Object Part Localization Using Exemplar-based Models

Jiongxin Liu

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Object Part Localization Using Exemplar-based Models

Jiongxin Liu

Object part localization is a fundamental problem in computer vision, which aims to let machines understand object in an image as a configuration of parts. As the visual features at parts are usually weak and misleading, spatial models are needed to constrain the part configuration, ensuring that the estimated part locations respect both image cue and shape prior. Unlike most of the state-of-the-art techniques that employ parametric spatial models, we turn to non-parametric exemplars of part configurations. The benefit is twofold: instead of assuming any parametric yet imprecise distributions on the spatial relations of parts, exemplars literally encode such relations present in the training samples; exemplars allow us to prune the search space of part configurations with high confidence.

This thesis consists of two parts: fine-grained classification and object part localization. We first verify the efficacy of parts in fine-grained classification, where we build working systems that automatically identify dog breeds, fish species, and bird species using localized parts on the object. Then we explore multiple ways to enhance exemplar-based models, such that they can be well applied to deformable objects such as bird and human body. Specifically, we propose to enforce pose and subcategory consistency in exemplar matching, thus obtaining more reliable hypotheses of configuration. We also propose part-pair representation that features novel shape composing with multiple promising hypotheses. In the end, we adapt exemplars to hierarchical representation, and design a principled formulation to predict the part configuration based on multi-scale image cues and multi-level exemplars. These efforts consistently improve the accuracy of object part localization.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Foremost, I am very grateful to my advisor Prof. Belhumeur for his continuous support of my PhD study, for his great inspiration and patient guidance on my research. By working with him on multiple research projects, I gradually acquired the skills of a capable researcher: I have learned how to spot a fundamental problem, how to develop and fit an idea in with the literature, how to verify if a method works at an early stage, and how to stress the contributions when writing a research paper. Moreover, his philosophy of "good idea is generally simple" has impacted my pattern of thinking, enabling me to stay sane when solving complicated problems. He has made me realize the importance of keeping "big picture" in mind, so that I will not be distracted by minor things in a project. He also demonstrates the quality of a talented speaker, which encourages me to continue horning my communication skills, from expressing high-level ideas to explaining technical details. I believe these lessons will greatly benefit my future career.

Besides my advisor, I am also very grateful to my cooperators on multiple projects: Prof. Jacobs, for being my unofficial advisor for the first few years, and helping me with technical implementations; Angjoo Kanazawa for working on the experiments and speed-up of DogSnap; Chun-Kang Chen, Shao-Chuan Wang, for building iOS interfaces of DogSnap; Chun-Chao Wang for collecting and labelling the fish dataset; Yinxiao Li for the help with my recent papers. You make me enjoy the benefit of teamwork: everyone can be very productive by focusing on what he is good at.

My sincere thanks also go to my second advisor Prof. Nayar. Although I didn't get direct supervision from him, I attended most of his lab meetings which brought another side of computer vision to me. I was amazed by many practical products powered by the imaging techniques developed in the lab. Perhaps, the most important question to ask when proposing a new project is "what practical usage will the project lead to". I will bear this in mind and pursue the things that make our life more convenient and wonderful. Besides, I also learned from him almost all the aspects of preparing a presentation, such as use of graphs, structure of slides, and choice of wordings.

I want to show gratitude to my colleagues in the department who accompanied me most in the past few years: Thomas Berg, for challenging and helping refine my immature ideas, and for putting up with distractions from my programming and English questions; Mohit Gupta, for the discussions of research topics, which helped me navigate the progression of PhD; Anne Flemming, for the assistance of important paperworks, and for careful proofreading of my papers; Cindy Walters, Jessica Rosa, Elias Tesfaye, Daisy Nguyen, Evelyn Guzman and other staff members of this department, for providing me with abundant resources to carry out my research work; Changyin Zhou, Brian Smith, Daniel Miau, and all the other CAVE and Vaplab members who gave me different viewpoints to what I have been pursuing, through instructive conversations about work and life; and finally Prof. Kender and Prof. Chang, for being my committee members.

I would also like to thank many other friends outside my immediate academic group: Hao Dang, Peiti Li, Wei An, Xinnan Yu, Jie Feng, Jing He, Kaili Zhang, Yan Wang, Ding Li, Chun-Yu Tsai, Jian Ma, Long Wang, Xiang Shi, and Qi Liang. It has been a great time enjoying dining, shopping, music, film, and travelling with you, which dispels my anxiety and stress, and makes my life colourful and fun. I am really lucky to have so many good friends, and will miss the days we were together.

Finally, I want to thank my parents, Zeming Liu and Guoyu Xu, for bringing me into the world, and supporting me spiritually throughout my life. I don't think I can reach current height were it not for their endless care. They show sufficient respect to what I am interested in, and encourage me to be myself. They also instilled determination and perseverance into my personality, which enabled me to pass over obstacles in the past few years. They deserve all the credit for my growth and success.

Life is a long journey, everyone who walked into my life has left precious memories. I will take them with me to embrace tomorrow. Thank you all!

This thesis is dedicated to my family and my friends.

Chapter 1

Introduction

Part is a very important concept in visual recognition. By decomposing an object into parts, we can easily describe the distinctive visual features of the object: first, the presence or absence of a part defines a basic-level object category. For instance, Car has Tire, Tree has Trunk, Bird has Wing, but Human does not have any of these parts; second, within a basic-level category such as Bird, the visual appearance at the parts capture the subtle information to differentiate its subcategories (*i.e.*, bird species). For example, the only evident difference between Kentucky Warbler and Canada Warbler lies in the Throat. Such example is related to a hot topic in computer vision: fine-grained classification, the goal of which is to distinguish between subcategories which are semantically and visually similar. Besides object categorization, parts also benefit attribute classification, such as recognizing the apparel, hair style, hand gesture on a human body. Therefore, the importance of parts motivates us to tackle the problem of object part localization. In this thesis, we target semantic parts (*e.g.*, Eye, Belly, Leg, etc.) that are well defined and can be labeled as keypoints.

1.1 Problem Statement

Given an image of object, we are interested in locating the parts on the object, which is essentially to understand how the object is situated within the image. This is a challenging problem, as (1) mapping raw image to a structured output is highly non-linear; (2) the part configuration of an object may vary widely across images due to different locations, scales, orientations (both in-plane and off-plane), and deformation of the object. In addition, the appearance variations of the object, compounded with variable background, adds to the complexity.

The feasibility and importance of part localization can be both demonstrated by fine-grained categories. On the one hand, subclasses of a basic-level category share the same set of parts and anatomic structure (for living objects), making it possible to detect the parts across the subcategories. On the other hand, visual appearance at parts provides the key to fine-grained classification as subcategories usually exhibit different shapes, colors, and textures at parts. However, as the difference can be very subtle, the classification task is even difficult to humans without proper training. Therefore, the first question we need to answer is how well machines can perform by using localized parts.

As proposed by [Hillel and Weinshall, 2006], subordinate class recognition can be solved with a two-stage algorithm: (1) identify the parts of the basic-level object; (2) differentiate subordinate classes using the features implementing the parts. This idea hasn't received enough attention in solving modern fine-grained problems where there are much larger number of subcategories. Therefore, we are among the first to instantiate the idea as a systematic approach for fine-grained classification, witch should be generalizable to different categories. In addition, we place emphasis on well-defined and detectable parts which ensure strict correspondence across different instances, facilitating both training and testing.

As the classification method is built on top of detected parts, the accuracy of part locations is critical to the overall performance. Our next focus is then on designing part localization methods that are capable of finding the parts regardless of the variations of the objects. Generally, object part localization relies on modeling the appearance and spatial relations of parts. However, previous appearance and spatial models have difficulty achieving the ideal effect due to the following reasons: (1) the local appearance at parts is inherently ambiguous and misleading, lacking sufficient cues to be distinct from negative patterns on and off the object. In addition, the same part may exhibit multi-modal appearance on different instances, thus having large intra-class variations; (2) poses[1] vary a lot across instances, making widely used spatial models such as multivariate Gaussian or tree-structured pair-wise Gaussian distributions not capable of accurately representing the space of plausible poses. To cope with the first problem, we seek to explore the pool of diverse appearance models, each of which may be limited, but performs much better than random. For the

[1] Pose means the configuration of parts, which can be either global (for all the parts) or local (for subset of parts).

second issue, we turn to non-parametric spatial models (*i.e.*, exemplars-based models [Belhumeur *et al.*, 2011]), which literally represent the pose space with a discrete set of training exemplars. However, the applicability of exemplars-based models in other domains than human faces is yet to be explored. Ideally, exemplars should impose strong constraints on the part configuration while having flexibility to adapt to a particular testing sample.

In summary, our problem statement goes as follows. *Object part localization is a fundamental and challenging problem in computer vision, which is also critical to high-level vision tasks such as fine-grained classification. However, the effect of parts has not received enough attention or study. We decide to build a generalizable working system for fine-grained classification using parts, from which we can verify if parts help, and evaluate how much performance gain we can achieve from them. As for object part localization, exemplar-based models have shown promising results on human faces, but their applicability to other domains (especially for highly deformable objects) is yet to be explored. We need to come up with methods that enhance the strength and expressiveness of exemplars by combining them with rich appearance models as well as flexible object representations.*

1.2 Our Approach

To demonstrate the effect of parts, we build fully automatic systems for fine-grained classification using part-based method (Chapter 3 and Chapter 4). The pipeline is: (1) detect the parts describable at point locations; (2) extract local features at or around the detected parts; (3) feed the features to the classification model. Our method differs from conventional image classification techniques in that it enforces strict part-level correspondences in the extracted features. In other words, a particular portion in the feature vector corresponds to a specific part. Extracting local features at or around parts also allows us to capture the subtle discriminative information that is only present in a small region on the object. Similar practice has been seen in human face recognition [Arca *et al.*, 2006; Kumar *et al.*, 2009; Yin *et al.*, 2011], but our work generalizes the part-based method to a broader domain than the well-studied human face. Specifically, we target fine-grained categories where detecting object parts with acceptable accuracy is much harder than that for human faces. Nevertheless, we manage to do so with well-designed method of part localization. After our work, several recent methods of fine-grained classification [Berg and Belhumeur, 2013; Gavves *et al.*, 2013; Xie *et al.*, 2013] also rely on parts.

Localizing the parts of an object with large appearance and pose variations calls for appropriate models. The appearance models are generally classifiers that differentiate the target part from other parts as well as the background (In the context of part localization, such classifier is called part detector). One typical implementation of the classifier is a binary Support Vector Machine (SVM) [Cortes and Vapnik, 1995] with gradient-based features such as SIFT [Lowe, 2004] and HOG [Dalal and Triggs, 2005]. Our method does not count on a single classification model as none of existing classifiers is perfect due to the problems discussed in Sec. 1.1. Instead, we aim to build a rich set of diverse models which collaboratively capture the visual appearance of a part. In Chapter 5, we build both pose and subcategory detectors for each part. The former one scores the local pose at a part regardless of the identities of the object, while the latter one tries to learn the class-specific features at the part. We push this idea to an extreme in Chapter 6, where we build detectors for each pair of parts. For an object with n parts, we have up to $n-1$ pair detectors corresponding to each part. In Chapter 7, we build much fewer detectors thanks to the power of Deep Convolutional Neural Network (DCNN), but we still have two DCNN models that capture part appearance based on part relations at different scales.

Appearance models are not the only key to solving part localization as there exist strong spatial relations between parts, entailing some spatial models to constraint the part locations. For this purpose, we employ exemplars for their simplicity and complete representation of pose variations. Exemplars just literally record the part configurations seen in the training samples, without assuming any distribution on the part configuration or part relation. Our work then is to exploit them in the pursuit of reliable and expressive spatial models. For reliability, the strategy is to combine exemplars with rich appearance models. In Chapter 5, we follow the paradigm of Consensus-of-Exemplars (CoE) approach [Belhumeur *et al.*, 2011], but change the evaluation of hypotheses (geometrically transformed exemplars) such that the highest-scoring ones are more likely to fit the testing sample in geometry. To do this, we let exemplars dictate not only the relative part locations, but also the local configurations in the neighborhood of each part (referred to as part pose). In addition, we force each exemplar to carry a unique yet unknown identity label. During testing, each hypothesis is scored based on (1) how likely the parts with the corresponding poses are at the hypothesized locations (pose consistency); and (2) how likely the parts from a particular class are at those locations (subcategory consistency). As a result of more specific evaluation, we achieve more reliable estimation of the query pose.

In Chapter 6, we further exert efforts on reliability and expressiveness of exemplar-based models. In this work, we propose part-pair representation, where an object is treated as a collection of part pairs. Therefore, the score of a hypothesis indicates how likely its part pairs are present in the image. Due to the large number of part pairs, we enjoy reliable estimation of query poses without considering the subcategory consistency as in Chapter 5. We also observe that in CoE approach, the plausible configurations are limited by the available exemplars. This is not a problem as long as the exemplars densely cover the pose space. However, the efficacy of CoE approach degrades given insufficient training samples and large articulated deformation (*e.g.*, human body). To improve the expressiveness of exemplars, we take two measures. First, we relax the pose constraints from an exemplar by focusing on a single part, where only the part pairs sharing the target part are considered. In such case, exemplars that do not strictly fit the testing sample can still be useful if the parts are within a tolerable error from the actual part locations. Second, we explicitly compose novel configurations from hypotheses, achieving potentially better matching between composed shape and the testing sample. As such, we bypass non-linear consensus which is problematic when the majority of top-scoring hypotheses are incorrect.

Finally, we propose a discriminatively trained formulation to infer part configurations in Chapter 7. The formulation contains image dependent spatial terms which combine multi-scale DCNN-based appearance models with hierarchical exemplar-based models. As we employ strong appearance models, we still achieve reliable estimation of part relations being present in the test image, thus ensuring the reliability of exemplars. The expressiveness of exemplars is boosted through hierarchical object representation. Specifically, the hierarchy is a tree structure, with each layer containing parts (tree nodes) at roughly the same level of granularity. Accordingly, we have exemplar-based models for each tree node, representing the poses of the corresponding part. Therefore, the granularity of spatial relations dictated by the exemplars also varies. Take human body as an example, the exemplars at a fine scale capture the local configurations of limbs (*e.g.*, Right arm, Left leg), while the exemplars at a coarse scale capture rough configurations of the body (*e.g.*, a standing person, an upside down person). In a nutshell, we improve the applicability of exemplar-based models by generating models at different levels of granularity and making an assumption of independence for models at different levels.

1.3 Thesis Contributions

The main contributions of this thesis are

- Two datasets of fine-grained categories with part labels: Columbia Dog Dataset and Columbia Fish Dataset (Chapter 3 and Chapter 4)
- Complete working systems that perform fine-grained classification on different categories, with iPhone App released (Chapter 3 and Chapter 4)
- A method of leveraging exemplars, which are originally identity free, to perform classification by inferring class-specific parts (Chapter 3)
- A method of enforcing pose and subcategory consistency on exemplar-based models to achieve reliable part localization (Chapter 5)
- A novel part-pair representation that enables customizable spatial constraints on the parts (Chapter 6)
- An explicit shape composing method that takes advantage of the part-pair representation, and eliminate the use of imperfect Consensus operation (Chapter 6)
- A novel discriminatively trained formulation to infer part configurations, featuring hierarchical object representation (Chapter 7)
- An efficient approximate algorithm that achieves good-enough results in searching for the optimal part configuration (Chapter 7)

1.4 Organization

Now we describe the outline of the thesis. There two topics about parts: part-based fine-grained classification and exemplar-based part localization. Before getting to the two topics, Chapter 2 reviews related works about them as well as object detection which shares some techniques with part localization, especially on the appearance models.

