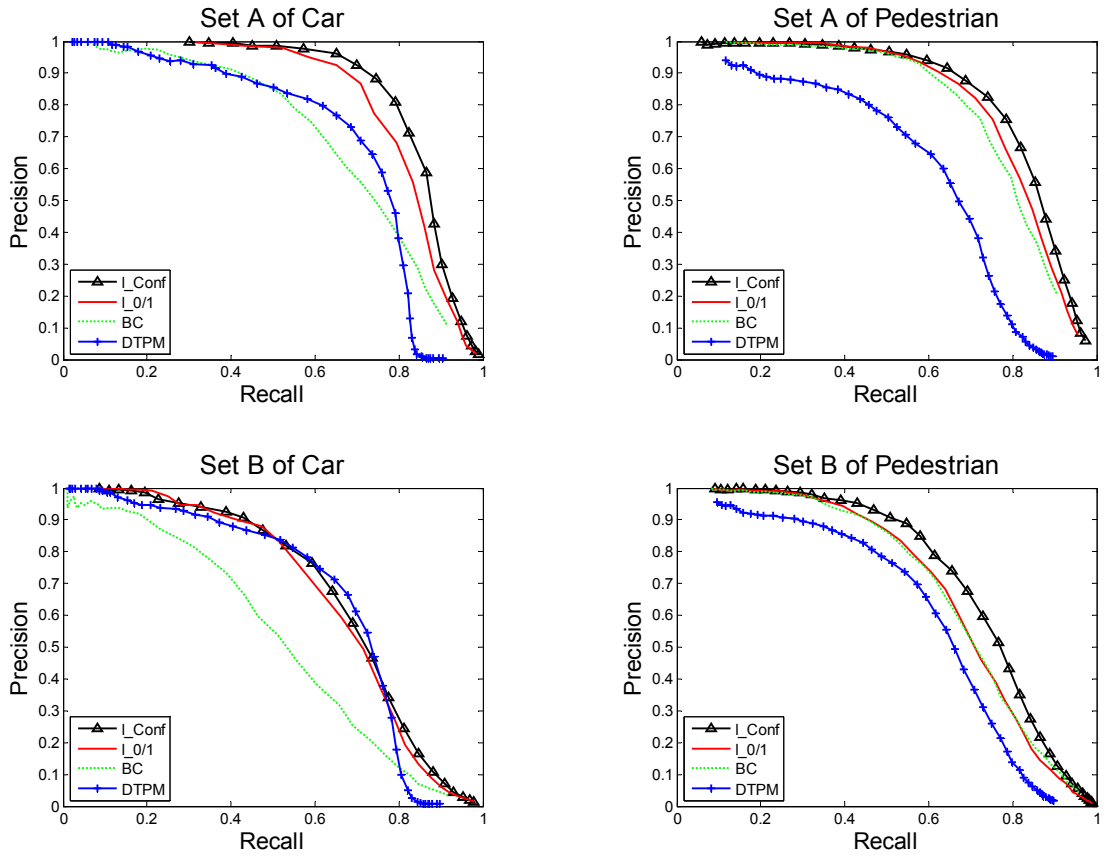

Figure 4-9 Precision-recall curves on category-level recognition.

Categorization performance. We evaluate the overall categorization performance in terms of the summed structural loss of "l_{Conf}" to compare the *multiclass classification* with DLR as this structural loss best represents the recognition confusions of the humans. Table 4-2 shows both the summed within-class (excluding non-object examples)

Table 4-2: Summed categorization losses. The within-class values are normalized by the number of recalled object samples, and the overall values are the original summarizations without normalization due to the large number of non-object samples.

		Within-class		Overall	
		DLR - $l_{0/1}$	DLR - l_{Conf}	DLR - $l_{0/1}$	DLR - l_{Conf}
Car	Set A	0.0427	0.0434	434.02	367.37
	Set B	0.0461	0.0421	1439.1	1156.8
Ped	Set A	0.0651	0.0634	706.86	668.86
	Set B	0.0804	0.0771	23169	11490

categorization confusion losses and the overall categorization confusion losses of them on the two sets (set A and set B) of cars and pedestrians respectively. It can be seen that DLR with "l_{Conf}" is mostly superior to the *multiclass classification* simulated by DLR with loss "$l_{0/1}$", especially when the overall confusion is considered. Since relatively we are putting more efforts on discriminating objects from non-objects as the confusion loss shows, the number of within-class categorization mistakes is usually greater than that of the *multiclass classification* with flat "$l_{0/1}$". This contrast shows that the mistakes of DLR with "l_{Conf}" are closer to those of the humans than the other. A concrete example of their detailed confusion matrices on set B of cars is shown in Figure 4-10. The confusion matrix of DLR with "l_{Conf}" is slightly closer to the human confusion matrix than the other, though they look quite similar. Other results (set A of cars, set A and set B of pedestrians) expose similar information, so they are not presented here. Such close results may due to the domination of object/non-object categorization loss in the whole confusion loss matrix and the margin rescaling strategy does not directly minimizing the misclassification error. Adding the loss function into the primal objective to directly optimize the training loss as suggested by [174] may get better results, which can be our future work.

All the results presented here demonstrate that the structural output modeling by the loss function is critical for the recognition performance, and our proposed DLR is superior to the *binary categorization* and the *multiclass classification* simulated by DLR with loss "$l_{0/1}$" on both category-level recognition (detection) and the overall structured categorization (including localization implicitly).

4.8.2 Effectiveness of The Chosen Features

Though the comparison is mainly done on different recognition strategies represented by the methodology of output structure modeling, the joint input-output feature representation is also very important to the performance of DLR. A good feature set should contain features that are both discriminative and representative for differentiating different output states, and the trained model will represent how these features are used

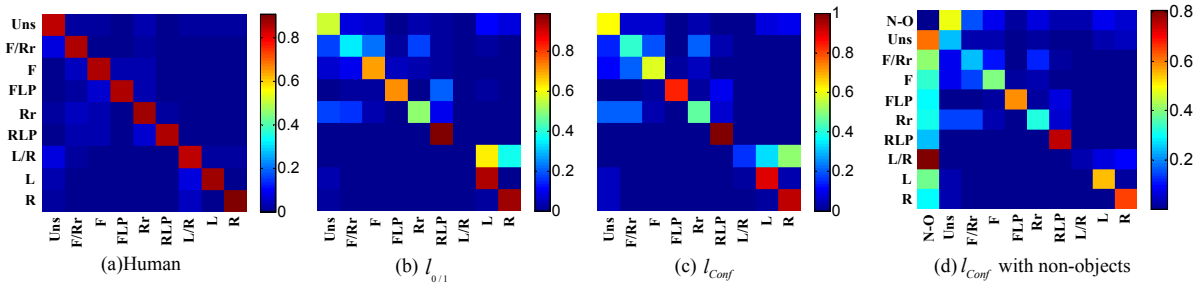

Figure 4-10 Recognition confusion matrices on set B of "car". The first three are within-class object categorization confusions of humans, DLR with zero-one loss (i.e. the *multiclass classification*), and DLR with the confusion loss, respectively. The last one is that of DLR with confusion loss on the whole categorization problem including the non-object category (denoted by "N-O"). Each row is normalized to sum to 1.

for recognition.

For the linear model we have chosen, the prediction is determined by the response scores of each possible output state y as shown in equation 4-21, which is just an inner product of the corresponding weight vector $\mathbf{w}(y)$ and the feature vector $\mathbf{x}$ of the data. It can be viewed as projecting the data to the trained weight vectors, and the projected values (i.e. the response scores) show the discrimination ability of the model on the data. The discrimination ability is represented by the partial ranking as defined in equation 4-7. We calculate the discrimination ability of each type of features (denoted by D^F) on ranking examples with the output y^i (i.e. $\forall \mathbf{x}_k \in \mathbf{S}^i, y_k = y^i$) against any other output y^j as:

$$D_{ij}^F = \frac{1}{|\mathbf{S}^i|} \sum_{k,\mathbf{x}_k \in \mathbf{S}^i} \frac{\left\langle \mathbf{w}^F\left(y^i\right) - \mathbf{w}^F\left(y^j\right), \mathbf{x}_k^F \right\rangle}{\left\langle \mathbf{w}\left(y^i\right) - \mathbf{w}\left(y^j\right), \mathbf{x}_k \right\rangle}$$

in which $\mathbf{w}^F\left(y^i\right)$ represents the part of the weights for feature F on output y^i while $\mathbf{w}\left(y^i\right)$ denotes the whole weight vector on y^i. This measurement exposes the details of how each type of features influences the recognition results.

Figure 4-11 illustrates the discrimination ability of features for DLR on set A of "car" and "pedestrian" respectively. It can be seen that different features have different abilities and the five groups of features are complementary. To just list a few immediate observations: coarse HOG features are very discriminative for differentiating the objects with unspecific direction information from the other output states while fine HOG features do the opposite; fine HOG features are also good at differentiating between frontal/back views and profile views; color works well on telling frontal cars from profile and rear cars and distinguishing frontal pedestrians with clear faces from the back view pedestrians; key part presence feature prevents from false alarming of license plate and missed declaration of the presence of clear faces; the cheep geometric scale feature is very informative for cars and also helpful for differentiating close pedestrians with clear

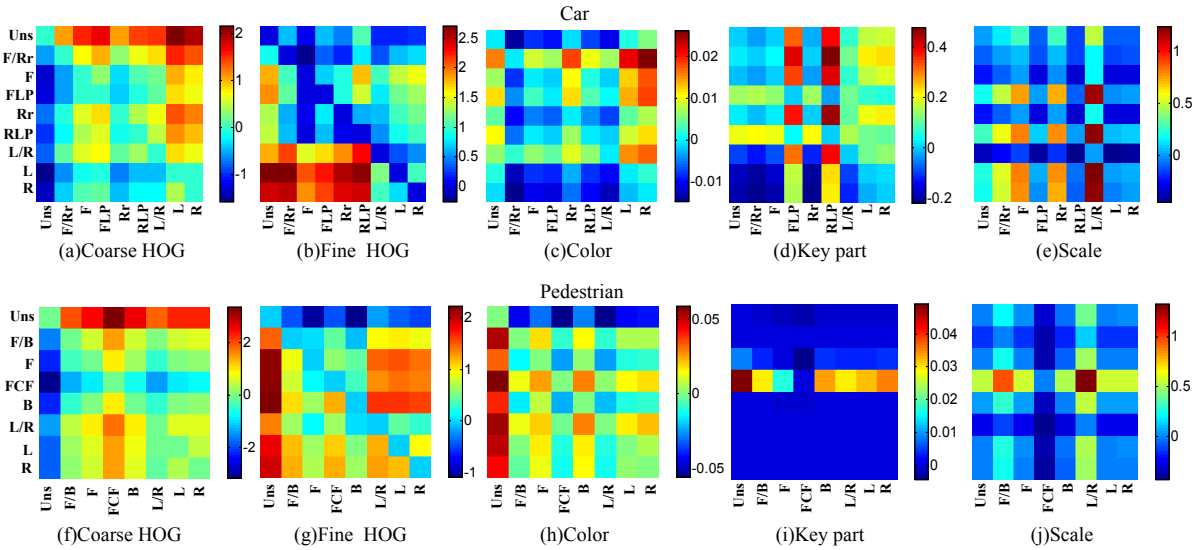

Figure 4-11 Discrimination ability of different groups of features. A large positive value in each matrix mean that the group of features are discriminative for preventing from misclassifying the examples with the output represented by the row to the wrong output label represented by the column. The larger the value is, the stronger the prevention is. In contrast, a large negative value indicates that such a group of features are not suitable for correctly classifying the examples with the output represented by the row.

face from the two directional ambiguous outputs (front/back and left/right). About the relative amount of contributions to the final decision, the high-dimensional HOG features play the most important role, followed by the scale feature. Color and key part presence feature are relatively less deterministic.

To understand more about the high-dimensional HOG features, we illustrate some of the weights for them in Figure 4-12. Note that the weights are used for discrimination (in our case one-against-others partial ranking) but not representation, so higher weights (brighter in the figure) mean that the corresponding features are more discriminative for differentiating the examples with the specific output from those with other outputs.

4.8.3 Robustness to Visual Difficulties

All the experiments were done on both set A and set B of the two object categories. As it can be seen from the category-level recognition results presented in Figure 4-9 and the summed structural loss for the overall categorization in Table 4-2, the visual difficulties do influence the performance of DLR. Compared to set A, the maximum $F1$ value $F1_{max}$ on set B of "car" decreases 17.6% and that of "pedestrian" decreases 11%, while the other two strategies have similar results. It seems better to discuss the robustness of different features against individual difficulties as shown in 3.5.2, which could be our future work.