Our method of fine-grained classification is described in Chapter 3 and Chapter 4. Chapter 3 develops a working system for dog breed classification using part-based approach. Chapter 4 presents follow-up works that apply a simplified approach (due to upgraded part localizers) to fishes and birds.

After verifying the benefit of parts, we are then focused on improving the accuracy of part localization with enhanced exemplar-based models, including Chapter 5, Chapter 6, and Chapter 7. Chapter 5 addresses part localization specifically for fine-grained categories. It proposes to enhance the exemplars by enforcing pose and subcategory consistency on the parts. However, it still uses the original consensus module to predict the final part locations. Chapter 6 goes beyond fine-grained categories, and introduces part-pair representation to improve the applicability of exemplar-based models. Chapter 7 employs hierarchical representation which yields expressive multi-level exemplars, and learns the formulation to score part configurations in a principled way. In summary, the three chapters extensively study and improve the application of exemplars in object part localization using different methodologies.

Finally in Chapter 8, we summarize this thesis and discuss the future works that can extend our ideas and methods to further advance the techniques of object part localization.

Chapter 2

Related Work

2.1 Fine-grained Classification

Conventional image classification focuses on recognizing basic-level categories, and generally favors the paradigm of bag-of-words (or bag-of-features) approach: a set of local features are extracted at generic locations or interesting points within an image, sampling both object and background; then the local features are coded and pooled to form a vector to represent the image content; finally, the vector is fed to a classifier to predict the class label. Such paradigm has demonstrated impressive levels of performance after years of development. Early works studied the features and pooling methods, including [Wallraven and Caputo, 2003; Csurka *et al.*, 2004; Jurie and Triggs, 2005; Grauman and Darrell, 2005; Lazebnik *et al.*, 2006]. More recent works achieved further improvement by developing new coding schemes [Wang *et al.*, 2009; Zhou *et al.*, 2010; Su and Jurie, 2011] and more sophisticated classification models [Gehler and Nowozin, 2009].

Unlike basic-level image classification, fine-grained classification aims at differentiating subclasses of the same basic-level categories, such as dog breeds and bird species. As the subcategories are visually similar, fine-grained classification relies on the features that can capture subtle differences in the appearance. As a result, it is important to focus on the target object rather than the background regions (background generally contains more noise than useful contextual information about the object identity). There have been approaches that try to exclude the background: [Yao *et al.*, 2011] uses pre-cropped foreground images, and mines discriminative features on the object. [Branson *et al.*, 2010] also uses pre-cropped images and build an interactive system for bird species classification, where users answer simple questions about the shape, color, and texture of the parts to assist the classification. [Nilsback and Zisserman, 2008] segments the object first, and uses a multiple kernel framework [Vedaldi *et al.*, 2009] to combine different features extracted from the object. [Belhumeur *et al.*, 2008; Kumar *et al.*, 2012] identifies plant species using images of leaves. They also rely on image segmentation to obtain the region of interest before extracting descriptors.

Face recognition is an extreme case of fine-grained classification, where the most effective methods use features extracted from local image patches on geometrically aligned faces [Arca *et al.*, 2006; Kumar *et al.*, 2009; Yin *et al.*, 2011; Berg and Belhumeur, 2012]. Same idea can be applied to general fine-grained problems where subtle differences in appearance across subcategories usually lie in the parts, but there are additional challenges. First, localizing the parts is possible as different subcategories still share common parts. However, due to wide intra-class variations of parts, it is not easy to locate them even with decent accuracy. Second, even if the part locations are correct, for most objects other than human faces, geometric alignment such as affine transformation is not suitable. Fortunately, it turns out to be sufficient that the parts establish strict correspondence between instances, as classifiers can capture the discriminative features of parts easily.

Vision community has realized the importance of parts in fine-grained classification, and there are multiple works including ours adopting part-based approach. Especially relevant works are [Farrell *et al.*, 2011; Zhang *et al.*, 2012], which use the Poselet framework [Bourdev *et al.*, 2010] to localize the bird parts (*e.g.*, head and body), and extract feature from those parts. [Parkhi *et al.*, 2012] employs Deformable Part Models (DPMs) [Felzenszwalb *et al.*, 2010b] to localize the head of cats and dogs, and use image segmentation to predict the body region. Color and texture features are extracted from these parts to identify the breeds. As DPMs contain the notion of parts (*i.e.*, the templates that covers sub-regions on an object), some methods also extract the descriptors from DPM parts directly [Zhang *et al.*, 2013; Chai *et al.*, 2013].

In recent years, Deep Convolutional Neural Network (DCNN) has achieved great success on image classification. It significantly advances the state of the art on large-scale datasets, such as the 1000-category ImageNet [Krizhevsky *et al.*, 2012; Sermanet *et al.*, 2014]. As DCNN has very large capacity to learn discriminative features, it can directly takes as input the raw image for general image classification. However, when it comes to fine-grained classification, part-based approach is still useful to achieve excellent performance, as shown in [Branson *et al.*, 2014; Zhang *et al.*, 2014a].

2.2 Object Detection

Object detection usually serves as the preprocessing for object recognition, as object bounding box naturally denotes the region of interest. However, detecting the object automatically and accurately is not easy, especially when there are large intra-class appearance variations. The difficulty of this problem has prompted great progress in the research of visual features and classification models. Since the ground-breaking work of [Viola and Jones, 2001], vision community gradually pushes the limit, and targets more challenging problems such as highly deformable objects in unconstrained aspects, compounded with cluttered background.

Classifier and feature are the basis of object detector. A variety of classifiers have been developed, including neural network [Rowley *et al.*, 1998], boosted classifier [Viola and Jones, 2001], support vector machine [Cortes and Vapnik, 1995], and random forest [Breiman, 2001; Gall *et al.*, 2011]. Object detector is generally applied in a sliding-window fashion, evaluating each possible sub-window in an image. This is a computationally expensive process, as the location, scale and orientation of the target object are unknown. To speed up object detection, cascaded structure [Viola and Jones, 2001; Zhang and Viola, 2007; Felzenszwalb *et al.*, 2010a; Cevikalp and Triggs, 2012] and branch-and-bound framework [Lampert *et al.*, 2008; Blaschko and Lampert, 2009] were designed.

Besides classifiers, there are extensive efforts in designing good features for object detection. The most popular ones are gradient-based features such as SIFT [Lowe, 2004] and HOG [Dalal and Triggs, 2005]. As HOG grids can form rectangular windows with various aspect ratios, they are widely used to detect all kinds of objects, as demonstrated by the Dalal-Triggs pedestrian detector [Dalal and Triggs, 2005], the well-known DPMs [Felzenszwalb *et al.*, 2010b], the Exemplar-SVM detector [Malisiewicz *et al.*, 2011], and other modern object detectors [Bourdev *et al.*, 2010; Zhu *et al.*, 2010]. However, HOG features have limitations. [Ren and Ramanan, 2013] claims that HOG lacks the ability to represent richer patterns than edge and contour. [Vondrick *et al.*, 2013] also reveals that in HOG space, instances of different classes may look the same, which is the cause of false positives in object detection. In addition, HOG features are criticized by its rigidity and lack of localization accuracy. To design better features, [Ren and Ramanan, 2013] proposes Histogram of Sparse Codes (HSC), which is a higher level image representation than HOG. [Dollár *et al.*, 2009; Dollár *et al.*, 2010] propose integral channel features, which are discriminatively selected to build a cascade detector [Zhang and Viola, 2007].

Parts also play an important role in object detection, no matter whether they carry semantic meanings or not. The most successful object detectors include DPMs [Felzenszwalb *et al.*, 2010b] and Poselets-based detectors [Bourdev and Malik, 2009; Bourdev *et al.*, 2010]. DPMs is built on the pictorial structure [Felzenszwalb and Huttenlocher, 2005], where part templates are arranged in a deformable configuration, and a mixture of models are used to capture the pose variations. Such flexibility in DPMs results in a much better model than rigid full-body detector [Dalal and Triggs, 2005]. [Bourdev and Malik, 2009; Bourdev *et al.*, 2010] introduce a new notion of parts, Poselets, which are tightly clustered in configuration space of keypoints, as well as appearance space. Poselet activations serve as context for each other, and false detections of Poselets can be suppressed by exploiting their mutual consistency.

To further improve the performance of object detection, additional information is incorporated, including object-level context and image segmentation. A straightforward instantiation of context is the co-occurrence of objects in multi-class detection tasks [Felzenszwalb *et al.*, 2010b; Sadeghi and Farhadi, 2011; Desai *et al.*, 2011]. Some works also mine the context information from the surrounding background regions [Li *et al.*, 2011] or even from the whole image [Blaschko and Lampert, 2009; Song *et al.*, 2011]. As object usually occupies the foreground in an image, image segmentation and object detection can be solved jointly, serving as context for each other, as shown in [Gould *et al.*, 2009; Gao *et al.*, 2011; Fidler *et al.*, 2013].

Recently, DCNN has also demonstrated its capacity in object detection tasks [Szegedy *et al.*, 2013; Sermanet *et al.*, 2014; Girshick *et al.*, 2014]. [Szegedy *et al.*, 2013] builds the last layer of DCNN as a regression model, which instead of outputting the probabilities of class labels, directly predicts the normalized coordinates of the object bounding box. [Sermanet *et al.*, 2014] uses a sliding-window paradigm to apply DCNN as conventional object detector. As the sliding-window detection is computationally expensive, speed-up can be achieved by generating region proposals for the object, and feeding them to DCNN to evaluate [Girshick *et al.*, 2014].

2.3 Part Localization

Part localization goes beyond object detection, as it requires additional understanding about the pose of object. As such, the localization results provide detailed information about the object, facilitating subsequent tasks such as fine-grained classification. Existing works mainly focus on two aspects of the problem: one is to build strong part detectors, and the other is to design reliable and expressive spatial models. As most of the techniques in object detector can be directly applied to part detector, we will mainly discuss the spatial models in this section.

Part localization relies on prior knowledge about the global configuration. In other words, the relations between parts encoded by spatial models should be exploited. Statistical shape models such as Active Shape Models [Milborrow and Nicolls, 2008] and Active Appearance Models [Cootes *et al.*, 2001] model the part configurations of an object with multivariate Gaussian distribution. Although the follow-up works improved on fitting the models to the image [Matthews and Baker, 2004; Saragih *et al.*, 2009], these methods still cannot handle a wide range of pose variations.

As people target more challenging datasets [Ramanan, 2006; Johnson and Everingham, 2010; Wah *et al.*, 2011], tree-structured model becomes very popular due to its generalization ability and computational efficiency. It is a graphical model, where nodes denote the parts and edges denote the part relations. Such model is applied in the same way as DPM, except for that parts are explicitly defined and annotated during training. Following the work of [Felzenszwalb and Huttenlocher, 2005], different variants have been developed [Everingham *et al.*, 2006; Yang and Ramanan, 2011; Zhu and Ramanan, 2012; Sapp and Taskar, 2013; Sun and Savarese, 2011; Branson *et al.*, 2011]. All of these methods use a mixture of trees to capture the pose variations. [Sun and Savarese, 2011; Branson *et al.*, 2011] additionally employ hierarchical representation to enrich the tree-structured model with parts at different granularity levels. To train the model on large-scale datasets, a fast structured SVM solver is developed in [Branson *et al.*, 2013]. In spite of these efforts, a fundamental problem with the tree-structured model still remains: as the model only captures pair-wise spatial relations between parts, higher-order part interactions are ignored, making the model lack strength and precision in constraining the whole configuration. Such drawback is not obvious in human pose estimation, presumably because the part interactions on human body are mainly confined within limbs. There are efforts to go beyond the tree structure. [Wang *et al.*, 2011] proposes a loopy model that organizes parts in a hierarchy, while [Ramakrishna *et al.*, 2014; Tompson *et al.*, 2014] use complete graph, treating all the other parts as the neighbors of a target part.

Another family of models is Constrained Local Model [Cristinacce and Cootes, 2006; Belhumeur *et al.*, 2011; Amberg and Vetter, 2011; Zhou *et al.*, 2013], which employs global shape model to guide the search of individual parts. [Amberg and Vetter, 2011] uses a generic 3D face model as the shape prior, limiting its applicability to relatively rigid objects such as human face. [Belhumeur *et al.*, 2011] proposes Consensus-of-Exemplars (CoE) approach that uses an ensemble of 2D exemplars to capture the shape variations. [Zhou *et al.*, 2013] presents exemplar-based graph matching, which replaces the consensus stage of CoE approach with explicit candidate selection. We find exemplar-based approach is very promising in that labeled exemplars literally capture plausible part configurations without assuming the distribution of part relations. However, their application in other domains than human faces is yet to be explored.

A very different thread of research is shape regression which maps image features to shape increment [Dantone *et al.*, 2012; Cao *et al.*, 2012]. It requires that features should have strong correlation to the shape, and such correlation should be consistent enough across different samples. Low-level features such as raw pixel values suffice for human faces, presumably due to that local visual patterns on human faces are relatively similar. However, for objects with more diverse appearance, it is hard to find appropriate features. Another limitation of shape regression is that it needs object bounding box as input, which is not practical.

DCNN has also been successfully applied to face part localization [Sun *et al.*, 2013] and human pose estimation [Toshev and Szegedy, 2014; Chen and Yuille, 2014; Tompson *et al.*, 2014]. [Sun *et al.*, 2013] proposes a cascade of three-level convolutional networks, where each level consists of multiple networks targeting different input regions. As for human pose estimation, [Toshev and Szegedy, 2014] builds DCNN-based part regressors to achieve the effect of holistic pose reasoning, while [Chen and Yuille, 2014; Tompson *et al.*, 2014] combine DCNN-based part detectors with graphical models.

Recently, we observed the trend of designing image dependent spatial models. [Pishchulin *et al.*, 2013a; Pishchulin *et al.*, 2013b] extend pictorial structure by using Poselet dependent unary and pairwise terms, where Poselets are detected on the fly [Bourdev *et al.*, 2010]. [Chen and Yuille, 2014] incorporates image dependent pairwise relations into the formulation of a graphical model. Our work in Chapter 7 also instantiates this idea.

Chapter 3

Dog Breed Classification

In this chapter, we build a working system for fine-grained classification, where the goal is to identify dog breeds. The main characteristic of such problem is that instances from different classes share the same part configuration (which makes classification difficult) but have wide variation in the shape and appearance of parts (which makes part detection difficult). However, we will show that using a state-of-the-art part localizer, we can detect the parts with high accuracy, and that extracting features based on the detected parts improves the classification performance.