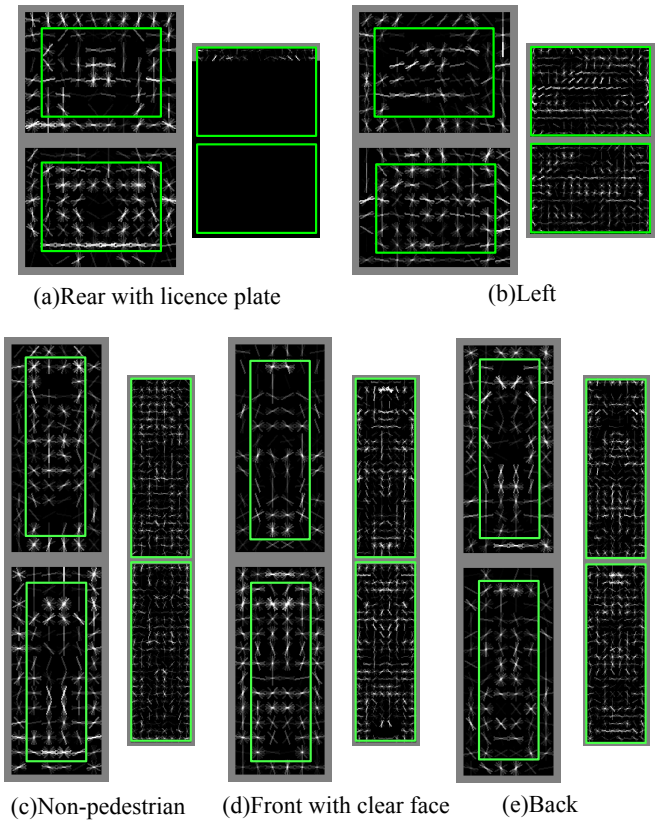

Figure 4-12 Weights of the two-layer HOG features for some outputs. The first two subgraphs are the weights of the "car" category and the other three are those for the pedestrians. In each subgraph, the left part shows the weights for coarse-level HOG features while the part on the right is for fine-level HOG features. The green bounding boxes show the areas of weights for the objects themselves without the close context. The upper and lower parts of each subgraph illustrate the positive values and the magnitude of the negative values of the weights respectively.

Note that the $F1$ value on set B of "pedestrian" is very small due to the high recall and very low precision. This is because the training on this dataset is so hard that the data-mining always terminates on the first few images, resulting in a low-precision model. The high overall structural losses shown in the last line of Table 4-2 also demonstrate such a case. However, it can be adjusted by changing the bias of model.

4.8.4 Computational Complexity

All the experiments were done on a 2.5Ghz 8-core Intel Xeon IBM server. We ran different experiments in parallel using a single thread for each of them. As presented before, we precomputed all the features (except the key part feature of license plate which depends on the training-testing split of the data) for all the images (512×384) in the dataset in advance, and a single thread takes about 25 seconds per image. The computation of the two-layer HOG and the color histograms takes most of the time though we wrote them in C. It dues to the fact that we have a very small lower bound of

the object scale (car 30-pixel width and pedestrian 45-pixel high) and those small scales take most of the time.

For "car", the training of DLR (with l_{Conf} or $l_{0/1}$) takes 1-2 days for set A and 4-5 days for set B, while for "pedestrian" it takes 4-5 days for set A and 9-10 days for set B. The binary categorization is usually 2-4 times faster. The testing or data mining on one image lasts about 20 seconds. If the multiple-core architecture of the machine was used to compute the filter response in parallel as in [77] and [175], the time for testing or data mining would be around 3s per image (plus another 3s for feature precomputation), and the training time will also be much shortened. Further improvements on the speed may be possible using Graphics Processing Unit (GPU) for parallel computation of the sliding-window based recognition as presented in [176].

Note that for both "car" and "pedestrian", set B has nearly twice as many examples as set A and the examples within it are harder for recognition, but the training time scales almost linearly with respect to the number of training examples. It is an advantage of our proposed perceptron-based online learning algorithm.

A group of our deep and layered recognition results are shown in Figure 4-13, which demonstrate that our proposed problem is solvable. Most of the results look good, correctly indicating the right semantic information and localization information of the objects, while some of the others make reasonable mistakes. Results on set B show the generalization and precision tradeoff of the model, on one side it recognizes difficult examples, while on the other side it decreases the precision.

4.9 Discussion

Recall the brief framework presented in 4.3 and the efforts we have made to solve a concrete deep and layered recognition problem, two issues are really critical: a) the structural loss function, and b) output-sensitive feature representation.

The top-down guidance represented by the loss function explores the structural information of the output space and leads the learning in generating a model which can predict desired deep and layered outputs. Therefore, designing a proper loss function is critical. We have mainly focused on semantic structures of the output, but geometric structures like the flexible object bounding box and geometric contextual relationships can be further explored.

Feature representation is important for all recognition problems, but especially critical for deep and layered object recognition. As mentioned before, it dues to the fact that DLR usually has richer output states than the others, and both the differentiation and the sharing relationships among them need to be represented. Our proposed output-sensitive features are good examples for designing proper features. Usually the more complex the output space is, the more designing work needs to be done on the bottom-up feature representation. A good feature set may not only ensure the performance

of recognition but also the efficiency of learning and testing. For the specific DLR instantiation on car and pedestrian recognition, there might be better features than the ones explored by us.

Due to the rich output structure and the high demands on feature representation, usually a large amount of data needs to be provided for training, therefore the efficiency and effectiveness of the learning algorithm becomes rather important. The proposed structured online learning algorithm is a valuable attempt on advancing the tools for solving practical deep and layered object recognition problems.

Compared to other object recognition problems, deep and layered object recognition has several advantages: a) its results are more colorful, more appropriate, and more flexible, b) it has higher category-level recognition performance, and c) it is closer to the way humans recognize objects. The limitations of DLR are two-fold: a) it is only applicable to object categories that have layered within-category properties, and b) currently the annotation for training is more expensive than that of the other recognition problems.

Generally speaking, deep and layered object recognition is harder than traditional object recognition problems, therefore it demands more on both modeling and learning. We hope it can drive the research on object recognition towards deeper and more intelligent stages.

4.10 Conclusion

We have posed a generic model for deep and layered object recognition, followed by detailed discussions on the evaluation and feature representation strategies. The proposed structured online learning algorithm SOnline efficiently solves the concrete deep and layered object recognition problem in the IAIR-CarPed dataset. Comparative results show the superiority of the proposed model on both deep and layered object classification and object detection compared to traditional multiclass recognition and binary recognition models. The experimental results demonstrate that DLR not only generates rich and adaptive outputs, but also improves the performance on traditional object categorization.

Figure 4-13 Some representative results of our DLR algorithm on the IAIR-CarPed dataset. The first 4 rows (a-l) are generated by the model trained on set A, while the results in the last row (m-o) are from the model trained on set B. Green and red bounding boxes represent car and pedestrian hypotheses respectively, while blue bounding boxes show the missed ones. The white box within each object bounding box is for contrasting. Semantic signs are attached close to the hypotheses.

CHAPTER 5
Saliency Based Opportunistic Search for Object Parsing

In this chapter we study the problem of object parsing, which seeks to understand objects in the scene beyond simply identifying their bounding boxes. Unlike the existing approaches which either treat it as a top-down matching problem or perform bottom-up grouping and top-down matching separately and sequentially, we propose a novel strategy called *saliency based opportunistic search* which systematically fuses these two. We tested our approach on a challenging statue face dataset and 3 real human face datasets. Results show that our approach significantly outperforms the predominant Active Shape Models (ASM) using far fewer exemplars. This framework can be applied to other object categories as well.

5.1 Motivation and Contribution

A better understanding of the scene sometimes requires knowing more about the details of the objects within it. For example, recognizing the facial expressions is important for understanding the atmosphere while knowing the poses of humans helps inferencing the ongoing activities and events. For these purposes, we need to *going inside the object's bounding box* to extract object parts and reason about their configurations, which is usually referred to as object parsing.

One common approach to solve the object parsing problem is to learn specific features for each object part and do top-down template matching with geometric constraints [177][178] [95]. A major disadvantage is that it ignores image grouping cues and therefore cannot tell accidental alignment in the background from nonaccidental object segments.

Another widely used strategy is to start with bottom-up segmentation of images and then search for correspondences between object parts in a few shape models and segments in images, i.e. perform bottom-up grouping and top-down matching sequentially [180]. However, segments comprising different object parts in the image are usually not equally salient due to uneven contrast, illumination conditions, clutter, occlusion and pose changes. Moreover, object parts themselves may have different scales. Therefore, depending on the data, segments belonging to different object parts may pop out at different grouping levels (with different numbers of segments) and they may not be predictable, as shown in Figure 5-1. One may choose to do over-segmentation to make sure all segments belonging to the object are covered, but then the saliency will be lost, and fake segmentation boundaries will cause many false positives of accidentally aligned object parts.

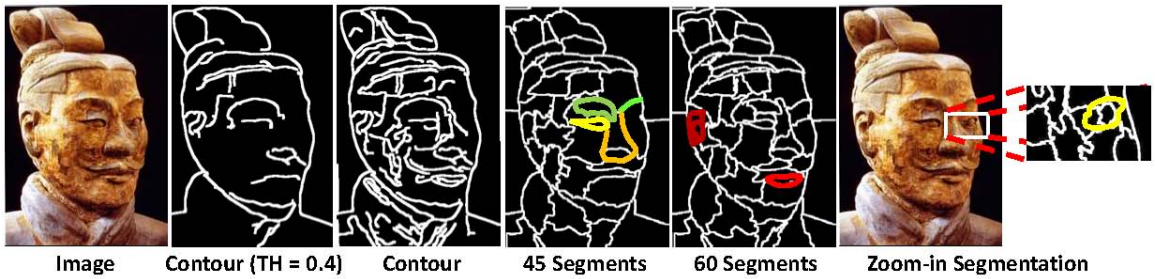

Figure 5-1 Saliency of contours and segments. The second image is a group of salient contours from contour grouping [179] by setting a lower threshold to the average edge strength, while the third one contains all the contours from contour grouping. It shows that by thresholding the saliency of contour segments, we either get some foreground contours missing (under-segmented) or have a lot of clutter come in (over-segmented). The same thing happens to image segmentation. Segments comprising object parts pop out in different grouping levels, representing different saliencies (cut costs). The last three images show such a case.

However, it is not easy to incorporate saliency. A naive way of using saliency is to find salient parts first, and then search for less salient ones condition on them. The drawback is that a hard decision has to be made in the first step of labeling salient parts, and mistakes arising from this step cannot be recovered later.

To make the best use of segment saliency for efficient object parsing while at the same time ensure its correctness and robustness, we propose a novel approach called *Saliency Based Opportunistic Search* with the following key contributions:

1. We novelly formulate the object parsing problem as a fused grouping and matching problem so that bottom-up image saliency is naturally incorporated. The overall cost function biases on matching salient segments which are more likely to be those of the objects while it also allows local editings to get the less salient ones guided by the previously matched ones.

2. Two levels of contextual information (figural object-level context and semantic part-level context) are used to ensure the correctness of matching results. The figural context is a global representation captures the overall shape of the object while semantic context serves as a global constraint for local part representations.

3. An opportunistic search strategy is proposed to well bridge the bottom-up segment grouping and top-down shape matching for object parsing. In the search process, bottom-up grouping is only adopted locally which ensures efficiency and precision, and the newly proposed part hypotheses based on these locally generated segments are matched together with all the formerly generated part hypotheses therefore it avoids falling into local minimums.

5.2 Related Work

It has been shown that humans recognize objects by their components [181] or parts [182]. The main idea is that object parts should be extracted and represented together with the relationships among them for matching to a model. This idea has been widely used for the task of recognize objects and their parts [95, 183, 184]. Figural and semantic contextual information play an important role in solving this problem. Approaches that take advantage of figural context include PCA and some template matching algorithms such as Active Shape Models (ASM) [185] and Active Appearance Models (AAM) [186]. Template matching methods like ASM usually use local features (points or key points) as searching cues, and constrain the search by local smoothness or acceptable variations of the whole shape. However, these methods require good initialization. They are sensitive to clutter and can be trapped in local minima. Another group of approaches are part-based models, which focus on semantic context. A typical case is pictorial structure [95]. Its cost function combines both the individual part matching cost and pair-wise inconsistency penalties. The drawback of this approach is that it has no figural context measured by the whole object. It may end up with many "OK" part matches without a global verification, especially when there are many faint object edges and occlusions in the image. Recently, a multiscale deformable part model was proposed to detect objects based on deformable parts [177], which is an example that uses both types of contextual information. However, the part-based model is just for improving the detection results by pursuing better data alignment, and the parts have no semantic meanings.