After collecting and analyzing the dataset, we design a hierarchy of parts (*e.g.*, face, eyes, nose, ears, etc.) that helps build correspondence between different samples. To localize the face parts, we first use a sliding window detector to locate dog face. Then we try to localize eyes and nose within the face region by using Consensus-of-Exemplars approach (CoE) [Belhumeur *et al.*, 2011] which combines appearance-based detections with a large set of exemplar-based geometric models. Based on this small set of face parts (eyes and nose), we align the test sample with training exemplars from each dog breed, and hypothesize locations of additional breed-specific parts, such as ear whose position and appearance vary greatly across breeds (detecting such parts is significantly more expensive and less reliable). Once all the part locations are estimated, we extract image features at and around the parts for use in classification. An illustration of our system at the testing stage is shown in Fig. 3.1.

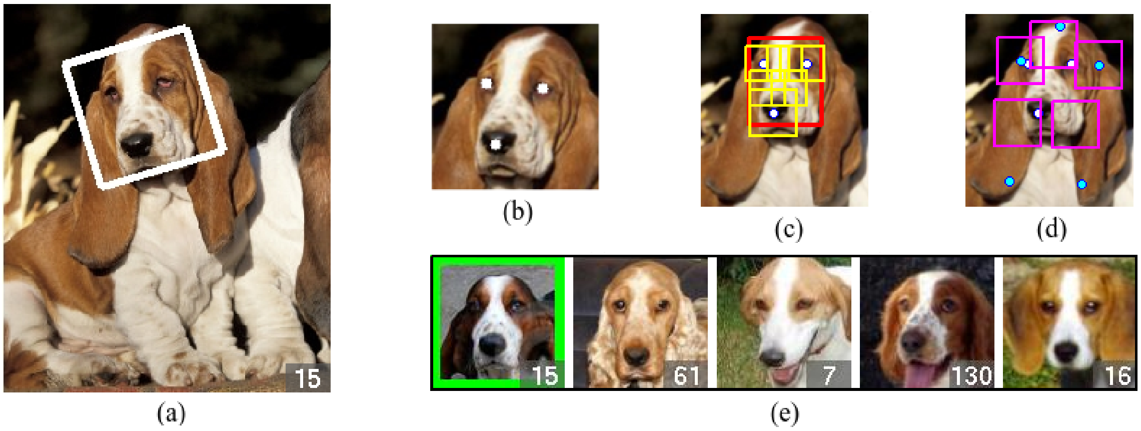


Figure 3.1: (a) Given an image, our method automatically detects dog face, (b) localizes eyes and nose, (c) aligns the face and extracts grayscale SIFT features (yellow windows) and a color histogram (red window), (d) infers remaining part locations from exemplars (cyan dots) to extract additional SIFT features (magenta windows), and (e) predicts the breed (green box) along with the next best guesses from left to right. The numbers correspond to breed names listed in Tab. 3.1.

3.1 Columbia Dog Dataset

To train and evaluate our system, we have created a dog dataset[1] of natural images of dogs, downloaded from sources such as Flickr, Image-Net, and Google. The dataset contains 133 breeds of dogs with $8,351$ images. The images were not filtered, except to exclude images in which the dog's face was not visible. Sample faces from all the breeds are shown in Fig. 3.2, and the list of breed names is shown in Tab. 3.1. Not only is there great variation across breeds – making detection a challenge, but there is also great variation within breeds – making identification a challenge. Please see the blowup of sample images of Breed 97: Lakeland terrier. Note the variations in the color, ear position, fur length, pose, lighting and even expression.

Each of the images was submitted to Amazons Mechanical Turk (MTurk) to have the breed label verified by multiple workers. Afterward, each image was submitted again to MTurk to have parts of the dog's face labeled. Eight points were labeled in each image by three separate workers. If there was gross disagreement amongst the workers in the locations of these points, we resubmitted the

[1] The dataset is available at http://faceserv.cs.columbia.edu/DogData/

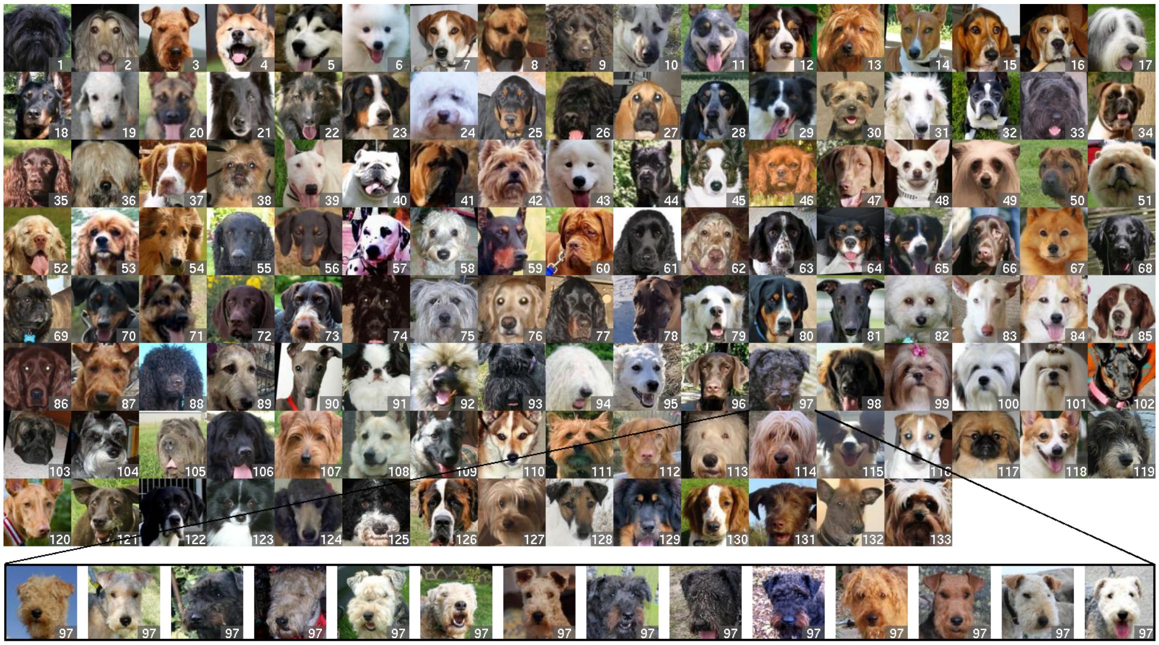


Figure 3.2: Sample faces from all the breeds. The breeds are numbered according to the alphabetical order of names.

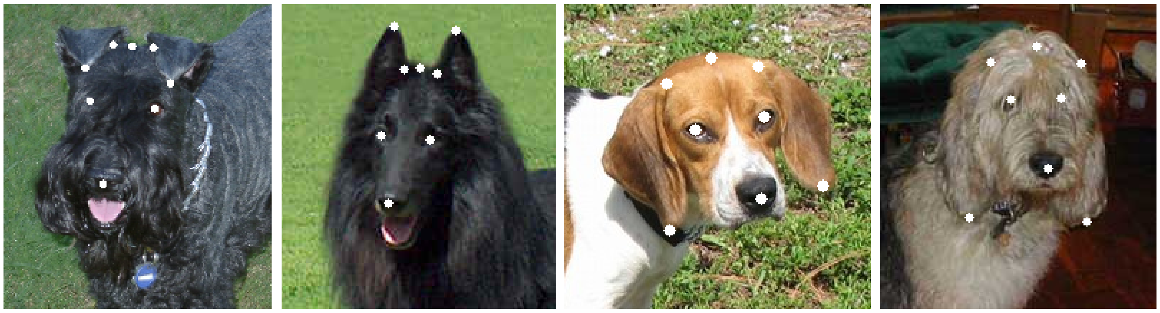


Figure 3.3: Sample dog images from our dataset, with parts labeled by MTurk workers.

image again for relabeling. The points that were labeled were the eyes, the nose, the tips of both ears, the top of the head, and the inner bases of the ears. In Fig. 3.3, we show the average location, over three workers, for these eight points.

1:Affenpinscher (80)	28:Bluetick coonhound (44)	55:Curly-coated retriever (63)	82:Havanese (76)	109:Norwegian elkhound (56)
2:Afghan hound (73)	29:Border collie (93)	56:Dachshund (82)	83:Ibizan hound (58)	110:Norwegian lundehund (41)
3:Airedale terrier (65)	30:Border terrier (65)	57:Dalmatian (89)	84:Icelandic sheepdog (62)	111:Norwich terrier (55)
4:Akita (79)	31:Borzoi (70)	58:Dandie dinmont terrier (63)	85:Irish red and white setter (46)	112:Nova scotia duck tolling retriever (67)
5:Alaskan malamute (96)	32:Boston terrier (81)	59:Doberman pinscher (59)	86:Irish setter (66)	113:Old english sheepdog (49)
6:American eskimo dog (80)	33:Bouvier des flandres (56)	60:Dogue de bordeaux (75)	87:Irish terrier (82)	114:Otterhound (44)
7:American foxhound (63)	34:Boxer (80)	61:English cocker spaniel (76)	88:Irish water spaniel (64)	115:Papillon (79)
8:American staffordshire terrier (82)	35:Boykin spaniel (66)	62:English setter (66)	89:Irish wolfhound (66)	116:Parson russell terrier (38)
9:American water spaniel (42)	36:Briard (81)	63:English springer spaniel (66)	90:Italian greyhound (73)	117:Pekingese (60)
10:Anatolian shepherd dog (62)	37:Brittany (62)	64:English toy spaniel (49)	91:Japanese chin (71)	118:Pembroke welsh corgi (66)
11:Australian cattle dog (83)	38:Brussels griffon (71)	65:Entlebucher mountain dog (53)	92:Keeshond (55)	119:Petit basset griffon vendeen (39)
12:Australian shepherd (83)	39:Bull terrier (87)	66:Field spaniel (41)	93:Kerry blue terrier (44)	120:Pharaoh hound (49)
13:Australian terrier (58)	40:Bulldog (66)	67:Finnish spitz (42)	94:Komondor (55)	121:Plott (35)
14:Basenji (86)	41:Bullmastiff (86)	68:Flat-coated retriever (79)	95:Kuvasz (61)	122:Pointer (40)
15:Basset hound (92)	42:Cairn terrier (79)	69:French bulldog (64)	96:Labrador retriever (54)	123:Pomeranian (55)
16:Beagle (74)	43:Canaan dog (62)	70:German pinscher (59)	97:Lakeland terrier (62)	124:Poodle (62)
17:Bearded collie (77)	44:Cane corso (80)	71:German shepherd dog (78)	98:Leonberger (57)	125:Portuguese water dog (42)
18:Beauceron (63)	45:Cardigan welsh corgi (66)	72:German shorthaired pointer (60)	99:Lhasa apso (53)	126:Saint bernard (37)
19:Bedlington terrier (60)	46:Cavalier king charles spaniel (84)	73:German wirehaired pointer (52)	100:Lowchen (42)	127:Silky terrier (51)
20:Belgian malinois (78)	47:Chesapeake bay retriever (67)	74:Giant schnauzer (51)	101:Maltese (60)	128:Smooth fox terrier (38)
21:Belgian sheepdog (80)	48:Chihuahua (68)	75:Glen of imaal terrier (55)	102:Manchester terrier (36)	129:Tibetan mastiff (60)
22:Belgian tervuren (59)	49:Chinese crested (63)	76:Golden retriever (80)	103:Mastiff (72)	130:Welsh springer spaniel (55)
23:Bernese mountain dog (81)	50:Chinese shar-pei (62)	77:Gordon setter (54)	104:Miniature schnauzer (53)	131:Wirehaired pointing griffon (37)
24:Bichon frise (77)	51:Chow chow (78)	78:Great dane (50)	105:Neapolitan mastiff (39)	132:Xoloitzcuintli (33)
25:Black and tan coonhound (46)	52:Clumber spaniel (61)	79:Great pyrenees (74)	106:Newfoundland (62)	133:Yorkshire terrier (38)
26:Black russian terrier (51)	53:Cocker spaniel (59)	80:Greater swiss mountain dog (57)	107:Norfolk terrier (58)	
27:Bloodhound (80)	54:Collie (71)	81:Greyhound (70)	108:Norwegian buhund (33)	

Table 3.1: List of breed names. Each breed name is numbered, with the number of images shown to the right.

3.2 Localizing Dog Face and Face Parts

Based on the annotations we collect, we divide the eight parts into two groups: generic parts (eyes and nose) and breed-specific parts (the rest five parts). There are great variations across breeds, especially on the breed-specific parts, which poses a challenge to the part localizer. Therefore, we choose to first localize the generic (also much easier) parts with high accuracy at relatively low cost, and incorporate the localization of breed-specific parts in the classification stage.

We design a three-step method to localize the generic parts: we first detect the dog face, producing top-5 candidate face windows; then localize the generic parts within each face window; in the end, the scores of face and parts are combined to determine the best face candidate as well as its corresponding part locations.

3.2.1 Face Detection

We create a dog face detector capable of detecting faces of all the dog breeds in our dataset. Although we do not view our dog face detector as a technical contribution, it is a necessary component of our complete vision system and is described briefly here.

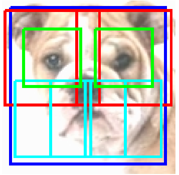


Figure 3.4: Feature windows for dog face detection. Different colors indicate different scales.

The detector is a SVM classifier with grayscale SIFT [Lowe, 2004] descriptors as features. Eight SIFT descriptors are extracted at fixed positions relative to the center point. The positions and scales are chosen to roughly align with the geometry of a dog's face (eyes and nose), as shown in Fig. 3.4. Once extracted, these descriptors are then concatenated into a single $1,024$-dimensional feature vector for our SVM classifier. We use about $4,700$ positive samples for training.

To generate the negative training samples from a dog image, we randomly scale the image, and randomly place windows in the image such that they have little overlap with the ground-truth dog face. We then extract the same eight SIFT descriptors within the non-face windows. As negative examples are plentiful, we randomly select about $13,000$ ones. With both positive and negative samples in hand, we train our SVM classifier using an RBF kernel.

As the SVM classifier is trained at a fixed rotation and scale, at detection time we must search not only over location, but also over rotation and scale. We threshold and merge the repeated detections with non-maximum suppression, and keep up to five detection windows with the highest scores as the candidates.

3.2.2 Face Part Localization

Given a candidate face window, we can prune the searching space when localizing the face parts. As the face detection result tells the rough location, size, and rotation of the face, we can normalize the face to facilitate the part localization, as shown in Fig. 3.1 (b).

To localize the parts of the dog face, we build on the Consensus-of-Exemplars approach [Belhumeur *et al.*, 2011]. The method can accurately localize the generic parts (eyes and nose); we handle more challenging face parts during the breed identification process (Sec. 3.3). Following [Belhumeur *et al.*, 2011], we combine low-level part detectors with labeled images that model part locations. We first train a sliding window SVM detector for each face part. If we let I denote a query image, let $\mathbf{p} = \{p^1, p^2, \dots, p^n\}$ denote the locations of the targets parts in the image, and let $D = \{d^1, d^2, \dots, d^n\}$ denote the detector responses for these parts in I, then our goal is to compute

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} P(\mathbf{p}|D). \tag{3.1}$$

In [Belhumeur *et al.*, 2011], probable locations for these parts are dictated by exemplars that have been manually labeled. These exemplars help create conditional independence between different parts. Let $X_k = \{x_k^1, x_k^2, \dots, x_k^n\}$ be the locations of the parts in the k^{th} exemplar image, Eq. 3.1 is then rewritten as

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \sum_{k=1}^{m} \int_{t \in T} \prod_{i=1}^{n} P(\Delta x_{k,t}^i) P(p^i|d^i) dt. \tag{3.2}$$

Here the summation is over all m exemplars, *i.e.*, in our case over all labeled examples of parts of dogs' faces. The integral is over similarity transformations t of the exemplars. $x_{k,t}^i$ denotes the part i's location in the k^{th} exemplar, transformed by t, and $\Delta x_{k,t}^i = x_{k,t}^i - p^i$ denotes the difference in location of the part i in the query image from that of the transformed exemplar. $\Delta x_{k,t}^i$ is modeled as a Gaussian distribution with zero mean. This amounts to introducing a generative model of part locations in which a randomly chosen example is transformed and placed in the image with noise added. After assuming independence of parts in the deviation from the model (multiplication in Eq. 3.2), we then marginalize the model out (summation and integral in Eq. 3.2).

This optimization is then solved by a RANSAC-like procedure, in which a large number of exemplars are randomly selected and fit to the modes of the detection response maps. For each hypothesis in which an exemplar is transformed into the image, the hypothesized part locations are combined with the detector output; the best fitting matches then pool information, reaching a consensus about the part locations. The consensus for part i is

$$\hat{p^i} = \arg\max_{p^i} \sum_{k,t \in \mathcal{M}} P(\Delta x_{k,t}^i) P(p^i|d^i), \tag{3.3}$$

where we sum over the best fits $\mathcal{M}$ (produced by RANSAC) between the exemplars and detectors.

3.2.3 Re-scoring Faces Using Localized Parts

As our dog face detector is not perfect, the hit rate is not satisfactory (about 90%) if we simply choose the detection window with the highest score. If the face detection fails, the subsequent processes are likely to fail too. Fortunately, we also find that among the five best scoring windows from a test image, the hit rate can be as high as 98% which is acceptable in our problem. As the face detection module is unaware of the face parts, we seek to re-score the obtained five face windows by incorporating the scores of their corresponding part estimations. Formally, let $\phi(\mathbf{f})$ denote the raw detection score of face window $\mathbf{f}$, $\psi(p^i)$ denote the raw detection score of location p^i for part i. After converting them to probabilities with Sigmoid function $\sigma(\cdot)$, the final score of the face is computed as

$$\hat{\phi}(\mathbf{f}) = \sigma\left(\phi(\mathbf{f})\right)\left(\prod_i \sigma\left(\psi(p^i)\right)\right)^{\frac{1}{n}}, \tag{3.4}$$

where n denotes the number of parts. In the end, we conduct the breed classification on the face window with the highest $\hat{\phi}(\mathbf{f})$.

3.3 Dog Breed Classification

Our classification algorithm focuses entirely on the face of dog. This is partly because face is largely a rigid object, simplifying the problem of comparing images of different dogs. However, we are also guided by the intuition that dog breeds are largely identifiable from their faces. A dog's body shape is not only difficult to identify and often not present in images, but also offers little additional information except in a more extreme cases (*e.g.*, dachshunds).

3.3.0.1 Formulation

If we denote the breed of a dog by B, our goal is to compute

$$\hat{B} = \arg\max_B P(B|I). \tag{3.5}$$

Let the part locations in the query image I be given by $\mathbf{p}$. Then

$$\hat{B} = \arg\max_B \int P(B|I,\mathbf{p})P(\mathbf{p}|I)d\mathbf{p}. \tag{3.6}$$

Here we integrate over all possible locations of the parts $\mathbf{p}$ in the image I.

If the part locations can be accurately localized, then $P(\mathbf{p}|I)$ is approximately a delta function about the true locations of the parts. Then if we write

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} P(\mathbf{p}|I), \tag{3.7}$$

we have

$$\hat{B} = \arg\max_{B} P(B|I, \hat{\mathbf{p}})P(\hat{\mathbf{p}}|I). \tag{3.8}$$

Note that $P(\hat{\mathbf{p}}|I)$ is independent of B, so that

$$\hat{B} = \arg\max_{B} P(B|I, \hat{\mathbf{p}}). \tag{3.9}$$

This means that we can break our problem into two parts. First, we must compute $\arg\max_{\mathbf{p}} P(\mathbf{p}|I)$ as explained in the previous section. Next, we must compute $\arg\max_{B} P(B|I, \hat{\mathbf{p}})$. Note that

$$P(B|I, \hat{\mathbf{p}}) = \frac{P(I|B, \hat{\mathbf{p}})P(B|\hat{\mathbf{p}})}{P(I|\hat{\mathbf{p}})} \tag{3.10}$$

where the denominator $P(I|\hat{\mathbf{p}})$ is a constant that does not affect which breed will maximize the probability. So

$$\hat{B} = \arg\max_{B} P(I|B, \hat{\mathbf{p}})P(B|\hat{\mathbf{p}}). \tag{3.11}$$

However, our knowledge of what constitutes a breed is completely given by our set of labeled exemplar images. We divide the information in these images into two parts. First, we let $\mathbf{p}^B$ denote the known locations of the parts of all exemplars for breed B. Then we let D^B denote descriptors characterizing the appearance of the exemplars for breed B. These descriptors are extracted at corresponding part locations given by $\mathbf{p}^B$. So we can rewrite Eq. 3.11 as

$$\hat{B} = \arg\max_{B} P(I|D^B, \mathbf{p}^B, \hat{\mathbf{p}})P(D^B, \mathbf{p}^B|\hat{\mathbf{p}}). \tag{3.12}$$

In approximating this, we assume that the breed appearance descriptors D^B are independent of their positions, and we have a uniform distribution over breeds. This allows us to rewrite Eq. 3.12 as

$$\hat{B} = \arg\max_{B} P(I|D^B, \mathbf{p}^B, \hat{\mathbf{p}})P(\mathbf{p}^B|\hat{\mathbf{p}}). \tag{3.13}$$

This suggests that we compute $P(I|D^B, \mathbf{p}^B, \hat{\mathbf{p}})$ by measuring how well the appearance of the query image at and around the part locations given by $\hat{\mathbf{p}}$ agrees with the appearance of our exemplars in their corresponding locations $\mathbf{p}^B$.

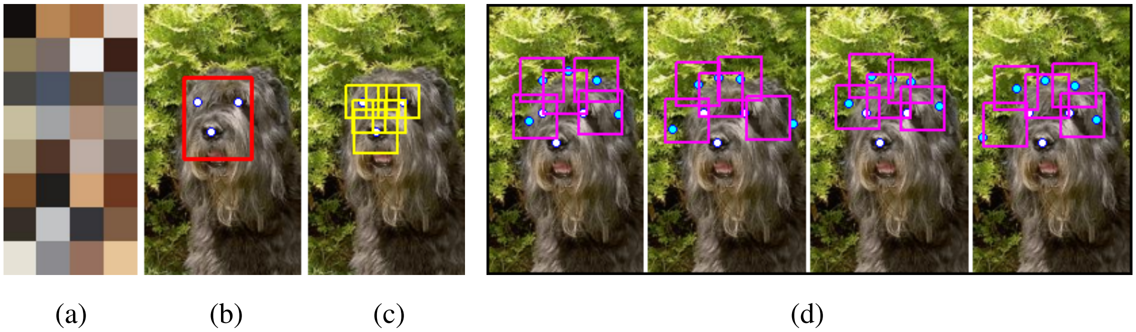


Figure 3.5: (a) The cluster centers used to create the color histogram. (b) The window used to extract the color histogram based on detected locations of the eyes and nose (white dots). (c) The SIFT descriptor windows (yellow) dictated by the eyes and nose. (d) Four different sets of inferred locations (cyan dots) and the SIFT descriptor windows (magenta) dictated by these.

3.3.1 Training and Testing

To evaluate the first probability in Eq. 3.13, we train one vs. all SVMs for each breed B, and we use two types of features: grayscale SIFT descriptors and a color histogram. We want to center the collection of SIFT features at places dictated by the part locations. However, at this point we are only able to locate the eyes and nose with high accuracy. Since the other parts are more breed-specific, we infer the locations of the those parts from exemplars of breed B when generating the negative training samples (to be analogous to the testing scenario). During testing, for each breed B we choose r exemplars whose eye and nose locations are closest to the query's after alignment with a similarity transform. These exemplars are the ones that are most likely in the same pose as the query image. Consequently, when we use these similarity transformations to infer the locations of additional face parts, the parts are likely to align with those of the query image of the same breed. For example, Fig. 3.5 (d) shows the detected locations (white dots) for the eyes and nose, and four different sets of locations for the remaining parts (cyan dots) inferred from training exemplars of the same breed.

To extract features, we center three SIFT descriptors at the eyes and nose. We center another three descriptors at the three midpoints along the lines connecting the eyes and nose. The windows for the six SIFT descriptors are shown in yellow in Fig. 3.5 (c). We place additional five descriptors at the bases of the ears and the midpoints of the lines connecting the eyes with the other inferred

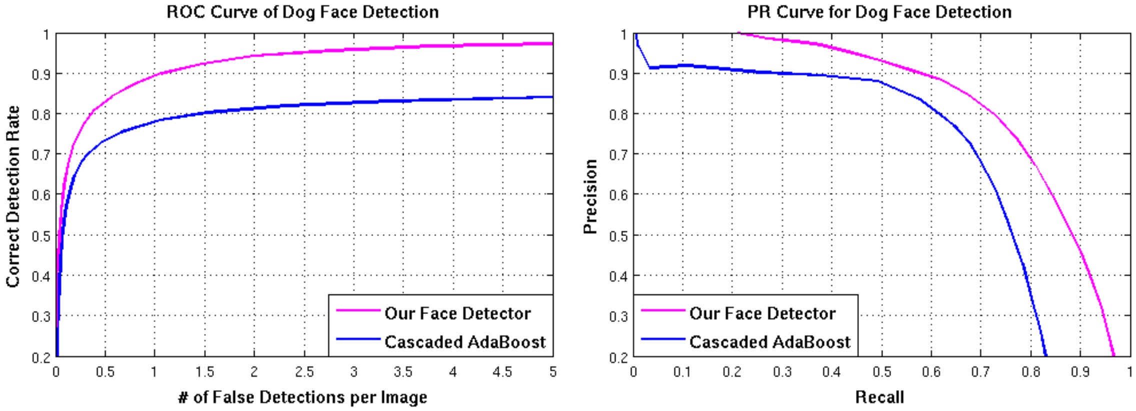


Figure 3.6: The ROC and Precision/Recall (PR) curves for dog face detection. The criterion of a correct detection is that the intersect over union ratio is above 0.6.

parts. The windows for the additional five SIFT descriptors are shown in magenta in Fig. 3.5 (d). The color histogram is computed over a rectangular region centered on the dog's face, shown in red in Fig. 3.5 (b). The histogram is created using 32 color centers computed from a k-means clustering across all exemplar images of all dogs, shown in Fig. 3.5 (a). The 32 color features along with the 11 SIFT features are concatenated to produce a $1,440$-dimensional feature vector.

Given a query image, the selected exemplars produces r feature vectors for each breed. We evaluate each of these using our one vs. all SVM and allow the best scoring feature vector to represent the breed. In practice, we choose $r = 10$.

The second probability in Eq. 3.13 can be computed directly from the distribution of part locations in our exemplars. Since we are aligning the eyes with a similarity transform, only the relative location of the nose could carry information about the breed. But we have not found it helpful to include this.

3.4 Experiments and Results

In this section, we evaluate our system extensively using our Dog dataset. There are 133 breeds of $8,351$ images. We randomly split the images of each breed in a fixed ratio to get $4,776$ training images and $3,575$ test images. We double the size of the training data by left-right flipping.

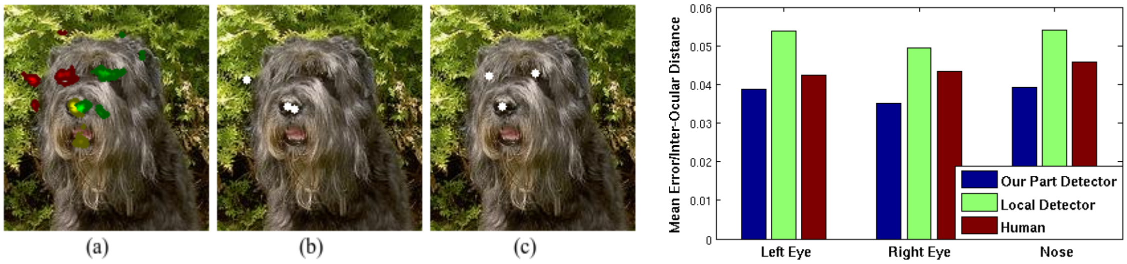


Figure 3.7: Left: An example of part detection. (a) Original image overlaid with heat maps from part detectors. Red is used for the left eye, green for the right eye, and yellow for the nose; better scores are shown as brighter. (b) Detected parts using the maximum value of the detection scores. (c) Detected parts using our full model. Right: Mean localization error divided by the inter-ocular distance.

3.4.1 Face Detection

We implemented a baseline face detector which is a cascaded AdaBoost detector with Haar-like features [Viola and Jones, 2001]. We compare the performance of our detector with the Adaboost detector on our test images in Fig. 3.6. While Adaboost detector has seen much success in detecting human faces, it is sometimes plagued by unwanted false positives. Perhaps due to the extreme variability in geometry and appearance of dog faces, this weakness in the cascaded Adaboost detectors is exacerbated in the dog face domain. Even training on considerably more data (mining hard negatives) and using 20 cascades, we could not create a detector with a desirable ratio of true positives to false positives.

3.4.2 Part Localization

We evaluate the accuracy of part localization given the ground-truth face windows. We compare our full model with a simpler method that locates a part at the mode of the SVM-based sliding window detector for each part. We also compare our results with the agreement of human labelers by determining the distance between the location indicated by one human labeler and the average of the other two. We show qualitative results and make a quantitative comparison in Fig. 3.7. Note that our full model improves over the results of just using a local part detector, and that our localization error is better than the agreement among human labelers.