5.3 Parsing by Fusing Grouping with Matching

5.3.1 Problem Definition and Modeling

We perform object parsing via a fusion of two operations: object segment (contour or region boundary) extraction and segment labeling, therefore it naturally combines the bottom-up image-based grouping and top-down model-based matching. To handle intra-class variations of the object, multiple models may be needed (Figure 5-2 shows the models used by us for face parsing). Therefore, finding the best model is also involved in the object parsing problem. Concretely, the problem can be formulated as follows:
Input:

- Model: $M = \{M_1, M_2, \ldots, M_m\}$; each model M_k has a set of labeled parts $\{p_1^k, p_2^k, \ldots, p_n^k\}$. They are all shape models made of contours and line segments.
- Image: I containing at lease one object instance.

Output:

- The best matched model M_k for each object instance.

- Extracted candidate segments $S = \{s_1, s_2, \ldots, s_l\}$, which are region boundaries and contours extracted at different grouping levels from the image (see Figure 5-1 for an example).
- Object part labels $L(S)$. If there exists a part p_j^k and s_i belongs to part p_j^k, then $L(s_i) = j$, or else $L(s_i) = 0$.

Figure 5-2 Different models for faces. They are hand designed models obtained from 7 real images, each of them representing one pose. Facial features are labeled in different colors.

Such a problem contains two highly correlated subproblems: bottom-up segment grouping and top-down shape matching. Therefore, we formulate the problem as minimizing the following cost function:

$$C^{parsing} = C^{matching} + C^{grouping} \tag{5-1}$$

where $C^{matching}$ measures the shape matching cost between shape models and labeled segments in the image, which relies much on the *correspondence* and the *context*, and $C^{grouping}$ is the grouping cost, which can be measured in different ways and in this paper it is mainly about the bottom-up *saliency* based editing cost.

The cost function above is based on the following three key issues.

1. **Correspondence** (u). *A way to measure the dissimilarity between a shape model and a test image.* The correspondence is defined on control points. Features computed on these control points represent the shape information and then the correspondences are used to measure the dissimilarity. Let $U^{\mathcal{M}} = \{a_1, a_2, \ldots, a_{N_a}\}$ be a set of control points on the model, and $U^{\mathcal{I}} = \{b_1, b_2, \ldots, b_{N_b}\}$ be the set on the image. We use u_{ij} to denote the correspondence between control points a_i and b_j where $u_{ij} = 1$ indicates they are matched, otherwise $u_{ij} = 0$. Note that this correspondence is different from the one between object parts and image segments.
2. **Context (x and y)**. *The idea of using the context is to* ***choose the correct context*** *on both model and test image sides for shape matching invariant to clutter and occlusion.* **x** and **y** are used here to denote the context selection of either segments or parts on the model and the image, respectively.
3. **Saliency**. *A property of bottom-up segments which represents how difficult it is to separate the segment from the background.* Coarse-level segmentation tends to produce salient segments, while finer-level segmentation extracts less salient ones,

but at the same time introduces background clutter. Local editing on the salient gap between two salient segments can help to get good segments out without bringing in clutter, but it needs contextual guidance.

Saliency based editing. Image segmentation may be hard to generate the right object segments when they have different saliency values. Under-segmentation could end up with unexpected leakages, while over-segmentation may introduce clutter. A solution for this problem is to do some local editings. For example, adding a small virtual edge at the leakage place can make the segmentation much better without increasing the number of segments. *Zoom-in* in a small area is also a type of editing that can be effective and efficient, as presented in Figure 5-1. **Small costs for editing can result in big improvement on shape matching cost**. This is based the shape integrity and the non-additive distance between shapes. However, editings need the contextual information from the model.

Suppose there are a set of possible editing operations $\mathbf{z}$ which might lead to better segmentation. $z_k = 1$ means that editing k is chosen, otherwise $z_k = 0$. Note that usually it is very hard to discover and precompute all the editings beforehand. Therefore, this editing index vector $\mathbf{z}$ is dynamic, and it appends on the fly. After doing some editings, some new segments/(part hypotheses) will come out, meanwhile we can still keep the original segments/parts. Therefore, a new variable $\mathbf{y}^{edit} = \mathbf{y}^{edit}(\mathbf{y}, \mathbf{z})$ is used to denote all the available elements which includes both the original ones in $\mathbf{y}$ and the new ones induced by editing $\mathbf{z}$. Let C_k^{edit} be the edit cost for editing k.

Our cost function (5-1) of object parsing can be written as follows:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}} \quad & C^{parsing}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) = C^{matching}(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{u}) + C^{grouping}(\mathbf{z}) = \\ & \sum_{i=1}^{N_a}[\ \beta \cdot \sum_{j=1}^{N_b} u_{ij} C_{ij}^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit}) + C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})] + \sum_k C_k^{edit}\mathbf{z}_k \qquad (5\text{-}2) \\ s.t. \quad & \textstyle\sum_j u_{ij} \le 1, \quad i = 1, ..., N_a \end{aligned}$$

$\mathbf{x}$: *selection indicator of model segments/parts.*
$\mathbf{y}$: *selection indicator of image segments/parts.*
$\mathbf{z}$: *selection vector of editing operations.*
$\mathbf{u}$: *correspondence of control points between the image and model.*
$\mathbf{y}^{edit}(\mathbf{y},\mathbf{z})$: *selection indicator of image segments/parts edited by* $\mathbf{z}$.

The three summations in equation (5-2) correspond to three different types of cost: *mismatch cost* $C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit},\mathbf{u})$, *miss cost* $C^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})$ and *edit cost* $C^{edit}(\mathbf{z})$. The mismatch cost, $C_{ij}^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit}) = \|f_i(\mathbf{x}) - f_j(\mathbf{y}^{edit})\|$ denotes the feature dissimilarity between two corresponding control points. To prevent the cost function from biasing to fewer matches, we add the miss cost $C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x}) = \|f_i^{full} - (\sum_j u_{ij}) f_i(\mathbf{x})\|$ to denote how much of the model has not been matched. It encourages more parts to be matched

on the model side. There is a trade-off between $C_{ij}^{\mathcal{M}\leftrightarrow\mathcal{I}}$ and $C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}$, where $\beta \geq 0$ is a controlling factor. Note that $\|\cdot\|$ can be any norm function[①]. The following subsections focus on the two parts of our cost function. Shape matching is performed on two levels of contexts and saliency based editing results in the opportunistic search approach.

5.3.2 Two-level Context Based Shape Matching

We extend the shape matching method called contour context selection in [180] to two different contextual levels: "figural context selection" and "semantic context selection".

Figural context selection. Figural context selection matches a segment-based holistic shape model to an object hypothesis represented by segments, which may have clutter and missing segments. We optimize the following cost function:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{u}} \quad C^{figural}(\mathbf{x},\mathbf{y},\mathbf{u}) = \sum_{i=1}^{N_a}[\ \beta \cdot \sum_{j=1}^{N_b} u_{ij} \underbrace{\|SC_i^{\mathcal{M}}(\mathbf{x}) - SC_j^{\mathcal{I}}(\mathbf{y})\|}_{C_{ij}^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})} + \underbrace{\|SC_i^{\mathcal{F}} - (\textstyle\sum_j u_{ij}) \cdot SC_i^{\mathcal{M}}(\mathbf{x})\|}_{C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})})]$$

$$\text{s.t.} \sum_{i,j,i',j'} u_{ij} u_{i'j'} C^{geo}_{i,j,i',j'} \leq C_{tol} \tag{5-3}$$

where $SC_i^{\mathcal{M}}(\mathbf{x})$ and $SC_j^{\mathcal{I}}(\mathbf{y})$ is defined as the Shape Context centered at model control point a_i and image control point b_j. $C^{geo}_{i,j,i',j'}$ is the geometric inconsistent cost of correspondences $\mathbf{u}$. C_{tol} is the maximum tolerance of the geometric inconsistency. We use Shape Context [10] as our feature descriptor. Note that the size of Shape Context histogram is large enough to cover the whole object model, and this is a set-to-set matching problem. Details for this algorithm can be found in [180].

Semantic context selection. Similarly we explore semantic context to select consistent object part hypotheses. We first generate part hypotheses using almost the same context selection algorithm as the one presented above. The selection operates on parts instead of the whole object. Figure 5-3 shows an example of generating a part hypothesis.

In semantic context selection, we reason about semantic object parts. Hence we abstract each part (on either model or test image) as a point located at its center with its part label. We place control points on each one of the part centers.

Suppose C_j^{part} is the matching cost of part hypothesis j. We use $w_j^P = \frac{e^{\gamma C_j^{part}}}{e^{\gamma}} \in [\frac{1}{e^{\gamma}}, 1], \gamma \in [0,1]$ as its weight. Then the cost function for semantic context selection is:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{u}} \quad C^{semantic}(\mathbf{x},\mathbf{y},\mathbf{u}) = \sum_{i=1}^{N_a}[\ \beta \cdot \sum_{j=1}^{N_b} u_{ij} \underbrace{w_j^P \|SC_i^{\mathcal{M}}(\mathbf{x}) - SC_j^{\mathcal{I}}(\mathbf{y})\|}_{C_{ij}^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})} + \underbrace{\|SC_i^{\mathcal{F}} - (\textstyle\sum_j u_{ij}) \cdot SC_i^{\mathcal{M}}(\mathbf{x})\|}_{C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})})] \tag{5-4}$$

[①] In our shape matching we used L_1 norm.

The variable definitions are similar to figural context selection, except for two differences: 1) selection variables depend on the correspondences and 2) Shape Context no longer counts edge points, but object part labels.

The desired output of $L(S)$ is implicitly given in the optimization variables. During part hypothesis generation, we put labels of candidate parts onto the segments. Then after semantic context selection, we confirm some labels and discard the others using the correspondence u_{ij} between part candidates and object part models.

5.3.3 Saliency Based Opportunistic Search

Object parsing using saliency based editing potentially requires searching over a very large state space. Matching object shape and its part configuration requires computing correspondences based on non-local context. Both of them have exponentially many choices. On top of that, we need to find a sequence of editings, such that the resulting segments and parts produced by these editings are good enough for matching.

The key intuition of our saliency based opportunistic search is that we start from coarse segmentations which produce salient segments and parts to guarantee low saliency cost. We iteratively match configuration of salient parts to give a sequence of bounds to the *search zone* of the space which needs to be explored. The possible spatial extent of the missing parts is bounded by their shape matching cost and the edit cost (equally, saliency cost). Once the search space has been narrowed down, we "zoom-in" to the finer scale segmentation to rediscover missing parts (hence with lower saliency). Then we "zoom-out" to do semantic context selection on all the part hypotheses. The new selection result improves the bound on the possible spatial extent and might suggest new search zones. This opportunistic search allows both high *efficiency* and high *accuracy* of object part labeling. We avoid extensive computation by narrowing down the search zone. Furthermore, we only explore less salient parts if there exist salient ones supporting them, which avoids producing many false positives from non-salient parts.

Search Zone. In each step t of the search, given $(\mathbf{x}^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{z}^{(t-1)}, \mathbf{u}^{(t-1)})$, we use

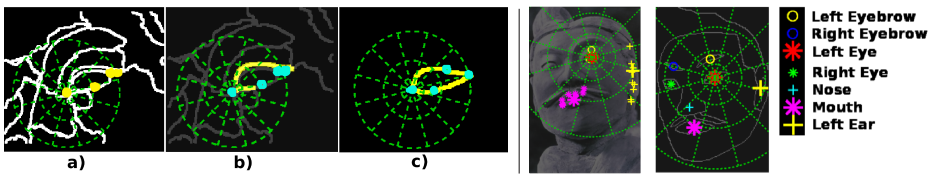

Figure 5-3 Semantic context selection. Left: Part hypothesizing. a) A local part region around the eye in the image, with segments and control points. c) A model template of the eye with control points. Selection result on the image is shown in b). Right: Consistent part grouping. Semantic-level shape context centered on the left eye captures semantic contextual information of the image. A subset of those parts form a mutually consistent context and we group them by matching with the semantic-level shape context on the model shown in the middle.