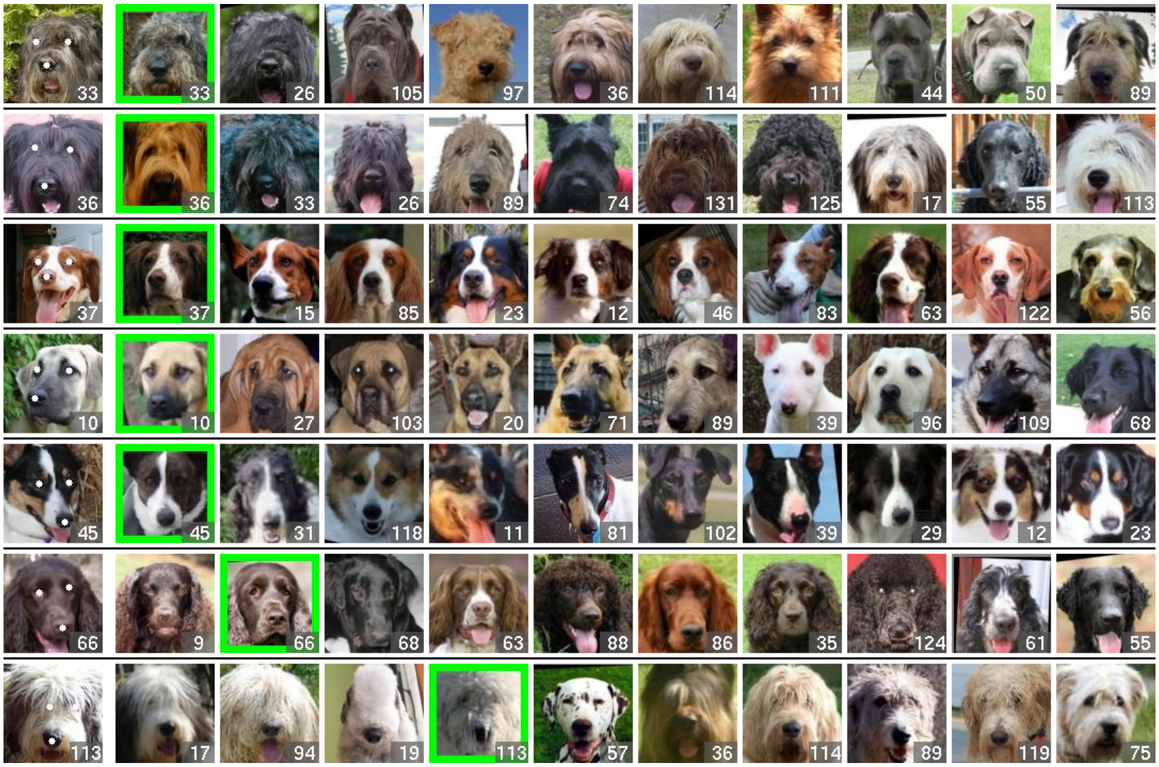


Figure 3.8: Classification examples. Testing samples are in the first column, the closest 10 breeds based on our method are shown to the right.

3.4.3 Breed Classification

As our goal is to design a working system, we evaluate the whole pipeline where the breed classification is the last module. Fig. 3.8 gives qualitative results for some query images. For each query in the first column, we overlay the image with the detected part locations for the eyes and nose. To better show the performance, we rank the breeds based on their probability score. Our system works with high accuracy, failing mostly when the face detection fails (these failures are excluded in these examples) or when the parts detection fails on samples in which fur completely occludes the eyes.

We evaluate our pipeline quantitatively, and compare with three other methods: a bag-of-words (BoW) model with spatial tiling [Vidaldi and Zisserman, 2011], a multiple kernel learning (MKL) approach [Vedaldi *et al.*, 2009] used in bird recognition [Branson *et al.*, 2010], and locally constrained linear coding (LLC) [Wang *et al.*, 2009] also applied to a bird dataset [Yao *et al.*, 2011]. We apply each of these methods inside a cropped window found by selecting the face window with

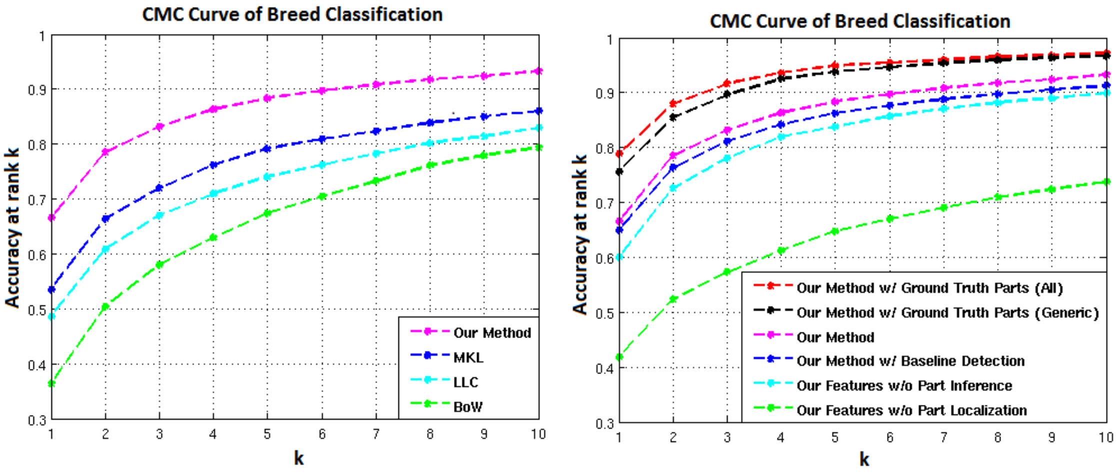
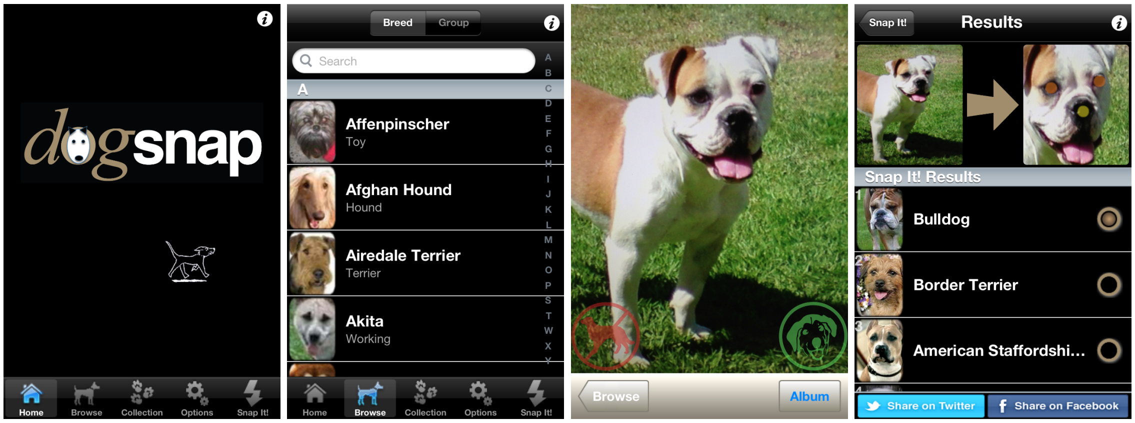


Figure 3.9: Performance curves for dog breed identification, showing how often the correct breed appears among the top 1–10 guesses. On the left we show our method compared with three other methods. On the right we show multiple variants of our method: from bottom to top, the first uses our feature set sampled on a grid within our detected face window; the second uses part localization, but only extracts features at generic parts, and uses the highest scoring window from the face detector; the third incorporates the breed-specific parts into the second variant; the fourth is our full method; the fifth one uses ground-truth locations of generic parts; the last one uses all the ground-truth parts.

the highest face detector score within the image; if we use the uncropped images, the performance of all methods is poor – below 20% on the first guess. This gives each method the benefit of face detection and allows us to evaluate the additional gain produced by our system using part detection. In Fig. 3.9-left we show performance curves for all the methods. Our method significantly outperforms existing approaches, getting the breed identification correct 67% of the time on the first guess vs. 54% for MKL, 49% for LLC, and 36% for BoW. This comparison demonstrates the effectiveness of part-based method in fine-grained classification.

In Fig. 3.9-right we show multiple variants of our approach. As a baseline, we use our feature set, extracted on a grid, rather than at part locations, given the best scoring face window. We can see that the use of parts results in a substantial improvement in the performance. We also evaluate the accuracy when we only use the features at generic parts (without part inference). Though it has



(a) (b) (c) (d)

Figure 3.10: Screenshots of our iPhone App. (a) Home screen. (b) Browse screen with the dog breeds. (c) Dog camera. (d) Detected dog face and parts, with results.

decent performance, we do get benefit by incorporating the features from breed-specific parts (about 5% boost on the first guess). We can also see that the use of parts to re-score face detection results in a further improvement in performance, eliminating approximately 20% of the errors for the top ten guess. To have an idea about how the accuracy of part localization matters, we use the ground-truth parts rather than detected parts. First, we only use the ground-truth generic parts, following the same classification procedure as described in Sec. 3.3.1. This gives a large improvement in the classification accuracy, which suggests that there is room for further improvement by making the part localization more accurate. Second, we use all the ground-truth parts, eliminating the need of inferring breed-specific parts from training exemplars. This time, we only get about 3% increase in the first guess accuracy, with even smaller improvement for the top ten guess. Therefore, our classification approach with part inference does provide an alternative when detecting certain parts is difficult but not very crucial.

To facilitate experimentation by ourselves and others with this algorithm, we have created and released a free iPhone App for dog breed identification (see Fig. 3.10). The App allows a user to photograph a dog and upload its picture to a server for face detection, part detection, and breed identification.

3.5 Discussion

One might expect that fine-grained classification problems would be extremely difficult, that telling a Beagle from a Basset Hound would be much harder than telling a car from a computer mouse. Our main contribution is to show that much of this difficulty can be mitigated by the fact that it is possible to establish accurate correspondence between instances from a large family of related classes. We extract visual features that can be effectively located using generic and breed-specific models of part locations. An additional contribution is the creation of a large, publicly available dataset for dog breed identification, coupled with a practical system that achieves high accuracy in real-world images.

Chapter 4

Fish and Bird Species Classification

As part-based approach has shown promising results in dog breed classification, we would like to apply it to other fine-grained categories such as fishes and birds. In this chapter, we build recognition systems that are capable of identifying fish species and bird species. This time, we employ an updated version of part localizer (referred to as CoE-ext), which will be described in Chapter 5. With the improved part localizer, we seek to detect all the parts explicitly regardless of the different levels of difficulty. As a result, the classification pipeline is simplified to have three steps: (1) localize the parts, (2) extract part-based features, (3) predict the class labels. Fig. 4.1 illustrates the pipeline with a test image of fish. In the following sections, we assume the part locations are detected, and will describe how we extract the part-based features for classification.

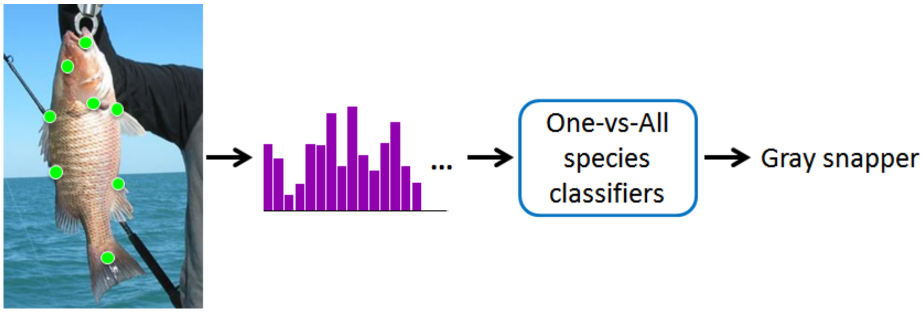


Figure 4.1: Pipeline of our fine-grained classification system: parts (green dots) are first detected from the test image, then features are extracted at the locations dictated by the parts, finally species classifiers predict the most likely label.

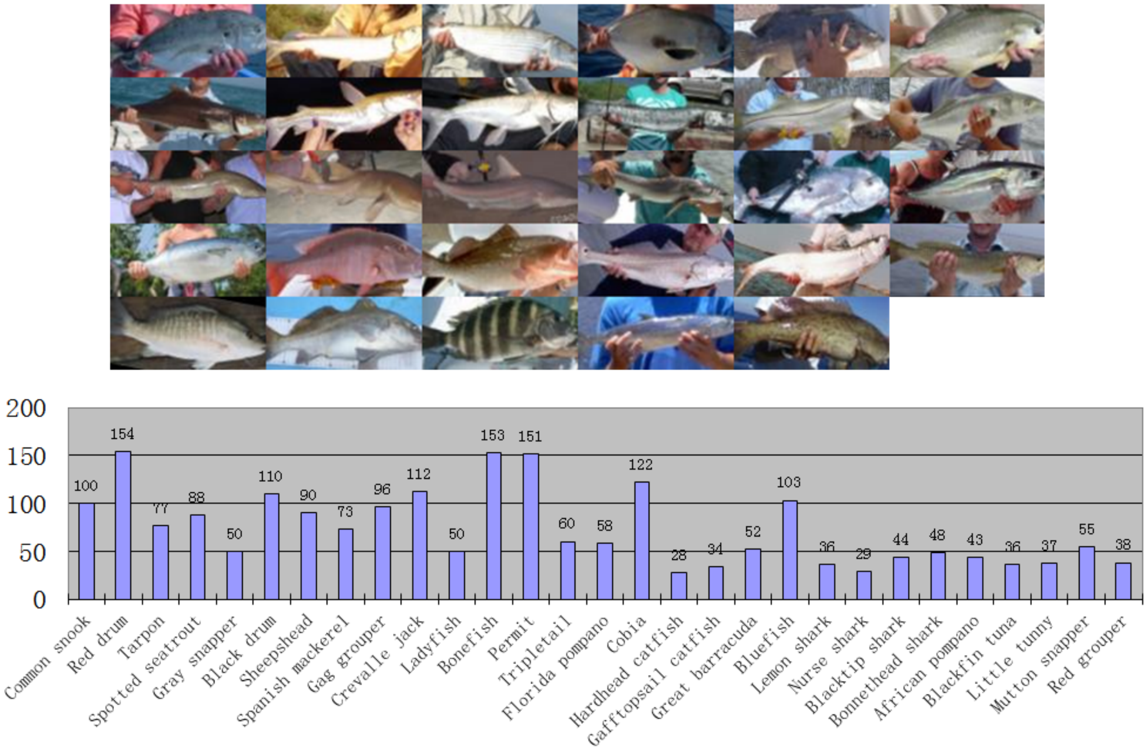


Figure 4.2: Top: sample fish images from 29 species. Bottom: the number of images per species.

4.1 Fish Species Classification

4.1.1 Columbia Fish Dataset

As there is no public fish dataset available, we collect our own data to evaluate our system. The fish dataset consists of $2,127$ images from 29 species. Some sample images and statistics of the dataset are shown in Fig. 4.2. We randomly partition the dataset with a fixed ratio for each class to generate $1,335$ training images and 792 testing images. We use the training set to build the part detectors and species classifiers, and apply them to the testing set.

Besides the fish images and species labels, we labeled eight fish parts that are common to all the species, as shown in Fig. 4.3. Similar to our dog dataset, the images was submitted to MTurk to have the species labels verified, and part locations annotated.

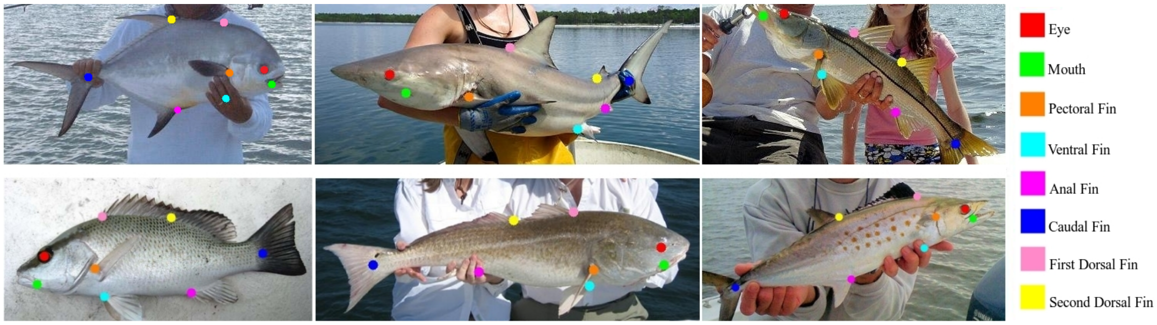


Figure 4.3: Examples of eight labeled parts.

Eye	Mouth	Second Dorsal Fin	Caudal Fin	Anal Fin	First Dorsal Fin	Pectoral Fin	Ventral Fin
3.72	4.29	6.03	6.01	6.94	4.96	4.23	6.21

Table 4.1: Average localization error for each fish part. Fish length is normalized to 100 pixels.

4.1.2 Fish Features

Given the detected parts, we first normalize the image such that the fish has fixed length (measured by the distance between Eye and Caudal Fin). Also the fish should be upright and facing right (left-right flipping may be used). From the normalized image, we extract three types of features: fine-scale SIFT features which capture the texture of the fish body (*i.e.*, fish scales), coarse-scale SIFT features which capture the shape and appearance at the parts, and color histograms which capture the color pattern of the fish body. We concatenate these features to represent a fish image. Fig. 4.4 shows the specific regions from which these features are extracted.

4.1.3 Results

We first measure the localization errors for the fish parts, which are listed in Tab. 4.1. From these numbers, we can see that parts that are more variable across different subcategories have larger localization errors. We can also see that the average error is around 5% of the fish length, which indicates that our part localizer makes reasonable predictions.

Using the part-based features, we build one vs. all species classifiers using SVM with RBF kernels. We evaluate the classification performance by plotting Cumulative Match Characteristic (CMC) curves (Fig. 4.5). The rank-1 accuracy of our method is about 72%, which is remarkable for

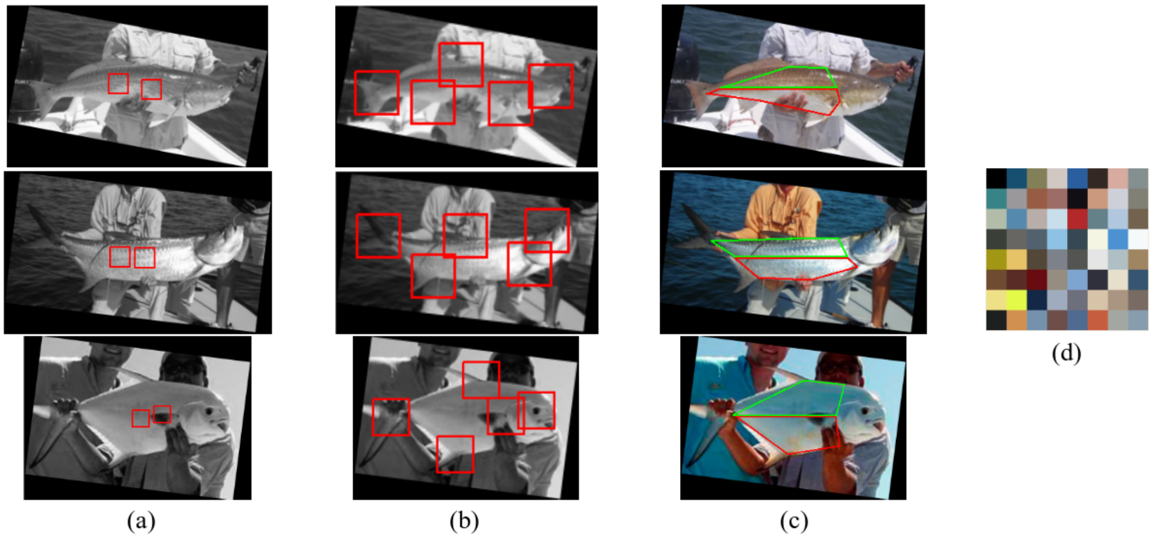


Figure 4.4: Illustration of fish features extracted from the normalized images. (a) Two fine-scale SIFT descriptors (grayscale) are extracted at the halfway points between upper and lower fins. (b) Five coarse-scale SIFT descriptors (grayscale) extracted at part locations. (c) Two color histograms extracted from two convex hulls of subsets of parts. (d) 64 RGB color centers learned with k-means.

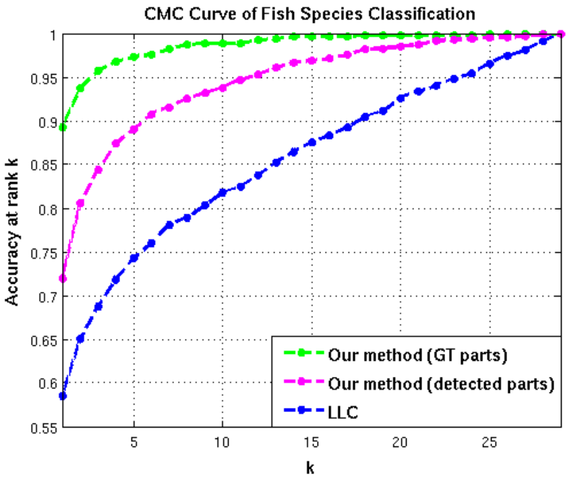
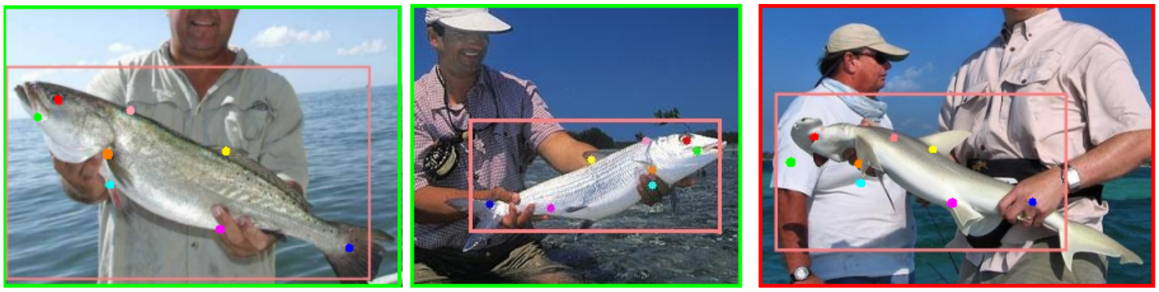


Figure 4.5: Cumulative Match Characteristic (CMC) curves for fish species classification.

such a challenging problem. Moreover, our method significantly outperforms a well-known image classification technique: LLC [Wang *et al.*, 2009], demonstrating again the benefit from parts. We



Top 5 guesses: 01.Spotted seatrout, 02.Gag grouper, 03.Red drum, 04.Tarpon, 05.Gray snapper

Top 5 guesses: 01.Bonefish, 02.Spotted seatrout, 03. Common snook 04.Tarpon, 05.Red drum

Top 5 guesses: 01.Bonefish, 02. Cobia, 03.Bonnethead shark, 04.Common snook, 05.Permit

Figure 4.6: Testing examples of fish species classification. Green words below the images indicate the correct labels. Success case is denoted with green frame, while failure case is denoted with red frame. Each image is overlaid with colored dots (*i.e.*, detected parts) and pink box (*i.e.*, object bounding box).

also show the upper bound of our classification method by using the ground-truth part locations. We observe a large gap between the accuracy of detected parts and ground-truth parts, presumably because the visual features are sensitive to the part locations. Some classification examples of our method are shown in Fig. 4.6.

4.2 Bird Species Classification

4.2.1 CUB-200-2011 Dataset

We also test our method on CUB-200-2011 [Wah *et al.*, 2011] dataset, which contains $11,788$ uncropped images of 200 bird species (about 60 images per species). We use the train/test split provided in the dataset for the experiments. There are roughly 30 images per species to train. It is a challenging dataset for both part localization and species classification as there are wide variations in the pose and appearance, as shown in Fig 4.7.

4.2.2 Bird Features

The part-based features are extracted in a similar way as the fish features. The features include grayscale SIFT and color histograms: we center 12 SIFT windows at the 15 parts (for symmetrical

Figure 4.7: Sample images from CUB-200-2011 bird dataset.

parts such as left/right eyes, we randomly choose one if both are visible), and the features for invisible parts are set zero. Based on the parts on the head and body, we construct two convex hulls respectively, and extract a color histogram from each region using 64 color bins.

4.2.3 Results

Please refer to Sec. 5.3.3 for the part localization results, where we compare with other methods of part localization. To demonstrate how the accuracy of part localization affects the classification, we feed the estimated part locations to our classification model, which is implemented as one vs. all SVMs with RBF kernels.

In Fig. 4.8, we plot the CMC curves showing the classification accuracy against ranked guesses. From the comparisons, we can see that the classification performance is consistently improved along with the increased accuracy of part localization. The upper bound of the classification accuracy is obtained by using the ground-truth part locations from human labelers.

There are other methods evaluated on this dataset, including Birdlets [Farrell *et al.*, 2011], Template Bagging [Yao *et al.*, 2012], and Pose Pooling [Zhang *et al.*, 2012]. We compare our method with them on the whole dataset as well as on a subset of 14 (Vireos and Woodpeckers) species in

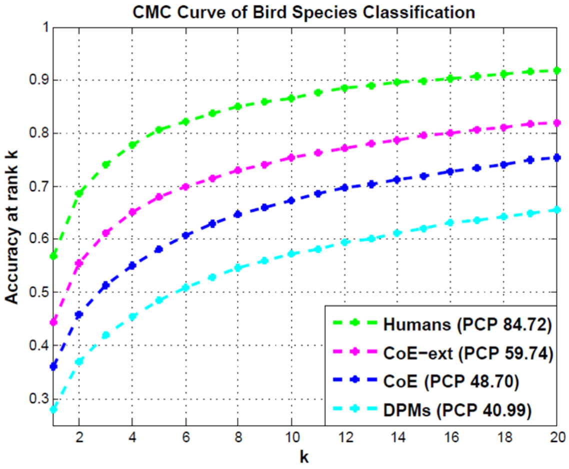


Figure 4.8: Cumulative Match Characteristic (CMC) curves for bird species classification. In the legend, we show the corresponding part localization methods as well as their localization accuracy. PCP means percentage of correct parts, please refer to Sec. 5.3.3 for more details.

Tab. 4.2. Although those methods may be more sophisticated in extracting the features or designing the classifiers, our method has much better results, which we believe should be attributed to the accurate part localization. Also note that we achieve the state-of-the-art performance on the dataset in a fully automatic setting (without using ground-truth bounding boxes or ground-truth part locations from the testing dataset). From the experiment, we do feel accurate part localization goes a long way towards building a working system for fine-grained classification.

Method	200 species	14 species
Birdlets	-	40.25%
Template Bagging	-	44.73%
Pose Pooling	28.18%	57.44%
Ours	**44.13%**	**62.42%**

Table 4.2: Mean average precision (mAP) on the full 200 categories as well as a subset of 14 categories from different classification methods. Birdlets and Template bagging are not directly comparable to ours as they use an earlier version of the dataset.

4.3 Discussion

Although part-based method seems very simple, it is really powerful in fine-grained classification, where the major burden is rested on the part localization. Part-based methods regain the attentions of vision community primarily due to the advance in object detection and part localization. We have used three examples of fine-grained classification to verify that parts benefit classification tasks that rely on local features with correspondences across instances. We may come up with more sophisticated feature descriptors or classification models, the effect of parts should still be discernible, as is shown by recent works that combine Deep Convolutional Neural Networks with parts for visual classification [Zhang *et al.*, 2014b; Branson *et al.*, 2014; Zhang *et al.*, 2014a].

Chapter 5

Exemplars with Enforced Part Consistency

In this chapter, we study the problem of part localization for fine-grained categories, where we attempt to localize the parts over the full object body. As full body is generally more deformable than near-frontal face, the part localizer in Sec. 3.2.2 haS difficulty handling the dramatically increased visual complexities. As a result, we observe a large gap between the localized parts and the ground-truth parts, which is also reflected in the fine-grained classification. We use bird part localization as the test case to design our new method. The experiments are conducted on a publicly available bird dataset [Wah *et al.*, 2011] that poses additional problems in contrast to our dog dataset: there are much wider variations in the part configuration due to different shapes, articulated deformation and unconstrained viewpoints; state-of-the-art object detector fails to achieve satisfactory accuracy in bird detection [1], making the strategy in Sec. 3.2 not applicable here.

To address the above problems, we bypass the object detection stage, and seek to build rich models for part appearance (*i.e.*, part detectors). We then apply these models under the framework of Consensus-of-Exemplars (CoE) approach, where we improve the hypothesis evaluation, such that geometrically correct hypotheses (generated from exemplars) have high scores. As the top-scoring hypotheses are concentrated in the image space and give a good estimation about the pose of testing

[1] We tried DPM detector [Felzenszwalb *et al.*, 2010b] on the bird dataset. Using the same criterion, the rank-1 accuracy is about 75% for birds, as opposed to 90% for dog faces.

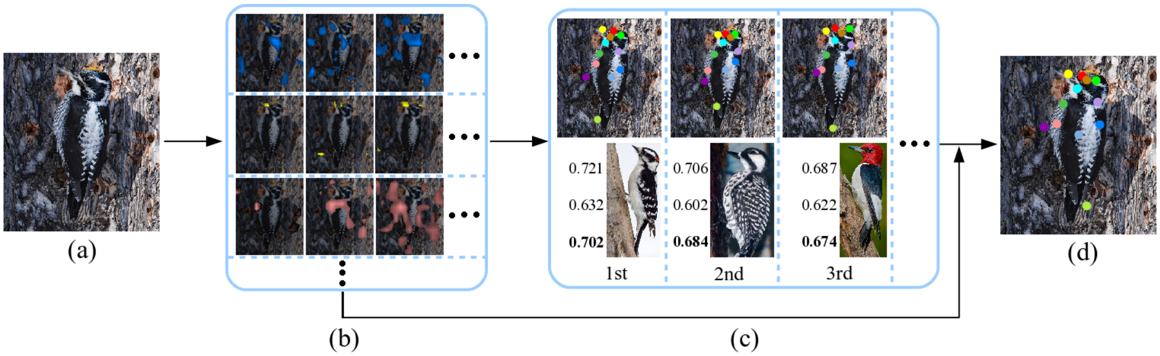


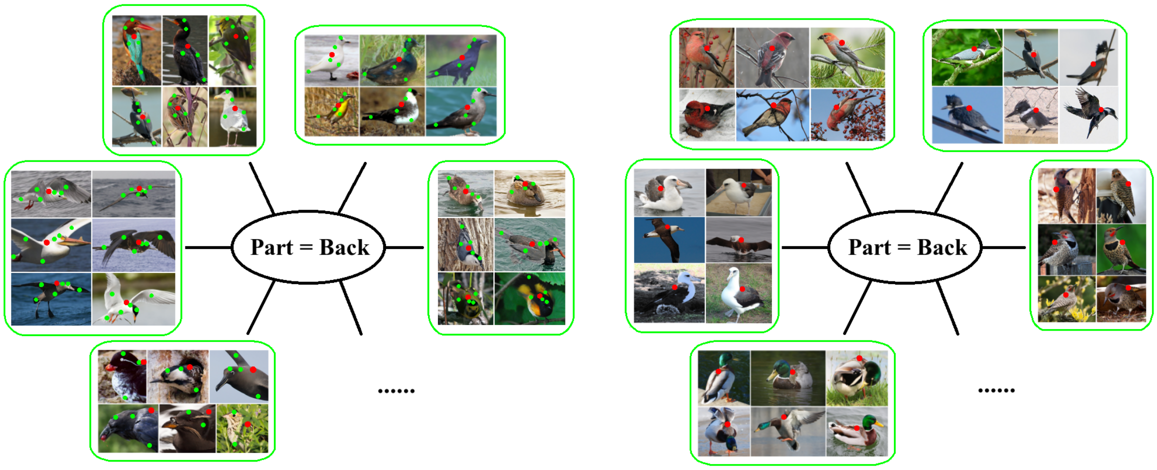
Figure 5.1: Overview of our part localization pipeline. (a) Given a test image, (b) an ensemble of part detectors are applied to the test image, generating detection response maps. (c) Exemplars are matched to the detection output, with their scores shown on the left (from top to bottom: pose consistency score, subcategory consistency score, overall score). The test image is overlaid with hypotheses at the top. (d) Consensus-of-Exemplars approach is employed to determine the final part locations.

sample, they are able to reach a more correct consensus in the final prediction.