$\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})$ to denote the increment of $C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})$ (the first summation in equation (5-2)). $\Delta C^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})$ and $\Delta C^{edit}(\mathbf{z})$ are similarly defined. By finding missing parts, we seek to decrease the cost (5-2). Therefore, we introduce the following criterion for finding missing parts:

$$\beta\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y},\mathbf{z})+\Delta C^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u})+\Delta C^{edit}(\mathbf{z})\leq 0 \tag{5-5}$$

We write $C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y},\mathbf{z})=C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})$ since $\mathbf{y}^{edit}$ depends on editing vector $\mathbf{z}$.

The estimation of bounds is based on the intuition that if all the missing parts can be found, then no *miss* cost is needed. Therefore, according to equation (5-4):

$$\Delta C^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x})\geq-\sum_i C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x},\mathbf{u}). \tag{5-6}$$

This is the upper bound for the increment of either one of the other two items in equation (5-5) when any new object part is matched.

Suppose a new editing $\mathbf{z}_\alpha^{(t)}=1|_{\mathbf{z}_\alpha^{(t-1)}=0}$ matches a new object part a_k to a part hypothesis in the image b_ℓ. Let $k\leftrightarrow\ell$ indicate $u_{k\ell}^{(t)}=1$ and $\sum_j u_{kj}^{(t-1)}=0$. Then this editing at least has to pay the cost of matching a_k to b_ℓ (we do not know whether others will also match or not):

$$C|_{k\leftrightarrow\ell}=\beta\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y},\mathbf{z})|_{k\leftrightarrow\ell}+C_\alpha^{edit}. \tag{5-7}$$

The first item on the right of equation (5-7) is the increment of *mismatch* $\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y}^{edit})$ when a new object part a_k get matched to b_ℓ. It can be computed based on the last state of the variables $(\mathbf{x}^{(t-1)},\mathbf{y}^{(t-1)},\mathbf{z}^{(t-1)},\mathbf{u}^{(t-1)})$. According to above equations,

$$\beta\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y},\mathbf{z})|_{k\leftrightarrow\ell}+C_\alpha^{edit}-\sum_i C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})\leq 0. \tag{5-8}$$

Since we use Shape Context for representation and matching, the *mismatch* is nondecreasing. And also the editing cost is nonnegative, so we abtain the bounds for the new editing $\mathbf{z}_\alpha^{(t)}=1|_{\mathbf{z}_\alpha^{(t-1)}=0}$. Let $\mathcal{Z}(k)$ denote the search zone for object part k. Then we can compute two bounds for $\mathcal{Z}(k)$:

$$\textbf{(Supremum)}\quad \mathcal{Z}^{sup}(k)=\{\mathbf{z}_\alpha|\Delta C^{\mathcal{M}\leftrightarrow\mathcal{I}}(\mathbf{x},\mathbf{y},\mathbf{z})|_{k\leftrightarrow\ell}\leq\frac{1}{\beta}\sum_i C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})\} \tag{5-9}$$

$$\textbf{(Infimum)}\quad \mathcal{Z}^{inf}(k)=\{\mathbf{z}_\alpha|C_\alpha^{edit}\leq\sum_i C_i^{\mathcal{F}\leftrightarrow\mathcal{M}}(\mathbf{x}^{(t-1)},\mathbf{u}^{(t-1)})\} \tag{5-10}$$

where $\mathcal{Z}^{sup}$ gives the supremum of the search zone, *i.e.* upper bound of zoom-in window size, and $\mathcal{Z}^{inf}$ gives the infimum of the search zone, *i.e.* lower bound of zoom-in window size. When the number of segments is fixed, the saliency of the segments decreases as the window size becomes smaller. $\mathcal{Z}^{sup}$ depends on *mismatch* and $\mathcal{Z}^{inf}$ depends on the *edit* cost (*i.e.* **saliency**). In practice, one can sample the space of the search zone, and check which ones fall into these two bounds.

Our opportunistic search is summarized in Algorithm 4.

Algorithm 4 Saliency Based Opportunistic Search

1: Initialize using figural context selection. *For each part k, compute $\mathcal{Z}(k)$ based on u from figural context selection. Set $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \mathbf{z}^{(0)}, \mathbf{u}^{(0)})$ to zeros. Set $t = 1$.*

2: Compute search zones for all the missing parts. *Find all missing parts by thresholding the solution $\mathbf{x}^{(t-1)}$.*

for each missing part p_k

If $\mathcal{Z}(k) = \emptyset$, compute search zone set $\mathcal{Z}(k)$ by equation (5-9) and (5-10).

end

3: Zoom-in search zone. *Update editing set $\mathbf{z}$.*

for each $x_k^{(t-1)}$ where $\mathcal{Z}(k) \neq \emptyset$

Perform Ncut segmentation for each zoom-in window indexed by elements in $\mathcal{Z}(k)$.

Generate part hypotheses. Set $\mathcal{Z}(k) = \emptyset$.

If no candidates can be found, go to the next missing part.

Update $\mathbf{z}$ from part hypotheses.

end

4: Evaluate configurations with re-discovered parts.

Terminate if $\mathbf{z}$ does not change.

Update $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{z}^{(t)}, \mathbf{u}^{(t)})$ with the rediscovered parts using equation (5-4).

Terminate if $C^{semantic}(\mathbf{x}, \mathbf{y}, \mathbf{u})$ does not improve.

$t = t + 1$. Go to step 2.

5.4 A Case Study: Face Parsing

5.4.1 Instantiation of Framework

We present more details on the opportunistic search using faces as an example in Figure 5-4. We found that usually the whole shape of the face is more salient than individual facial parts. Therefore, the procedure starts with figural context and then switches to semantic context. We concretize our algorithm for this problem in the following steps. The same procedure can be applied to similar objects.

1. **Initialization: Object Detection.** Any object detection method can be used, but it is not a necessary step (figural context selection can also be used to do that [180]). We used shape context voting [187] to do this task, which can handle different poses using a small set of positive training examples.

2. **Context-based Alignment.** First, use $C^{figural}$ in equation (5-3) to select the best matched model M_k and generate the correspondences $u^{figural}$ for rough alignment①. When the loop comes back again, update the alignment based on $u^{semantic}$. Estimate locations for other still missing parts.

① In practice, we kept the best two model hypotheses and finally chose the one with lower overall cost.

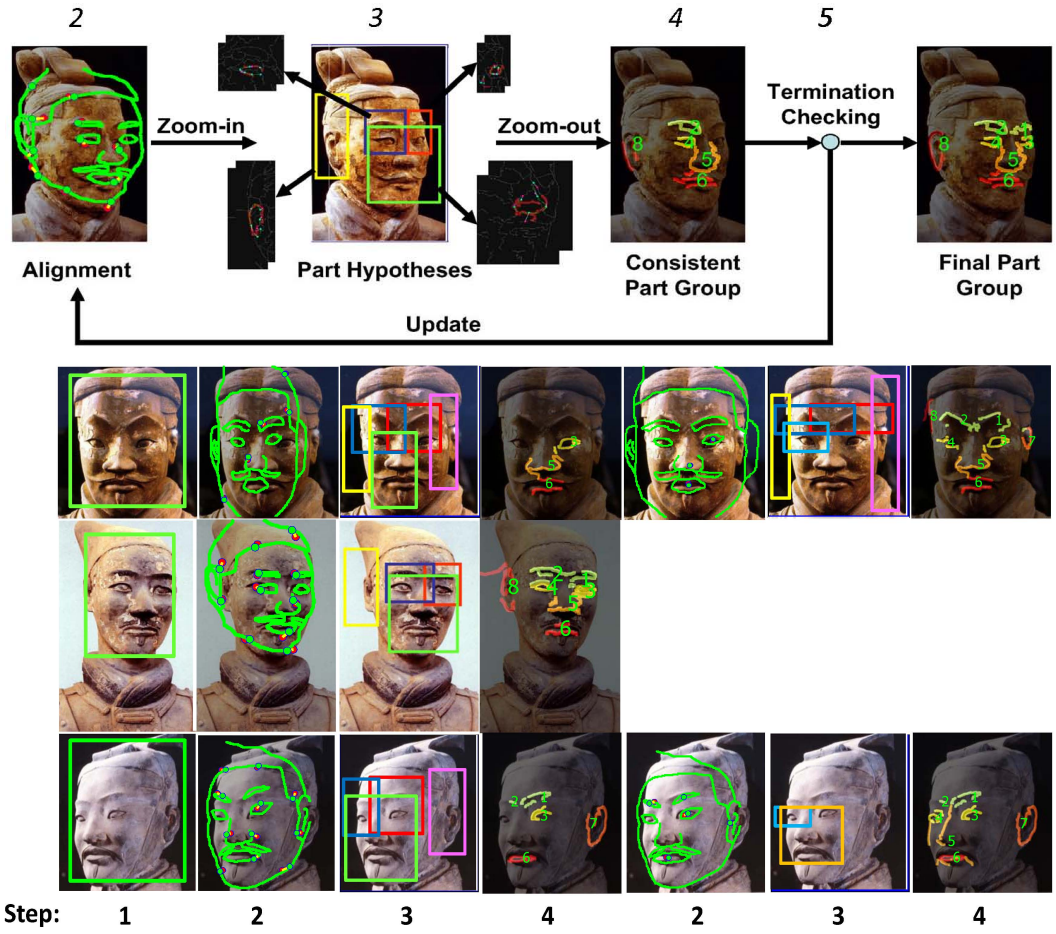

Figure 5-4 Saliency based opportunistic search, using faces as an example. Top: the flowchart. Bottom: results of each step for 3 different examples. Typically the iteration converges after only one or two rounds. Rectangles with different colors indicate the zoom-in search zones for different parts. Note that when zoom-in is performed for the first time, two adjacent parts can be searched together for efficiency. This figure is best viewed in color.

3. **Part Hypotheses Generation.** Zoom in on these potential part locations by cropping the regions and do Ncut segmentation to get finer scale segmentation. Then match them to some predefined part models. The resulting matching score is used to prune out unlikely part hypotheses, according to the bound of the cost function.

4. **Part Hypotheses Grouping.** Optimize $C^{semantic}$ in equation (5-4). Note that the best scoring group may consist of only a subset of the actual object parts.

5. **Termination Checking.** If no better results can be obtained, then we go to the next step. Or else we update **semantic context** and go back to step 2.

6. **Extracting Facial Contours.** This is a special step for faces only. With the final set of facial parts, we optimize $C^{figural}$ again to extract the segments that correspond to the face silhouette, which can be viewed as a special part of the face.

5.4.2 Implementing Two-level Context Selection

For simplification, we do not consider any editing in figural context selection. Then equation (5-3) is an integer programming problem, we relaxed the variables to solve it with LP. Details of this context selection algorithm can be found in the paper of Zhu et al.[180].

For semantic context selection, we need to search for correspondences and part selection variables simultaneously because they are highly dependent, unlike the situation in figural context selection. Therefore, we introduce a *correspondence context vector* $P_{ij}^{\mathcal{M}} = u_{ij}\mathbf{x}$ to expand the selection space for model parts:

$$P_{ij}^{\mathcal{M}} \in \{0,1\}^{|U^{\mathcal{M}}|} : P_{ij}^{\mathcal{M}}(i') \Leftrightarrow u_{ij} = 1 \wedge \mathbf{x}(i') = 1 \tag{5-11}$$

Similarly, we define the *correspondence context vector* for image parts,

$$P_{ij}^{\mathcal{I}} \in \{0,1\}^{|U^{\mathcal{I}}|} : P_{ij}^{\mathcal{I}}(j') \Leftrightarrow u_{ij} = 1 \wedge \mathbf{y}(j') = 1 \tag{5-12}$$

In addition to the cost in equation (5-4), constraints on *context correspondence vector* $P^{\mathcal{M}}, P^{\mathcal{I}}$ are enforced such that the semantic context viewed by different parts are consistent with each other. These constraints are summarized by the table 5-1. The cost function and constraints are linear. We relaxed the variables and solved it with LP.

Table 5-1: Constraints on context correspondence vector $P^{\mathcal{M}}, P^{\mathcal{I}}$. For example, *Context completeness* requires that contexts must include all the matched parts. If both i and i' are matched parts, the context viewed from i must include i', *i.e.* $(\mathbf{y}(i) = 1) \wedge (\mathbf{y}(i') = 1) \Rightarrow \sum_j P_{ij}^{\mathcal{M}}(i') = 1$, which is relaxed as the constraint in row 4. Other constraints are constructed in a similar way.