Specifically, we decompose the visual complexity of parts by clustering their local configurations (referred to as part poses), as well as their subcategory labels. Appearance models are built for each cluster, yielding pose and subcategory detectors for each part. The clustering results also allow us to augment the exemplars: exemplars dictate not only the relative part locations, but also the local pose of the parts. In addition, we let each exemplar carry a unique yet unknown subcategory label. With these in hand, we enforce pose and subcategory consistency on the exemplars, and evaluate each generated hypothesis for the test image based on (1) how likely the corresponding part poses co-occur in the image, (2) how likely the image features implementing the parts belong to the same class. The overview of our localization method at the testing stage is shown in Fig. 5.1.

5.1 Pose and Subcategory Detectors

As the building block of our method, we build part detectors that score pose-specific and subcategory-specific features. To do this, we group part samples based on their poses and species labels.



(a) Pose clusters

(b) Subcategory clusters

Figure 5.2: Examples of pose clusters and subcategory clusters of the part Back (marked with a red dot in each image). In (a), the set of visible neighboring parts are marked with green dots. Note that the local part configurations within a cluster are very similar.

5.1.1 Pose Detector

We first group part samples with similar poses (defined as local part configuration). To represent the pose of a part, we turn to the keypoint annotations of its neighboring parts. Let X_k denote the configuration of the k-th exemplar, which is a vector containing the visibility flags $\{v_k^i\}$ and image locations $\{x_k^i\}$ where i is the part index. Given X_k, we can represent the pose of part i with a local offset vector $\triangle_k^i = [\triangle x_k^{i,j_1}, v_k^{j_1}, \ldots, \triangle x_k^{i,j_m}, v_k^{j_m}]$ where $j_1, \ldots, j_m$ are the indices of m ($m = 6$ in our experiment) predefined neighboring parts shared by all the exemplars. If part j is not visible, then $v_k^j = 0$, and $\triangle x_k^{i,j} = (0, 0)$; otherwise, $\triangle x_k^{i,j}$ is computed as $x_k^j - x_k^i$. To deal with size variations, $\triangle_k^i$ is normalized such that $\sum_j \|\triangle x_k^{i,j}\|^2 = \sum_j v_k^j$. The local offset vectors across all the samples form a pose space, whose subdivisions define the pose types of part i. For simplicity, we use k-means to generate N types for each part. Fig. 5.2 (a) shows several examples of pose clusters of the part Back.

For each pose cluster of each part, a detector is built by using the samples in that cluster as positive training data. A much larger set of negative samples are randomly drawn from image regions with little overlap with the target part. Therefore, by design, the detectors are trained to score the local poses across different subcategories.

5.1.2 Subcategory Detector

We group the part samples of birds from the same subcategory, assuming they share similar appearance in terms of color and texture. This assumption holds in our problem because the class labels are fine-grained. The number of clusters is fixed as we have a fixed number (which is 200) of bird species. Fig. 5.2 (b) shows several examples of subcategory clusters.

Similar to pose detectors, a subcategory detector is built for each cluster of each part. To make the subcategory detectors learn species-specific features, we do two things during the training: we first normalize the orientation of parts to reduce the noise in the features. The normalization is done by aligning each part sample to a reference part sample using Procrustes analysis with "reflection" enabled. The alignment is based on the local offset vectors defined in Sec. 5.1.1. Second, we run the pose detectors exhaustively on the training images, and collect false positives (which are off the correct part locations) as the negative training samples. Therefore, the subcategory detectors are able to learn subcategory-specific features across different poses.

5.1.3 Implementation Details

We use linear SVMs implemented in LIBSVM [Chang and Lin, 2011] to build pose and subcategory detectors. The features are HOG descriptors extracted using VLFeat toolbox [Vedaldi and Fulkerson, 2008]. The scale of a part is normalized by normalizing its local offset vector. After that, HOG descriptors are extracted from a window centered at that part, which contains 5×5 cells with bin size 8. During detection, we use a scaling factor of 1.2 to build image pyramid from the test image, and scan the images at each scale.

Because pose and subcategory detectors play different roles in our method (see Sec. 5.2), there are some differences in their features. For pose detectors, we extract two additional HOG descriptors at a coarser scale and a finer scale, which are two levels above and below the normalized scale in the image pyramid. For subcategory detectors, we extract three additional color histograms using 64 color bins, which are obtained through k-means in the RGB color space of the training images. These histograms are computed over three regions: an inner circle and two outer rings.

5.2 The Approach

Following [Belhumeur *et al.*, 2011], we cast the problem of part localization as fitting likely exemplars to an image, with the assumption that we can always find a configuration similar to the testing sample's from a sufficiently large training set. Recall that X_k is the k-th exemplar, which contains the visibilities and locations of all the parts. By using a similarity transformation t, we map X_k to the test image, obtaining an exemplar-based model $X_{k,t}$. Our goal is to estimate its conditional probability $P(X_{k,t}|I)$, which measures how likely the shape of X_k is present in the image at a certain location, scale, and orientation.

In [Belhumeur *et al.*, 2011], all the information of the image comes in the form of response maps D, and $P(X_{k,t}|I)$ is computed as

$$P(X_{k,t}|I) = P(X_{k,t}|D) = \prod_{i=1}^{n} P(x_{k,t}^i|d^i), \tag{5.1}$$

where n denotes the number of parts, $x_{k,t}^i$ is the image location of part i, and d^i is the corresponding response map. However, this formulation cannot be directly applied to our problem. First, it assumes there is a single detection response map for each part, while we have many response maps per part. These maps are from an ensemble of detectors applied over scales. Second, it does not address part visibilities, while there are 736 different combinations of visible parts in the dataset CUB-200-2011 [Wah *et al.*, 2011].

Besides addressing the above issues, our major contribution is enforcing pose and subcategory consistency on $X_{k,t}$ to obtain a more accurate estimation of $P(X_{k,t}|I)$.

5.2.1 Pose Consistency

One component of $P(X_{k,t}|I)$ is the score of pose consistency. To evaluate the score, we first generate a collection of response maps for all the parts × all the pose types, denoted as D_p. The key point is that for each exemplar X_k, we know the visibility of each part; if a part is visible, we also know its pose type. So when evaluating the likelihood $P(X_{k,t}|D_p)$, we choose the response maps corresponding to the pose types of X_k. With these in hand, we compute $P(X_{k,t}|D_p)$ as

$$P(X_{k,t}|D_p) = \left(\prod_{i,v_k^i=1}^{n} P(x_{k,t}^i|d_p^i[c_k^i, s_{k,t}^i]) \right)^{\frac{1}{\sum_i v_k^i}}, \tag{5.2}$$

where v_k^i denotes the visibility flag of part i, $d_p^i[c_k^i, s_{k,t}^i]$ denotes the response map for pose type c_k^i at scale $s_{k,t}^i$. $s_{k,t}^i$ can be determined based on the scaling factor in the transformation t and the scale level of part i in the original exemplar X_k. To get the probabilities in Eq. 5.2, each response map is converted to a probability map using the detector calibration method described in [Divvala *et al.*, 2012]. Because the exemplars usually cannot fit the configuration of a testing sample perfectly, the probability maps are smoothed before evaluating $P(X_{k,t}|D_p)$. For efficiency, we use an Max filter implemented by [Dollár, 2009]. The filter radius is estimated by measuring the distance between the corresponding parts of two globally similar exemplars after geometric alignment.

Because of the way $P(X_{k,t}|D_p)$ is computed, it is not plagued by false detections in other irrelevant response maps. Also, because of the reduced visual complexity in each pose cluster, a correctly chosen response map can give fairly accurate estimation of the part locations. For these reasons, the estimation of $P(X_{k,t}|D_p)$ is more reliable than $P(X_{k,t}|D)$ in [Belhumeur *et al.*, 2011]. From Eq. 5.2, we can see that given the response maps, the cost of subsequent computations (*i.e.*, evaluating a fixed number of $X_{k,t}$'s) is independent of the number of pose types, as opposed to [Yang and Ramanan, 2011; Zhu and Ramanan, 2012]. Therefore, we can increase the number of pose types without affecting the inference speed much.

5.2.2 Subcategory Consistency

Subcategory Consistency means that the appearance at all the parts should agree with each other on the class membership. Here, we assume that the image cues are contained in D_s, a collection of response maps for all the parts × all the subcategories. Given a subcategory label l, we evaluate the likelihood of the image region at $X_{k,t}$ containing an object from this subcategory as

$$P(X_{k,t}|l, D_s) = \left(\prod_{i,v_k^i=1}^{n} P(x_{k,t}^i | d_s^i[l, s_{k,t}^i, \theta_{k,t}^i]) \right)^{\frac{1}{\sum_i v_k^i}}, \tag{5.3}$$

where $d_s^i[l, s_{k,t}^i, \theta_{k,t}^i]$ denotes the response map for part i of subcategory l, at scale $s_{k,t}^i$ and orientation $\theta_{k,t}^i$ (the subcategory detectors are rotation invariant). $\theta_{k,t}^i$ can be computed based on the rotation angle in the transformation t and the original orientation of X_k's part i. We use the same method as pose detector calibration to convert the response maps to probability maps. After computing $P(X_{k,t}|l, D_s)$ for all possible l's, the second component of $P(X_{k,t}|I)$ is defined as

$$P(X_{k,t}|D_s) = \max_{l} P(X_{k,t}|l, D_s). \tag{5.4}$$

5.2.3 Generating Hypotheses

After evaluating $X_{k,t}$'s pose and subcategory consistency, we evaluate its overall score as

$$P(X_{k,t}|I) = P(X_{k,t}|D_p)^{\alpha} P(X_{k,t}|D_s)^{(1-\alpha)}, \tag{5.5}$$

where $\alpha \in [0, 1]$ controls the weights of $P(X_{k,t}|D_p)$ and $P(X_{k,t}|D_s)$. α is determined through cross-validation, and $\alpha = 0.8$ works best in our experiment. Our goal here is to generate and select the highest scoring $X_{k,t}$'s.

Because applying the subcategory detectors in a sliding-window fashion is very expensive (they need to search over scales and orientations), but not necessary (they are built on top of the activations of pose detectors), we only generate the response maps for pose detectors, and construct a random transformation t for each X_k as follows:

- Randomly choose two parts and a scaling factor.
- Select the two response maps of the two parts at the corresponding scales.
- Randomly choose a local maxima from each map as mode.
- Compute similarity transformation t that maps the two parts of X_k to the two modes.
- If the scaling factor or rotation angle in t is beyond a predefined range, t is discarded.

By repeating the above procedure multiple times for each exemplar, we generate a large set of models $\{X_{k,t}\}$, whose conditional probabilities are computed using Eq. 5.5. An illustration is shown in Fig. 5.3. The top scoring models then constitute the set of likely hypotheses.

Although subcategories detectors are only applied to the generated $\{X_{k,t}\}$, it is still very expensive to extract the features due to the large number of $X_{k,t}$'s (about $380,000$ in our experiment). Therefore, we approximate the procedure by computing $P(X_{k,t}|D_p)$ for all the models first (which is relatively much faster), and then keeping the top ranked models (*e.g.*, 400) which will be re-ranked after incorporating $P(X_{k,t}|D_s)$. We observe that the performance is not hurt by this approximation as $P(X_{k,t}|D_p)$ already gives a fairly accurate estimation of the correctness of each $X_{k,t}$.

As the models usually cannot match the testing sample perfectly, we also need to address the issue when evaluating $P(X_{k,t}|D_s)$. Because we extract the features at the part locations dictated

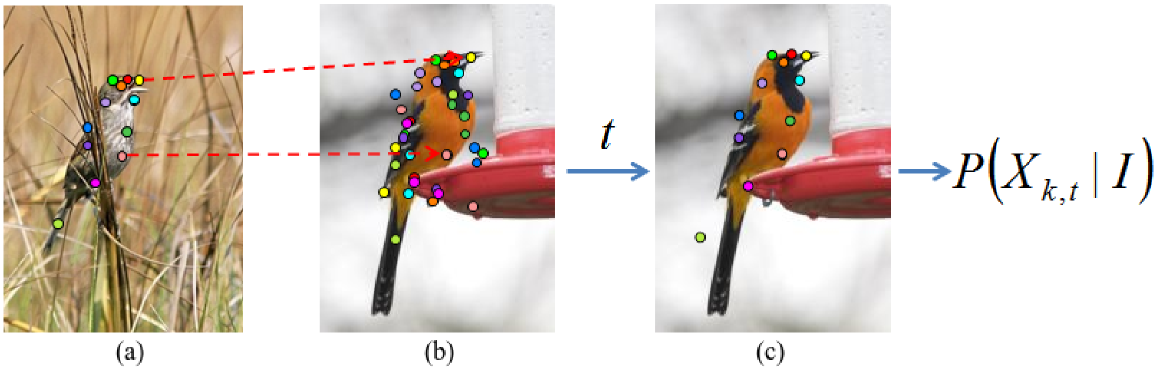


Figure 5.3: An example of matching an exemplar to the test image. (a) An exemplar X_k and its annotated parts. (b) The test image I overlaid with detection modes for each part. (c) With similarity transformation t, exemplar X_k is mapped to the test image as a hypothesis $X_{k,t}$ which is assigned a probability score.

by the models, small errors in the part locations may lead to severe underestimation of $P(X_{k,t}|D_s)$. Therefore, we adopt a group-based re-ranking strategy. Given the ranked list of 400 models based on $P(X_{k,t}|D_p)$, we group them when obtaining the subcategory scores for their parts. More specifically, for part i, we successively pick a model from the ranked list, find and remove other remaining models from the list which have similar local offset vectors as the initially picked one. The similarity is quantified by the sum of the squared distances (SSD) between the corresponding parts. These models form a group where the maximum $P(x^i_{k,t}|d^i_s[l, s^i_{k,t}, \theta^i_{k,t}])$ in Eq. 5.3 is shared by models in the group. Such approximation is not ideal, but it works in the case where underestimation of $P(X_{k,t}|D_s)$ has larger negative effect than tolerable errors in the top-ranked part locations. Also note that the consensus stage in Sec. 5.2.4 can reduce the errors through Gaussian smoothing.