Constraint Name	Formulation
Self consistency	$\sum_j P_{ij}^{\mathcal{M}}(i) = \mathbf{y}(i),\ \sum_i P_{ij}^{\mathcal{I}}(j) = \mathbf{x}(j)$
One-to-one matching	$\sum_i P_{ij}^{\mathcal{M}}(i') \le \mathbf{y}(i'),\ \sum_j P_{ij}^{\mathcal{M}}(i') \le \mathbf{y}(i')$ $\sum_i P_{ij}^{\mathcal{I}}(j') \le \mathbf{x}(j'),\ \sum_j P_{ij}^{\mathcal{I}}(j') \le \mathbf{x}(j')$
Context reflexitivity	$P_{ij}^{\mathcal{M}}(i') \le P_{ij}^{\mathcal{M}}(i),\quad P_{ij}^{\mathcal{I}}(j') \le P_{ij}^{\mathcal{I}}(j)$
Context completeness	$\mathbf{y}(i) - \sum_j P_{ij}^{\mathcal{M}}(i') \le 1 - \mathbf{y}(i'),\quad \mathbf{x}(j) - \sum_i P_{ij}^{\mathcal{I}}(j') \le 1 - \mathbf{x}(j')$
Mutual context support	$\sum_j P_{ij}^{\mathcal{M}}(i') = \sum_{j'} P_{i'j'}^{\mathcal{M}}(i),\quad \sum_i P_{ij}^{\mathcal{I}}(j') = \sum_{i'} P_{i'j'}^{\mathcal{I}}(j)$

5.5 Experiments and Results

Datasets. We tested our approach on both statue faces from the Emperor-I dataset [188] and real faces from various widely used face databases (UMIST, Yale, and Caltech Faces). Quantitative comparison was done on the Emperor-I dataset and we also show some qualitative results on a sample set of all these datasets. The statue face dataset

has some difficulties that normal faces do not have: lack of color cue, low contrast, inner clutter, and great intra-subject variation.

Comparison measurement. The comparison is between Active Shape Models [185] and our approach. Since we extract facial parts by selecting contours, our desired result is that the extracted contours are all in the right places and correctly labeled. However, ASM generates point-wise alignment between the image and a holistic model. Due to the differences, we chose to use "normalized average point alignment error" measurement for alignment comparison.

Since our results are just labeled contours, we do not have point correspondences for computing the point alignment error. Therefore, we relaxed the measurement to the distance between each ground truth key point and its closest point on the contours belong to the same part. To make the comparison fair, we have exactly the same measurement for ASM by using spline interpolation to generate "contours" for its facial parts. We use 0.35 times the maximum height of the ground truth key points as an approximation of the distance between two eyes invariant to pose changes as the our normalizing factor.

Experiments. There are two aspects of our Emperor-I dataset that may introduce difficulties for ASM: few training examples with various poses and dramatic face silhouette changes. Therefore, we designed three variants of ASM to compensate for these challenges, denoted in our plots as "ASM1","ASM2","ASM3". Table 5-2 shows the differences. Basically, ASM2 and ASM3 disregard face silhouette and work on fewer poses that may have relatively more exemplars. Note that ASM3 even combined the training data of the three near-frontal poses as a whole. We used "leave-one-out" cross-validation for ASM. For our method, we picked up 7 images for different poses (one for each pose), labeled them and extracted the contours out to work as our holistic models. Moreover, we chose facial part models (usually combined by 2 or 3 contours) from a total of 23 images which also contained these 7 images. Our holistic models are shown in Figure 5-2 and Figure 5-5 shows those averaged ones for ASM.

Table 5-2: Comparison of experimental details on Emperor-I dataset

Method	No. of Poses	Silhouette	No. of Training	No. of Test	Average point error
ASM1	7	w	138	86	0.2814
ASM2	5	w/o	127	81	0.2906
ASM3	3	w/o	102	70	0.3208
Ours	7	w	7+16	86	0.1503

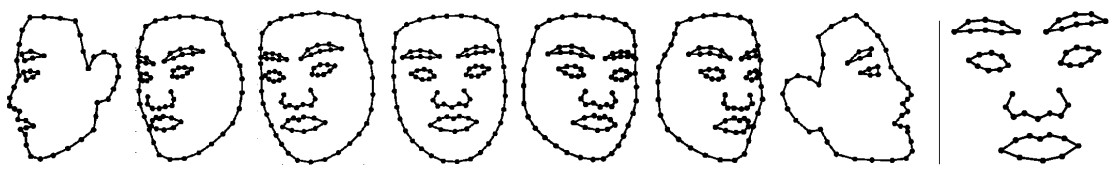

Figure 5-5 Left: averaged models for ASM1. Right: averaged model for ASM3.

In Figure 5-6, we show the alignment errors for all the facial parts together and

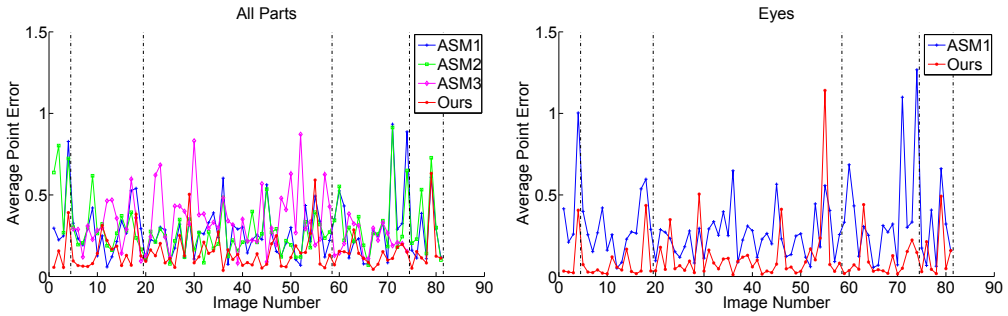

Figure 5-6 Average point error vs. image number. All the values are normalized by the estimated distance of two eyes in each image. The vertical dot-dash lines separate images of different poses.

Table 5-3: Average error, normalized by distance between eyes for ASM vs. our method

Method	Global	Eyebrows	Eyes	Nose	Mouth	Silhouette
ASM1	0.3042	0.2923	0.2951	0.2715	0.2524	0.3126
Ours	0.1547	0.2015	0.1142	0.1546	0.1243	0.1353

also those only for the eyes. Other facial parts have similar results so we leave them out. Instead, we provide a summary in Table 5-3 and a comparison in the last column of Table 5-2, where each entry is the mean error across the test set or test set fold, as applicable. We can see that our method performs significantly better than ASM on all facial parts with significantly fewer training examples. We provide a qualitative evaluation of the results in Figure 5-7, where we compare the result of ASM and our method on a variety of images containing both statue faces and real faces. These images show great variations, especially of those statue faces. Note that the models are only trained on statue faces.

5.6 Conclusion

We proposed a heuristic object parsing framework which incorporates two-level contexts and a saliency based opportunistic search strategy. The combination of figural context defined on the whole object shape and semantic context defined on object parts ensures the correctness of the chosen model and the object parsing results. Saliency based opportunistic search provides an efficient and effective way to combine the bottom-up segment grouping and the top-down shape matching for parsing the objects with great intra-class variations and inner clutters. Experimental results on several challenging face datasets demonstrate that our approach can accurately label object parts such as facial features and resist to accidental alignment.

Figure 5-7 A subset of the results. Upper group is on the Emperor-I dataset and the lower is for real faces from various face databases (1-2 from UMIST, 3-4 from Yale, and 5-7 from Caltech). Matched models, control points and labeled segments are superimposed on the images.

CHAPTER 6
Conclusion and Future Directions

6.1 Conclusion

This thesis focuses on promoting the research on two key components of scene understanding: object recognition and object parsing. Based on a systematic overview of the 50 years' research on object recognition and parsing, four core and unsolved issues have been summarized, which have motivated the research in this thesis. For object recognition, a novel problem of deep and layered recognition inspired by human vision is proposed along with a benchmark dataset annotated via a strict psychophysical experiment, then a generic strategy for solving this problem has been presented with outperforming experimental results on the benchmark dataset compared to other recognition strategies. The realization of the proposal is ensured by a new efficient structured online learning algorithm with great scalability as stated in this thesis. For object parsing, a novel approach utilizing both local and global representations by fusing the bottom-up segment grouping and top-down shape matching is proposed, which has great robustness against heavy clutters and partial occlusions. The key idea of it is also applicable to other tasks like scene parsing.

More concretely, this thesis advances the research in the following aspects.

1. **Proposing a novel problem called *deep and layered object recognition* for adaptively interpreting the objects, and building a new benchmark dataset (IAIR-CarPed) for it.** The annotated results of IAIR-CarPed show that human rapid recognition of objects without specific visual difficulties is obviously layered, which reveals the fact that in their daily lives the object recognition results of humans depend on the input stimuli. This dataset and the human recognition results can serve as a new benchmark for deep and layered object recognition, and the evaluation criterion based on the human confusions between different semantics can well represent the performance of a computer vision algorithm compared to the human recognition results. Unlike other datasets, we annotated the visual difficulties separately so that it can be used to analyze the robustness of the recognition system in details and compare different vision systems.

2. **A generic model has been proposed for solving the deep and layered object recognition problem, with a novel structured online learning algorithm (*SOnline*) for efficient computation.** Based on the latest progress in machine learning on structured prediction, a generic structured prediction model

has been built for DLR, along with the analysis of several possible loss functions and feature representation strategies. The efficient SOnline algorithm makes it possible to apply DLR on the concrete deep and layered object recognition task in the IAIR-CarPed dataset. Comparative results show the superiority of the model on both deep and layered object classification and object detection compared to traditional multiclass recognition and binary recognition models. The experimental results demonstrate that DLR not only generates rich and adaptive outputs, but also improves the performance on traditional object categorization. Therefore, it provides a new idea for the research on object recognition. Potentially, the SOnline algorithm can be used for many other structured prediction problems with large amounts of training data, especially those online learning and prediction applications.

3. **A new approach named *saliency based opportunistic search* is proposed for robust object parsing, which can effectively fuse bottom-up grouping and top-down matching.** Such a heuristic search approach optimizes the grouping loss and the matching loss simultaneously. It encourages more and better object part matching results while at the same time constrains their positions and saliency; therefore it can avoid false matching and explore bottom-up grouping gradually. Experiments on challenging statue faces demonstrate the robustness of the approach to partial occlusions, inner clutters and data defacement, and show that it can generate significantly better results than the currently dominant approach using much fewer exemplars.

6.2 Future Work

The research work presented in this thesis opens the door to a broad space of deep interpretation of objects and the scene behind them. We tried to make the IAIR-CarPed dataset, the DLR recognition strategy and the opportunistic search approach as generic and extensible as possible, and we have only presented some typical instantiations to demonstrate the ideas, therefore much more work can be done to further explore these ideas.

Specifically, the following four aspects could be our future research work.

1. **A unified *model* for tightly-coupled object recognition and parsing, which may also be extended to *layered scene understanding*.** Currently object recognition and object parsing are loosely coupled. In the current solution of DLR, parts are modeled together with the object, but in the learning process the parts (currently only the key part) are treated separately for simplicity which may be insufficient in a parsing perspective. In object parsing, the object recognition (model selection) is also done separately. However, recognition seems to be necessary to

parsing while parsing may be beneficial to recognition. A desired unified model will seek to optimize both of them together, either simultaneously or iteratively. An even more ambitious goal is to go further up to integrate the object and scene relationship for a whole framework modeling scene, objects and object parts together. However, it's hard to model the relationship between scene and object parts and such a relationship must be very weak. So it is better to model it as a layered scene understanding problem: at the first level, scene is recognized by itself and also the objects within it, then the parsing of some interested objects (down to the parts of them) may result in a deeper understanding of the scene. For example, face expressions represented by the facial features can reveal the atmosphere of the scene, e.g. happy. This is exactly what has been shown in Figure 1-2 for scene understanding.

2. **Different types of supervision for reducing *annotation* expense.** For a unified object recognition and parsing, or even scene understanding, a thorough annotation of the object and parts will be a very expensive work. Therefore ways to reduce the expense should be considered. Weak supervision and semi-supervision are two plausible choice. There are many learning algorithms for these two types of learning problems, which can be introduced for solving the desired problem.

3. **Different *learning* methods: discriminative? generative? or hybrid?** As the problem and model become bigger and bigger with more layers, a uniform discriminative learning method will be more and more hard to design. Instead, generative methods have the advantage of being able to represent complex and layered relationships, but they usually have weaker discriminability than discriminative ones. Maybe a hybrid method can have a better trade-off between them. Therefore how to choose the learning method is a problem worth researching.

4. **Deep and layered object recognition and parsing in *videos*.** We would like to extend the ideas to videos by making use of the temporal relationships between frames. The recognition and parsing results in nearby frames should be similar while the evolution of them along the time axis should obey some rules, which may be referred to as semantic timeline. A global optimization along such a timeline will be very promising, which can also go up to a higher level understanding of the activities when the interested objects are humans.

REFERENCE

[1] D. Marr. Vision: a computational investigation into the human representation and processing of visual information[M]. W. H. Freeman, San Francisco, 1982

[2] Joseph Mundy. Object Recognition in the Geometric Era: A Retrospective. J. Ponce, M. Hebert, C. Schmid, Zisserman A., (Editors) Toward Category-Level Object Recognition, 2006. 3–28

[3] M. Riesenhuber, T. Poggio. Hierarchical Models of Object Recognition in Cortex[J]. Nat Neurosci. 1999, **2**(11):1019–1025

[4] T. Serre, L. Wolf, S.M. Bileschi, M. Riesenhuber, T. Poggio. Robust Object Recognition with Cortex-Like Mechanisms[J]. IEEE Trans Pattern Anal Mach Intell. March 2007, **29**(3):411–426

[5] T. Serre, A. Oliva, T. Poggio. A Feedforward Architecture Accounts for Rapid Categorization[J]. Proceedings of the National Academy of Sciences (PNAS). 2007, **104**(15):6424–6429

[6] G. H. BakIr, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, S. V.N. Vishwanathan. Predicting Structured Data[M]. Advances in neural information processing systems, Cambridge, MA, USA: MIT Press, 2007

[7] M.B. Blaschko, C.H. Lampert. Learning to Localize Objects with Structured Output Regression[C]. In: Proceedings of European Conference on Computer Vision. 2008, I: 2–15

[8] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. Int J Comput Vis. November 2004, **60**(2):91–110

[9] N. Dalal, B. Triggs. Histograms of oriented gradients for human Detection[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2005, 886–893

[10] Serge Belongie, Jitendra Malik, Jan Puzicha. Shape Matching and Object Recognition Using Shape Contexts.[J]. IEEE Trans Pattern Anal Mach Intell. 2002

[11] Antonio Torralba, Rob Fergus, William T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition[J]. IEEE Trans Pattern Anal Mach Intell. 2008, **30**:1958–1970

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009

[13] Gabriel J. Brostow, Julien Fauqueur, Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database[J]. Pattern Recognition Letters. 2009, **30**(2):88 – 97

[14] P. Dollar, C. Wojek, B. Schiele, P. Perona. Pedestrian detection: A benchmark[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009, 304–311

[15] Ashutosh Saxena, Justin Driemeyer, Andrew Y. Ng. Robotic Grasping of Novel Objects using Vision[J]. International Journal of Robotics Research. Feb 2008, **27**(2):157–173

[16] S. Se, D.G. Lowe, J.J. Little. Vision-based global localization and mapping for mobile robots[J]. IEEE Trans Robot. 2005, **21**(3):364–375

[17] Al Mansur, Yoshinori Kuno. Specific and Class Object Recognition for Service Robots through Autonomous and Interactive Methods[J]. IEICE - Trans Inf Syst. 2008, **E91-D**(6):1793–1803

[18] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld. Face Recognition: A Literature Survey[J]. ACM Computing Surveys. December 2003, **35**(4):399–458

[19] Manuel Möller, Michael Sintek, Paul Buitelaar, Saikat Mukherjee, Xiang Sean Zhou, Joerg Freund. Medical Image Understanding Through the Integration of Cross-Modal Object Recognition with Formal Domain Knowledge[C]. Proceedings of the First International Conference on Health Informatics. 2008, 134–141

[20] Wei Hong, Bogdan Georgescu, Xiang Sean Zhou, Sriram Krishnan, Dorin Comaniciu. Database-guided simultaneous multi-slice 3D segmentation for volumetric data[C]. In: Proceedings of European Conference on Computer Vision. Springer-Verlag, 2006, 397–409

[21] Zhuowen Tu, Xiang Sean Zhou, Adrian Barbu, Luca Bogoni, Dorin Comaniciu. Probabilistic 3D polyp detection in CT images: The role of sample alignment[C]. In Proc. Conf. Computer Vision and Pattern Recognition, volume II. 2006, 1544–1551

[22] D. Comaniciu, X.S. Zhou, S. Krishnan. Robust Real-Time Myocardial Border Tracking for Echocardiography: An Information Fusion Approach[J]. IEEE Trans Med Imaging. July 2004, **23**(7):849–860

[23] Thorsten Joachims, Thomas Finley, Chun Nam John Yu. Cutting-plane training of structural SVMs[J]. Machine Learning. October 2009, **77**(1):27–59

[24] Antoine Bordes, Léon Bottou, Patrick Gallinari, Jason Weston. Solving multiclass support vector machines with LaRank[C]. 2007

[25] L. G. Roberts. Machine perception of three-dimensional solids[C]. J. Tippett, D. Berkowitz, L. Clapp, C. Koester, A. Vanderburgh, (Editors) Optical and Electro-optical Information Processing. MIT Press, 1965, 159 ‑ 197

[26] S. Dickinson. Object Categorization: Computer and Human Vision Perspectives[M], Cambridge University Press, 2009

[27] Ming-Kuei Hu. Visual Pattern Recognition by Moment Invariants[J]. IRE Trans Inform Theory. February 1962, **8**(2):179–187

[28] A. Guzman Arenas. Analysis of Curved Line Drawings Using Context and Global Information[J]. Machine Intelligence. 1971, **VI**:325–376

[29] T.O. Binford. Visual Perception by Computer[C]. IEEE Conf. Systems and Controls. 1971

[30] G.J. Agin, T.O. Binford. Computer Description of Curved Objects[J]. IEEE Trans Comput. April 1976, **25**(4):439–449

[31] R. Nevatia, T.O. Binford. Structured Description of Complex Objects[C]. Proc. 3rd International Joint Conference on Artificial Intelligence. 1973, 641–647

[32] R. Nevatia, T.O. Binford. Description and Recognition of Curved Objects[J]. Artificial Intelligence Journal. February 1977, **8**(1):77–98

[33] Mourad Zerroug, Ramakant Nevatia. From an Intensity Image to 3-D Segmented Descriptions[C]. In: Proceedings of European Conference on Computer Vision. London, UK: Springer-Verlag, 1996, 11–24

[34] David G. Lowe. Perceptual Organization and Visual Recognition[M]. Norwell, MA, USA: Kluwer Academic Publishers, 1985

[35] H.J. Wolfson, Y. Lamdan. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme[C]. Proceedings of the 2nd International Conference on Computer Vision.

1988, 238–249

[36] J.L. Mundy, A.J. Heller. The Evolution and Testing of a Model-Based Object Recognition System[C]. Proceedings of the 3rd International Conference on Computer Vision. 1990, 268–282

[37] M. Kirby, L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces[J]. IEEE Trans Pattern Anal Mach Intell. 1990, **12**(1):103–108

[38] M. Turk, A. Pentland. Eigenfaces for Recognition[J]. Journal of Cognitive Neuroscience. 1991, **3**(1):71–86

[39] Hiroshi Murase, Shree K. Nayar. Visual Learning And Recognition Of 3-D Objects From Appearance[J]. Int J Comput Vis. January 1995, **14**(1):5–24

[40] Sameer Nene, Shree K. Nayar, Hiroshi Murase. Columbia Object Image Library (COIL-100)[R]. Tech. Rep. CUCS-006-96, Columbia University, 1996

[41] Ales Leonardis, Horst Bischof. Dealing with occlusions in the eigenspace approach[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. Los Alamitos, CA, USA: IEEE Computer Society, 1996

[42] Kohtaro Ohba, Katsushi Ikeuchi. Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects[J]. IEEE Trans Pattern Anal Mach Intell. 1997, **19**:1043–1048

[43] Baback Moghaddam, Alex Pentland. Probabilistic Visual Learning for Object Representation[J]. IEEE Trans Pattern Anal Mach Intell. 1997, **19**:696–710

[44] O.I. Camps, C.-Y. Huang, T. Kanungo. Hierarchical Organization of Appearance-Based Parts and Relations for Object Recognition[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. Los Alamitos, CA, USA: IEEE Computer Society, 1998

[45] M.J. Black, A.D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation[J]. Int J Comput Vis. January 1998, **26**(1):63–84

[46] Tinne Tuytelaars, Krystian Mikolajczyk. Local invariant feature detectors: a survey[J]. Found Trends Comput Graph Vis. 2008, **3**(3):177–280

[47] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2001

[48] B. Wu, R. Nevatia. Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007, 1–8

[49] K. Mikolajczyk, B. Leibe, B. Schiele. Multiple Object Class Detection with a Generative Model[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006

[50] Krystian Mikolajczyk, Andrew Zisserman, Cordelia Schmid. Shape recognition with edge-based features[C]. British Machine Vision Conference. 2003, vol. 2, 779–788. URL http://lear.inrialpes.fr/pubs/2003/MZS03

[51] Lei Yang, Nanning Zheng, Jie Yang, Mei Chen, Hong Cheng. A Biased Sampling Strategy for Object Categorization[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[52] C. Harris, M.J. Stephens. A Combined Corner and Edge Detector[C]. Alvey Vision Confer-

ence. 1988, 147–152

[53] P. R. Beaudet. Rotationally invariant image operators[C]. Proceedings of the International Joint Conference on Pattern Recognition. 1978, 579 - 583

[54] S.M. Smith, J.M. Brady. Susan: A New Approach to Low-Level Image-Processing[J]. Int J Comput Vis. May 1997, **23**(1):45–78

[55] Krystian Mikolajczyk, Cordelia Schmid. Scale and affine invariant interest point detectors[J]. Int J Comput Vis. 2004, **60**(1):63–86. URL http://lear.inrialpes.fr/pubs/2004/MS04

[56] Krystian Mikolajczyk, Cordelia Schmid. An affine invariant interest point detector[C]. In: Proceedings of European Conference on Computer Vision. Springer, 2002, 128–142

[57] J.L. Crowley, A.C. Parker. A Representation for Shape Based on Peaks and Ridges in the Difference of Low-Pass Transform[J]. IEEE Trans Pattern Anal Mach Intell. March 1984, **6**(2):156–169

[58] H. Bay, A. Ess, T. Tuytelaars, L.J. Van Gool. Speeded-Up Robust Features (SURF)[J]. International Journal on Computer Vision and Image Understanding. June 2008, **110**(3):346–359

[59] T. Kadir, A. Zisserman, M. Brady. An Affine Invariant Salient Region Detector[C]. In: Proceedings of European Conference on Computer Vision. 2004, I: 228–241

[60] J. Matas, O. Chum, M. Urban, T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions[J]. Image and Vision Computing. September 2004, **22**(10):761–767

[61] X.F. Ren, J. Malik. Learning a classification model for segmentation[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2003, 10–17

[62] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool. A comparison of affine region detectors[J]. Int J Comput Vis. 2005, **65**(1):43–72

[63] Andrew Johnson, Martial Hebert. Surface Registration by Matching Oriented Points[C]. International Conference on Recent Advances in 3-D Digital Imaging and Modeling. 1997, 121–128

[64] Koen E. A. van de Sande, Theo Gevers, Cees G. M. Snoek. Evaluating Color Descriptors for Object and Scene Recognition[J]. IEEE Trans Pattern Anal Mach Intell. 2010

[65] Krystian Mikolajczyk, Cordelia Schmid. A Performance Evaluation of Local Descriptors[J]. IEEE Trans Pattern Anal Mach Intell. 2005, **27**:1615–1630

[66] D. Gabor. Theory of Communication[J]. Journal of the Institute for Electrical Engineers. 1946, **93**(3):429–459

[67] L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, M.A. Viergever. General Intensity Transformations and Differential Invariants[J]. Journal of Mathematical Imaging and Vision. 1994, **4**:171–187

[68] W.T. Freeman, E.H. Adelson. The Design and Use of Steerable Filters[J]. IEEE Trans Pattern Anal Mach Intell. 1991, **13**:891–906

[69] F. Schaffalitzky, A. Zisserman. Multi-view Matching for Unordered Image Sets[C]. In: Proceedings of European Conference on Computer Vision. 2002, I: 414–431

[70] L.J. Van Gool, T. Moons, D. Ungureanu. Affine/Photometric Invariants for Planar Intensity Patterns[C]. In: Proceedings of European Conference on Computer Vision. 1996, I:642–651

[71] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge[J]. Int J Comput Vis. 2010, **88**(2):303–338

[72] Krystian Mikolajczyk, Bastian Leibe, Bernt Schiele. Local Features for Object Class Recog-

nition[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2005, 1792–1799

[73] M. Stark, B. Schiele. How Good are Local Features for Classes of Geometric Objects[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007

[74] Tong Lin, Hongbin Zha. Riemannian Manifold Learning[J]. IEEE Trans Pattern Anal Mach Intell. 2008, **30**:796–809

[75] Antonio Torralba, Kevin P. Murphy, William T. Freeman, Mark A. Rubin. Context-based vision system for place and object recognition[J]. In: Proceedings of IEEE International Conference on Computer Vision. 2003:273–280

[76] Yan Ke, Rahul Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. Los Alamitos, CA, USA: IEEE Computer Society, 2004, 506–513

[77] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, Deva Ramanan. Object Detection with Discriminatively Trained Part Based Models[J]. IEEE Trans Pattern Anal Mach Intell. 2009, **99**(RapidPosts)

[78] P. Sinha, B. Balas, Y. Ostrovsky, R. Russell. Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About[J]. Proc IEEE. 2006, **94**(11):1948–1962

[79] Yong Gao, Yangsheng Wang, Xuetao Feng, Xiaoxu Zhou. Face Recognition Using Most Discriminative Local and Global Features[J]. In: Proceedings of International Conference on Pattern Recognition. 2006, **1**:351–354

[80] Jian Huang, Pong C. Yuen, J. H. Lai, Chun-hung Li. Face recognition using local and global features[J]. EURASIP J Appl Signal Process. 2004, **2004**:530–541

[81] Xilin Chen, Shiguang Shan, Yu Su, Wenchao Zhang, Baochang Zhang, Wen Gao. Combining Local and Global Features for Face Recognition[J]. Technical report of IEICE PRMU. 2007-08-27, **107**(206):111–118. URL http://ci.nii.ac.jp/naid/110006423301/en/

[82] K. Murphy, A. Torralba, D. Eaton, W. Freeman. Object detection and localization using local and global features[J]. Towards Category-Level Object Recognition. 2005, **1**

[83] Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, Mark C. Benfield, Erik G. Learned-Miller. Combining Local and Global Image Features for Object Class Recognition[C]. Proceedings of the IEEE Workshop on Learning in Computer Vision and Pattern Recognition. 2005

[84] Josef Sivic, Bryan Russell, Alexei A. Efros, Andrew Zisserman, Bill Freeman. Discovering Objects and Their Location in Images[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2005

[85] S. Savarese, J. Winn, A. Criminisi. Discriminative Object Class Models of Appearance and Shape by Correlatons[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006, 2033 – 2040

[86] E.B. Sudderth, A. Torralba, W.T. Freeman, A.S. Willsky. Describing Visual Scenes Using Transformed Objects and Parts[J]. Int J Comput Vis. May 2008, **77**(1-3):291–330

[87] S. Lazebnik, C. Schmid, J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006, 2169–2178

[88] M.A. Fischler, R.A. Elschlager. The Representation and Matching of Pictorial Structures[J]. IEEE Trans Comput. January 1973, **22**(1):67–92

[89] G. Carneiro, D.G. Lowe. Sparse Flexible Models of Local Features[C]. In: Proceedings of European Conference on Computer Vision. 2006, III: 29–43

[90] Markus Weber, Max Welling, Pietro Perona. Unsupervised Learning of Models for Recognition[C]. In: Proceedings of European Conference on Computer Vision. London, UK: Springer-Verlag, 2000, 18–32

[91] R. Fergus, P. Perona, A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2003, 264–271

[92] B. Leibe, A. Leonardis, B. Schiele. Robust Object Detection with Interleaved Categorization and Segmentation[J]. Int J Comput Vis. May 2008, **77**(1-3):259–289

[93] R. Fergus, P. Perona, A. Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2005, 380–397

[94] David Crandall, Pedro Felzenszwalb, Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2005, 10–17

[95] Pedro F. Felzenszwalb, Daniel P. Huttenlocher. Pictorial Structures for Object Recognition[J]. Int J Comput Vis. 2005, **61**(1):55–79

[96] Guillaume Bouchard, Bill Triggs. Hierarchical Part-Based Visual Object Categorization[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. Washington, DC, USA: IEEE Computer Society, 2005, 710–715

[97] Y. Jin, S. Geman. Context and Hierarchy in a Probabilistic Image Model[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006, 2145–2152

[98] Long Zhu, Alan Yuille. A Hierarchical Compositional System for Rapid Object Detection. Y. Weiss, B. Schölkopf, J. Platt, (Editors) In: Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, 2006. 1633–1640

[99] L. Wolf, S. Bileschi, E. Meyers. Perception Strategies in Hierarchical Vision Systems[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006

[100] J. Mutch, D. G. Lowe. Multiclass Object Recognition with Sparse, Localized Features[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006

[101] Y. Wu, N.N. Zheng, Q.B. You, S.Y. Du. Object Recognition by Learning Informative, Biologically Inspired Visual Features[C]. In: Proceedings of IEEE International Conference on Image Processing. 2007, I: 181–184

[102] 马颂德，张正友. 计算机视觉[M]. 北京科学出版社, 1998

[103] A. Torralba. Contextual Priming for Object Detection[J]. Int J Comput Vis. 2003, **53**(2):169–191

[104] D. Hoiem, A.A. Efros, M. Hebert. Recovering Surface Layout from an Image[J]. Int J Comput Vis. October 2007, **75**(1):151–172

[105] G. Heitz, D. Koller. Learning Spatial Context: Using Stuff to Find Things[C]. In: Proceedings of European Conference on Computer Vision. 2008

[106] Chaitanya Desai, Deva Ramanan, Charless Fowlkes. Discriminative Models for Multi-class Object Layout[C]. In: Proceedings of IEEE International Conference on Computer Vision.

2009

[107] S. Palmer. Vision Science: Photons to Phenomenology[M]. MIT Press, 1999

[108] David Navon. Forest before trees: The precedence of global features in visual perception[J]. Cognitive Psychology. 1977, **9**(3):353 – 383

[109] Lin Chen. Topological structure in visual perception[J]. Science. 1982, **218**(4573):699–700

[110] Timothy J. Buschman, Earl K. Miller. Top-Down Versus Bottom-Up Control of Attention in the Prefrontal and Posterior Parietal Cortices[J]. Science. 2007, **315**(5820):1860–1862

[111] Thomas Serre Sharat Chikkerur, Tomaso Poggio. A Bayesian inference theory of attention: neuroscience and algorithms[R]. Tech. Rep. MIT-CSAIL-TR-2009-047/CBCL-280, Center for Biological and Computational Learning, Massachusetts Institute of Technology, October 2009

[112] Stella X. Yu, Ralph Gross, Jianbo Shi. Concurrent Object Recognition and Segmentation by Graph Partitioning[C]. In: Advances in Neural Information Processing Systems. MIT Press, 2002, 1383–1390

[113] M. P. Kumar, P. H. S. Torr, A. Zisserman. OBJ CUT[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2005, 18–25

[114] T. Yeh, T. Darrell. Fast concurrent object localization and recognition[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009

[115] Hedi Harzallah, Frédéric Jurie, Cordelia Schmid. Combining efficient object localization and image classification[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[116] Minh Hoai Nguyen, Lorenzo Torresani, Fernando de la Torre, Carsten Rother. Weakly supervised discriminative localization and classification: a joint learning process[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[117] Catalin Ionescu, Liefeng Bo, Cristian Sminchisescu. Structural SVM for Visual Localization and Continuous State Estimation[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[118] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, Song-Chun Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition[J]. Int J Comput Vis. 2005, **63**(2):113–140

[119] L.J. Li, R. Socher, L. Fei Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009, 2036–2043

[120] Xiong Yang, Tianfu Wu, Song-Chun Zhu. Evaluating Information Contributions of Bottom-up and Top-down Processes[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[121] Christiane Fellbaum. WordNet: An Electronic Lexical Database (Language, Speech, and Communication)[M]. The MIT Press, 1998

[122] Luis von Ahn, Laura Dabbish. Labeling images with a computer game[C]. CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems. New York, NY, USA: ACM Press, 2004, 319–326

[123] Henry Schneiderman, Takeo Kanade. A Statistical Model for 3D Object Detection Applied to Faces and Cars[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2000

[124] Shivani Agarwal, Aatif Awan, Dan Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation[J]. IEEE Trans Pattern Anal Mach Intell. 2004, **26**:1475–1490

[125] C.P. Papageorgiou, T. Poggio. A Trainable System for Object Detection[J]. Int J Comput Vis. June 2000, **38**(1):15–33

[126] N. Dalal, B. Triggs. Histograms of Oriented Gradients for Human Detection[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2005, 886–893

[127] B. Wu, R. Nevatia. Detection and Segmentation of Multiple, Partially Occluded Objects by Grouping, Merging, Assigning Part Detection Responses[J]. Int J Comput Vis. April 2009, **82**(2):185–204

[128] S. Munder, D.M. Gavrila. An Experimental Study on Pedestrian Classification[J]. IEEE Trans Pattern Anal Mach Intell. November 2006, **28**(11):1863–1868

[129] M. Enzweiler, D.M. Gavrila. Monocular Pedestrian Detection: Survey and Experiments[J]. IEEE Trans Pattern Anal Mach Intell. 2009, **31**(12):2179 –2195

[130] A. Ess, B. Leibe, L.J. van Gool. Depth and Appearance for Mobile Scene Analysis[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007, 1–8

[131] R. Fergus, P. Perona, A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2003

[132] Li Fei-Fei, Rob Fergus, Pietro Perona. One-Shot Learning of Object Categories[J]. IEEE Trans Pattern Anal Mach Intell. 2006, **28**:594–611

[133] G. Griffin, A. Holub, P. Perona. Caltech-256 object category dataset[R]. Technical Report 7694, CalTech, 2007

[134] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, William T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation[J]. Int J Comput Vis. May 2008, **77**:157 - 173

[135] Stan Bileschi. StreetScenes: Towards Scene Understanding in Still Images[D]. Ph.D. thesis, Massachusetts Institute of Technology, 2006

[136] Jamie Shotton, John Winn, Carsten Rother, Antonio Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context[J]. Int J Comput Vis. January 2009, **81**(1):2–23

[137] B. Yao, X. Yang, S.C. Zhu. Introduction to a Large-Scale General Purpose Ground Truth Database: Methodology, Annotation Tool and Benchmarks[C]. EMMCVPR07. 2007, 169–183

[138] C.H. Lampert, H. Nickisch, S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009, 951–958

[139] A. Farhadi, I. Endres, D. Hoiem, D.A. Forsyth. Describing objects by their attributes[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2009, 1778–1785

[140] Q. Yuan, A. Thangali, V. Ablavsky, S. Sclaroff. Multiplicative kernels: Object detection, segmentation and pose estimation[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2008

[141] Jania Aghajanian, Jonathan Warrell, Simon J.D. Prince, Peng Li, Jennifer L. Rohn, Buzz Baum. Patch-based Within-Object Classification[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009, 1125–1132

[142] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, Patrick J. Rauss. The FERET

Evaluation Methodology for Face-Recognition Algorithms[J]. IEEE Trans Pattern Anal Mach Intell. 2000, **22**:1090–1104

[143] Gary B. Huang, Manu Ramesh, Tamara Berg, Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments[R]. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007

[144] A. Ferencz, E.G. Learned Miller, J. Malik. Building a Classification Cascade for Visual Identification from One Example[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2005, 286–293

[145] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, E. Kayafas. License Plate Recognition From Still Images and Video Sequences: A Survey[J]. September 2008, **9**(3):377–391

[146] L. Fei-Fei, R. Fergus, Pietro Perona. Learning Generative Visual Models From Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories[C]. 2004

[147] A. Opelt, A. Pinz, A. Zisserman. Incremental learning of object detectors using a visual shape alphabet[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2006, 3–10

[148] C.H. Lampert, M.B. Blaschko, T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2008, 1–8

[149] B. Kulis, K. Grauman. Kernelized locality-sensitive hashing for scalable image search[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009, 2130–2137

[150] A. Torralba, K.P. Murphy, W.T. Freeman. Sharing Visual Features for Multiclass and Multiview Object Detection[J]. IEEE Trans Pattern Anal Mach Intell. May 2007, **29**(5):854–869

[151] L. Lin, S.W. Peng, J. Porway, S.C. Zhu, Y.T. Wang. An Empirical Study of Object Category Recognition: Sequential Testing with Generalized Samples[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007, 1–8

[152] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, T. Poggio. A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex[R]. Tech. Rep. AI Memo 2005-036/CBCL Memo 259, Massachusetts Inst. of Technology, 2005

[153] Andrea Vedaldi, Varun Gulshan, Manik Varma, Andrew Zisserman. Multiple Kernels for Object Detection[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[154] Boris Babenko, Steve Branson, Serge Belongie. Similarity Metrics for Categorization: From Monolithic to Category Specific[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[155] G. Griffin, P. Perona. Learning and using taxonomies for fast visual categorization[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2008, 1–8

[156] Matthieu Guillaumin, Jakob Verbeek, Cordelia Schmid. Metric Learning Approaches for Face Identification[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[157] A. Ferencz, E.G. Learned Miller, J. Malik. Building a Classification Cascade for Visual Identification from One Example[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2005, I: 286–293

[158] Lubomir Bourdev, Jitendra Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2009

[159] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results[Z]. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2009

[160] M. Marszalek, C. Schmid. Semantic Hierarchies for Visual Object Recognition[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2007, 1–7

[161] Alexander Binder, Motoaki Kawanabe, Ulf Brefeld. Efficient Classification of Images with Taxonomies[C]. In: Proceedings of Asian Conference on Computer Vision. 2009

[162] Alon Zweig, Daphna Weinshall. Exploiting Object Hierarchy: Combining Models from Different Category Levels[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007, 1–8

[163] Ashish Kapoor, Raquel Urtasun, Trevor Darrell. Probabilistic Kernel Combination for Hierarchical Object Categorization[J]. TR UCB/EECS. 2009

[164] Yuanhao Chen, Long Zhu, Chenxi Lin, Alan Yuille, Hongjiang Zhang. Rapid Inference on a Novel AND/OR graph for Object Detection, Segmentation and Parsing[C]. In: Advances in Neural Information Processing Systems. 2007

[165] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables[J]. J Mach Learn Res. 2005, **6**:1453–1484

[166] Yang Wu, Zejian Yuan, Yuanliu Liu, Nanning Zheng. Discriminative Structured Outputs Prediction Model and Its Efficient Online Learning Algorithm[C]. IEEE International Workshop on Emergent Issues in Large Amounts of Visual Data. 2009, 2087–2094

[167] Christoph H. Lampert, Matthew B. Blaschko. Structured prediction by joint kernel support estimation[J]. Machine Learning. December 2009, **77**(2-3):249–269

[168] Manik Varma, Debajyoti Ray. Learning The Discriminative Power-Invariance Trade-Off[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007, 1–8

[169] Thorsten Joachims, Thomas Hofmann, Yisong Yue, Chun-Nam Yu. Predicting Structured Objects with Support Vector Machines[J]. Communications of the ACM. 2009, **52**(11):97–104

[170] P.F. Felzenszwalb, D. McAllester, D. Ramanan. A discriminatively trained, multiscale, deformable part model[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2008

[171] Juho Rousu, Craig Saunders, Sandor Szedmak, John Shawe-Taylor. Kernel-based Learning of Hierarchical Multilabel Classification Models[J]. Journal of Machine Learning Research. 2006, **7**:1601–1626

[172] C. Garcia, G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis[J]. IEEE Trans Multimedia. Sep 1999, **1**(3):264–277

[173] Huaifeng Zhang, Wenjing Jia, Xiangjian He, Qiang Wu. Learning-Based License Plate Detection Using Global and Local Features[C]. ICPR. 2006, II: 1102–1105

[174] Martin Szummer, Pushmeet Kohli, Derek Hoiem. Learning CRFs using Graph Cuts[C]. In: Proceedings of European Conference on Computer Vision. 2008, 1–14

[175] B. Wu, R. Nevatia. Detection and Segmentation of Multiple, Partially Occluded Objects by

Grouping, Merging, Assigning Part Detection Responses[J]. Int J Comput Vis. April 2009, **82**(2):185–204

[176] Christian Wojek, Gyuri Dorkó, André Schulz, Bernt Schiele. Sliding-Windows for Rapid Object Class Localization: A Parallel Technique[C]. Proceedings of the 30th DAGM symposium on Pattern Recognition. Berlin, Heidelberg: Springer-Verlag, 2008, 71–81

[177] P. Felzenszwalb, D. McAllester, D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model.[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2008

[178] V. Ferrari, F. Jurie, C. Schmid. Accurate Object Detection with Deformable Shape Models Learnt from Images[C]. In: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition. 2007, 1–8

[179] Qihui Zhu, Jianbo Shi. Untangling Cycles for Contour Grouping[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2007

[180] Qihui Zhu, Liming Wang, Yang Wu, Jianbo Shi. Contour Context Selection for Object Detection: A Set-to-Set Contour Matching Approach[C]. In: Proceedings of European Conference on Computer Vision. 2008

[181] I. Biederman. Recognition by Components: A Theory of Human Image Understanding[J]. PsychR. 1987, **94**(2):115–147

[182] A.P. Pentland. Recognition by Parts[C]. In: Proceedings of IEEE International Conference on Computer Vision. 1987, 612–620

[183] Y. Amit, A. Trouve. POP: Patchwork of Parts Models for Object Recognition[J]. Int J Comput Vis. November 2007, **75**(2):267–282

[184] E.B. Sudderth, A. Torralba, W.T. Freeman, A.S. Willsky. Learning Hierarchical Models of Scenes, Objects, and Parts[C]. In: Proceedings of IEEE International Conference on Computer Vision. 2005, II: 1331–1338

[185] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham. Active Shape Models: Their Training and Application[J]. Comput Vis Image Underst. January 1995, **61**(1):38–59

[186] T.F. Cootes, G.J. Edwards, C.J. Taylor. Active Appearance Models[J]. IEEE Trans Pattern Anal Mach Intell. June 2001, **23**(6):681–685

[187] Liming Wang, Jianbo Shi, Gang Song, I fan Shen. Object Detection Combining Recognition and Segmentation[C]. In: Proceedings of Asian Conference on Computer Vision. 2007, 189–199

[188] Ching Chen. The First Emperor of China[M]. Voyager Company, 1994

ACKNOWLEDGEMENT

I would like to give my most sincere thanks to my advisors Professor Nanning Zheng and Professor Jianbo Shi, and my supervisor Professor Zejian Yuan for their guidance, help and support. Professor Nanning Zheng has led me to the exciting research field of computer vision. His great vision, religious attitude, endless enthusiasm and peaceful mind have influenced me a lot, and will continue to shape me in the future. Professor Jianbo Shi was my advisor when I was visiting the University of Pennsylvania, and he continues to be one of my best teachers and friends. I have learnt much from him about how to do good research and how to be a professional researcher. Professor Zejian Yuan is a great supervisor and friend. His erudition, patience and help without reservation make me always feel supported.

I also thank my thesis committee members and thesis reviewers Professor Rongchun Zhao, Jingmin Xin, Jianru Xue, Licheng Jiao, Jiuqiang Han and Qiguang Wang not only for their constructive comments and advice on the thesis, but also as great role models on how to be a successful researcher. Moreover, I would like to thank Professor Yuehu Liu for his encouragement whenever I face a great challenge. I sincerely thank all the other professors and teachers in the Institute of Artificial Intelligence and Robotics for all their help and care, which make me always feel being at home.

Over the past six years it has been my great honor to work with my older academic brothers Qubo You, Shaoyi Du, Xuetao Zhang, Jianyi Liu, Lei Xiong, Xiyan He, Lei Cao, Yanyun Qu, Gaofeng Meng, my colleagues and friends Dr. Liming Wang from Fudan University, Dr. Qihui Zhu, Praveen Srinivasan, Katerina Fragkiadaki, Jack Sim and Jeffrey Byrne from the University of Pennsylvania, and my dear younger academic brothers Yuanliu Liu, Geng Zhang, Zheng Ma, Huaizu Jiang, Wei Liu, Hong Ji, Jihua Zhu, Xiaowei Zhang, Dapeng Chen, Yudong Liang, Xiongdong Sheng, Cao Cui, Zigang Li, Youliang Chen, Yongli Liu, Bo Chen, Jingjun Wu, and Jun Wen. It was a great time to be with them.

I thank my wife Ying Wang for her unconditional love and unending patience, understanding and support. She is the most important fortune in my life. I also thank my parents-in-law very much for their trust, care and encouragement, which have helped me over the hump. I owe everything to my parents for their dedication, hard work and support, and to my younger sister Yaqin who had sacrificed her chance to continue her education in her teenage due to the poor old days of my family. Finally, I thank all my relatives and friends, near and far, for their support and care.

Without anyone of all the people mentioned above, I cannot come this far.

ACHIEVEMENTS

Academic Papers:

[1] Wu Y, Liu YL, Yuan ZJ, and Zheng NN. Human Confusion Costs for Object Classification [J], Optical Engineering, 2011, 50(2) (SCI Journal).

[2] Wu Y, Zheng NN, Yuan ZJ, Jiang HZ, and Tie Liu. Detection of salient objects with focused attention based on spatial and temporal coherence [J], Chinese Science Bulletin (accepted).

[3] Wu Y, Yuan ZJ, Liu YL, and Zheng NN. Discriminative Structured Outputs Prediction Model and Its Efficient Online Learning Algorithm[C]. IEEE International Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD). In Proceedings of IEEE 12th International Conference on Computer Vision (ICCV) Workshops. 2009:2087-2094.

[4] Wu Y, Zhu QH, Shi JB, and Zheng NN. Saliency Based Opportunistic Search for Object Part Extraction and Labeling[C]. In Proceedings of European Conference on Computer Vision (ECCV), 2008, 4: 760-774 (EI: 20084911762116).

[5] Wu Y, Zheng NN, You QB, and Du SY. Object Recognition by Learning Informative, Biologically Inspired Visual Features[C]. In Proceedings of IEEE International Conference on Image Processing (ICIP), 2007:181-184 (EI: 20083111425825).

[6] Zhu QH, Wang LM, Wu Y, and Shi JB. Contour Context Selection for Object Detection: A Set-to-Set Contour Matching Approach, Proceedings of European Conference on Computer Vision (ECCV), 2008, 2:774-787(EI: 20084911762015).

[7] You QB, Zheng NN, Gao L, Du SY, and Wu Y. Analysis of Solution for Supervised Graph Embedding [J], International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), 2008, 22(7): 1283-1299(SCI: 396NM ; EI: 20090511879913).

[8] You QB, Zheng NN, Du SY, and Wu Y, Neighborhood Discriminant Projection for Face Recognition[J], Pattern Recognition Letters (PRL), 2007, 28(10):1156-1163 (SCI:178VX; EI: 20071910596389).

[9] Du SY, Zheng NN, Ying SH, You QB, and Wu Y. An extension of the ICP algorithm considering scale factor[C]. In Proceedings of IEEE International Conference on Image Processing (ICIP), 2007:193-196 (EI: 20083111426388).

[10] Du SY, Zheng NN, You QB, Wu Y, Yuan MJ, and Wu JJ. Rotated haar-like features for face detection with in-plane rotation[C]. In Proceedings of 12th International Conference on Virtual Systems and Multimedia (VSMM), 2006:128-137 (SCI: BFG65, EI: 20064810275651).

[11] You QB, Zheng NN, Du SY, and Wu Y. General Solution for Supervised Graph Embedding[C]. In Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2007:782-789 (EI: 20080311039349).

[12] You QB, Zheng NN, Du SY, and Wu Y. Neighborhood Discriminant Projection for Face Recognition[C]. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR), 2006(2):532-535 (EI: 20071510540736).

[13] Wu Y, Zheng NN, Liu YL, and Yuan ZJ. Deep and Layered Object Recognition [J]. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) (under review).

[14] Wu Y, Liu YL, Yuan ZJ, and Zheng NN. A Psychophysically Annotated Dataset with Deep and Layered Semantics for Object Recognition [J], Pattern Recognition Letters (under review).

Projects Participated:

[1] 基于视觉信息的环境感知与目标识别关键技术, 国家973课题(项目编号: 2007CB311005), 2007-2011.

[2] 基于视觉注意的显著性目标检测与跟踪方法研究, 国家自然科学基金(项目编号: 90820017), 2008-2012.

[3] 视觉认知的环境感知与推理学习, 国家973前期预研重大项目(项目编号: 2006CB708303), 2006-2008.

[4] 基于视觉感知与认知机理的图像分析与识别系统, 国家863重大项目(项目编号: 2006AA01Z192), 2006-2008.

[5] 面向公共安全保障的网络化智能人像处理与认证系统, 国家863重大项目(项目编号: 2005AA147060), 2005-2006. (总评结果为优)

Software Written:

[1] 登记号: 2006SR10176. 人脸图像拼接系统软件 v1.0. 首发日期2006.04.01. 公布日期2006.07.31.

[2] 登记号: 2006SR10883. 人脸图像元素入库系统软件 v1.0. 首发日期2006.04.01. 公布日期2006.08.11.