5.2.4 Predicting Part Configuration

After ranking the hypotheses based on $P(X_{k,t}|I)$, we keep M ($M = 40$) highest-scoring ones with indices $\{k_m\}_{m=1,\ldots,M}$. These hypotheses need to reach a consensus on the part locations. We first predict the visibility flag v^i for each part i through voting:

$$v^i = \begin{cases} 1 & : \quad \sum_m v^i_{k_m} > \tau M \\ 0 & : \quad \text{Otherwise} \end{cases}, \tag{5.6}$$

where threshold τ is determined through cross-validation, such that the False Invisibility Rate defined in Sec. 5.3.1 is comparable with that of human annotators (about 6%). If part i is predicted as visible, we use a modified version of Eq. 3.3 to estimate its location p^i by combining the hypotheses and the probability maps of the corresponding pose types:

$$\hat{p^i} = \arg\max_{p^i} \sum_{k,t \in \mathcal{M}} P(\Delta x^i_{k,t}) P(p^i | d^i_p[c^i_k, s^i_{k,t}]), \tag{5.7}$$

where $\mathcal{M}$ denotes the set of top-M hypotheses.

As can be seen here, the pose detectors play an important role in finding the parts while the subcategory detectors focus on verifying the hypotheses supported by the pose detectors.

5.3 Experiments and Results

We evaluate our method extensively on CUB-200-2011 [Wah *et al.*, 2011] dataset, which contains $11,788$ uncropped images of 200 bird species (about 60 images per species). We use the train/test split provided in the dataset for all the experiments. There are roughly 30 images per species to train, and we do left-right flipping to increase the size of training data. A total of 15 parts were annotated by pixel location and visibility flag in each image through Amazon Mechanical Turk (MTurk).

5.3.1 Evaluation Metrics

To gain a thorough view of our method, we use four metrics to evaluate the localization performance: Percentage of Correctly estimated Parts (PCP), Average Error (AE), False Visibility Rate (FVR), and False Invisibility Rate (FIR). “Correct estimation” means the detected part is within 1.5 standard deviation of an MTurk user’s click if visible or the part is correctly estimated as invisible. “Average error” is computed by averaging the distance between predicted part locations and ground truth (if both are visible), normalized on a per-part basis by the standard deviation and bounded at 5. “False Visibility Rate” is the percentage of parts that are incorrectly estimated as visible. “False Invisibility Rate” is the percentage of parts that are incorrectly estimated as invisible. Note that AE best indicates the localization precision.

5.3.2 The Number of Pose Types

To determine the number of pose types, we only consider pose consistency in this experiment (*i.e.*, set $\alpha = 1$ in Eq. 5.5). As shown in Tab. 5.1, we change the number of types for each part from 1 to $2,000$. The pose detectors are implemented with linear SVM except for the extreme case where we only have one type. Due to the large visual complexity in this case, we use non-linear SVM with RBF kernel to build the detector, which is the same as [Belhumeur *et al.*, 2011]. From the comparisons, we can see that with roughly the same FIR (which can be adjusted by the parameter τ in Sec. 5.2.4), the performance measures of PCP, AE and FVR are consistently improved as the number of types increases up to 500. To explain this, on the one hand, the larger the number of types, the more the visual complexity can be reduced. On the other hand, finer granularity of pose types makes the constraints on pose consistency stronger. As the number goes beyond $1,000$, the result becomes slightly worse, possibly due to that there are much fewer positive training samples. Given more than 50 pose types, our method already outperforms [Belhumeur *et al.*, 2011]. We choose 200 types in subsequent experiments as it is a good trade-off between accuracy and speed ($1.5\times$ faster than 500 types and $2.6\times$ faster than $1,000$ types).

To verify that an ensemble of linear detectors alone do not contribute to the performance improvement, we relax the pose consistency by collapsing the output of all 200 pose detectors to a single response map for each part using pixel-wise maximum. Now it is equivalent to the case where we have only one pose type. However, the accuracy drops a lot, with 47.08% PCP, 2.30 AE, 39.36% FVR, and 7.12% FIR. It implies that enforcing pose consistency is critical in our method.

We also try an alternative method to generate the part types. [Divvala *et al.*, 2012] uses Latent-SVM learning to optimize the ensemble of detectors, leading to appearance-based clustering. As the visual appearance is coupled with pose, [Divvala *et al.*, 2012] actually groups samples similar in pose but with more noise than our pose clustering. From the comparisons in Tab. 5.1, we can see that clustering by geometry is better at decomposing the visual complexity and is a better fit for our pose consistency evaluation.

5.3.3 Part Localization

We compare our work with three state-of-the-art techniques: Poselets [Bourdev *et al.*, 2010], Deformable Part Models (DPM) [Branson *et al.*, 2011], and Consensus of Exemplars [Belhumeur *et*

# of Pose Types	PCP	AE	FVR	FIR
1	48.70%	2.13	43.90%	6.72%
10 (Pose)	45.79%	2.37	44.21%	4.14%
50 (Pose)	53.07%	2.08	34.02%	4.40%
100 (Pose)	54.66%	2.00	31.21%	4.87%
200 (Pose)	56.88%	1.92	**30.16%**	4.32%
500 (Pose)	**57.03%**	**1.91**	30.21%	4.34%
1,000 (Pose)	56.63%	1.94	31.26%	**3.91%**
2,000 (Pose)	56.50%	1.97	32.35%	4.08%
10 (App.)	43.30%	2.55	42.10%	**4.48%**
50 (App.)	48.86%	2.29	32.55%	6.43%
100 (App.)	51.05%	2.20	32.01%	5.97%
200 (App.)	**52.10%**	**2.15**	**31.30%**	5.65%
500 (App.)	52.00%	2.17	31.57%	5.71%
1,000 (App.)	51.32%	2.21	32.27%	5.46%
2,000 (App.)	51.07%	2.26	32.31%	5.58%

Table 5.1: Part localization results using different numbers of pose types. The best performance is achieved with 500 pose types for each part. Appearance-based clustering can also be used to generate pose types, which is inferior to ours in terms of the performance. Please refer to Sec. 5.3.1 for the meaning of each metric.

al., 2011]. For Poselets-based part localization, we obtain the Poselet activations from the authors of [Zhang *et al.*, 2012], and predict the location of each part as the average prediction from its corresponding Poselet activations. For DPM, we obtain the localization results from the authors. Note that [Branson *et al.*, 2011] only detects 13 parts, omitting the two legs. We implemented and modified original Consensus-of-Exemplars approach to deal with part visibility. Without considering visibility, we will get almost 100% FVR and zero FIR, making the results not comparable. Also note that larger FVR generally leads to larger AE.

As shown in Tab. 5.2, our part localization outperforms state-of-the-art techniques on all the

Method	PCP	AE	FVR	FIR
Poselets	27.47%	2.89	47.90%	17.15%
DPM	40.99%	2.65	32.62%	6.18%
CoE	48.70%	2.13	43.90%	6.72%
Ours (CoE-ext)	**59.74%**	**1.80**	**28.48%**	**4.52%**
Human	84.72%	1.00	20.72%	6.03%

Table 5.2: Part localization results from different methods. Our method significantly outperforms state-of-the-art techniques on all the four metrics.

metrics. The large error rate of Poselets agrees with the fact that by design, they do not target localizing the individual parts with high accuracy. Compared with the results in Tab. 5.1, our full model achieves remarkable improvement on AE by incorporating the subcategory consistency. In a separate experiment, we set $\alpha = 0$ in Eq. 5.5, and obtain 58.28% for PCP, 1.86 for AE, 28.88% for FVR, and 5.32% for FIR. It indicates that pose consistency and subcategory consistency are complementary to each other. For bird species classification, incorporating subcategory consistency when detecting the parts also leads to 3% increase in the rank-1 accuracy.

Assuming almost all the parts are visible (by setting $\tau = 0$ in Eq. 5.6), our full model obtains 54.36% for PCP, 1.85 for AE, 60.03% for FVR, and 0.28% for FIR, which are not much worse except for FVR. Some examples of our bird part localization are shown in Fig. 5.4. Although birds have very wide variations in appearance and pose, and birds reside in very different environments, our method is still able to detect most of the parts correctly.

5.3.4 Application: BirdSnap

As our fine-grained system has shown impressive results on bird species classification (Sec. 4.2) using our bird part localizers, we expect to see further improvement by replacing the classification module with a more sophisticated part-based method: POOF [Berg and Belhumeur, 2013]. To this end, we build another application BirdSnap which serves as a field guide for bird species in North America. Moreover, it has the feature of visual recognition which identifies the bird species in user-

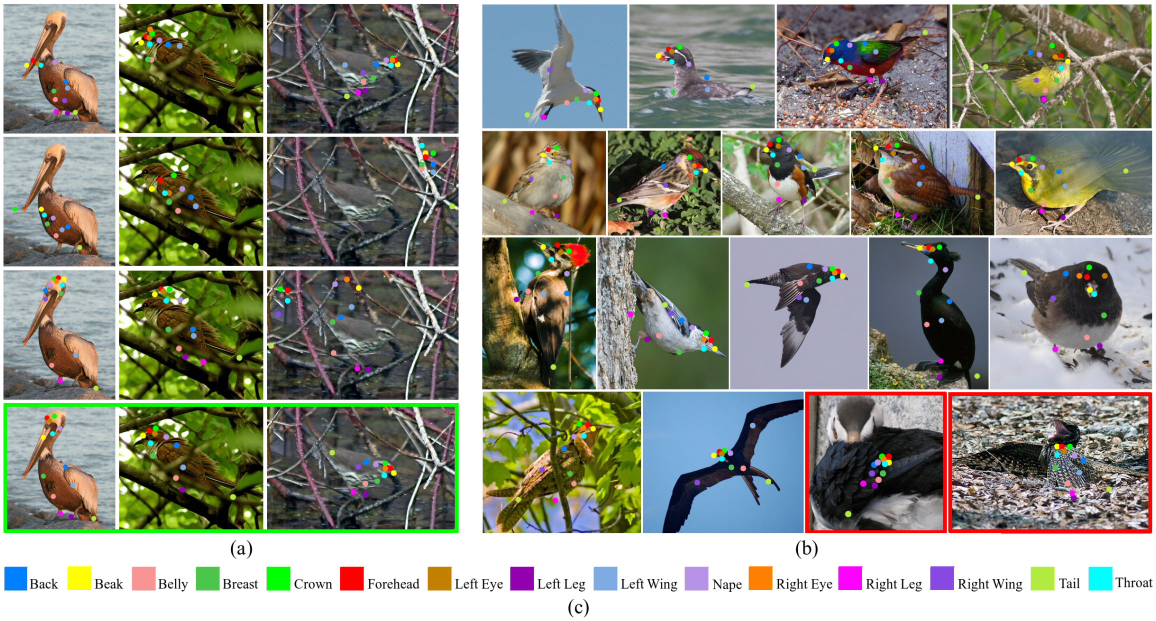


Figure 5.4: Examples of bird part localization. (a) compares the four methods (from top to bottom: Poselets, DPM, Consensus of Exemplars, Our method) on three testing samples. (b) gives more examples of part localization with our method. Red frames denote the failure cases. (c) shows the color codes for the 15 parts.

uploaded images. Both the iPhone App and website [2] of BirdSnap have been released. Screenshots of our iOS App are shown In Fig. 5.5.

5.4 Discussion

Compared with frontal or near-frontal dog faces, birds have much larger variations in pose and appearance. As the original Consensus-of-Exemplars approach does not handle the variations well, the performance of its part localizer is not satisfactory. Therefore, we approach the problem by significantly reducing the visual complexities and building an ensemble of part detectors. More importantly, we take advantage of these detectors to enforce pose and subcategory consistency on exemplar-based models. As a result, we generate more reliable hypotheses of part configuration, which better exploits the consensus module. We also show how to efficiently generate the hypothe-

[2] URL is http://birdsnap.com/

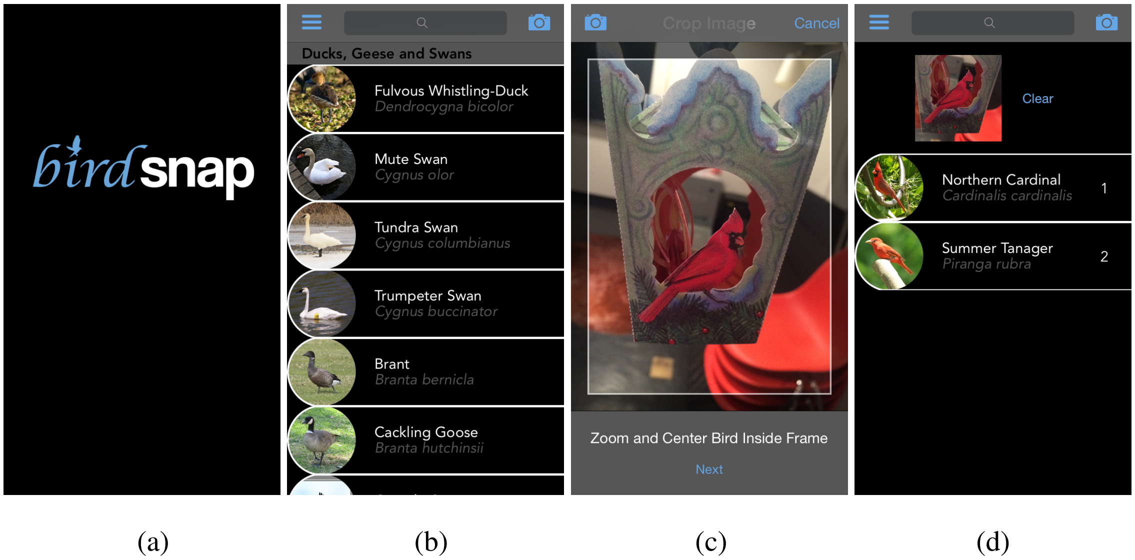


Figure 5.5: Screenshots of our iPhone app. (a) Home screen. (b) Browse screen with the bird species. (c) Bird camera. (e) Classification results.

ses using a heuristic re-ranking strategy. The improved quality in the hypotheses over [Belhumeur *et al.*, 2011] is the key to the large-margin improvement on the localization accuracy. Experimental results demonstrate that our method achieves state-of-the-art performance for part localization on the challenging bird dataset CUB-200-2011.

Chapter 6

Exemplars under Part-pair Representation

In previous chapter, we have shown that combining exemplars with rich appearance models improves the quality of top-scoring hypotheses, thus benefiting the Consensus-of-Exemplars approach. However, it has additional limitations: first, it needs fine-grained class labels to build subcategory detectors, limiting the generalization ability of the method; second, as an exemplar imposes constrains on all the parts, it can only applied to testing samples with very similar configurations. In other words, we need a sufficiently large set of diverse training exemplars to densely cover the configuration space (which may not be possible even for medium-scale datasets); third, the consensus stage is likely to fail when top-scoring hypotheses differ a lot in the part locations.

These limitations motivate us to find a better way to represent objects, such that we can obtain a rich set of appearance models without using fine-grained class labels, and we can compose novel configurations with existing exemplars, improving the coverage of plausible poses. As for the third limitation described above, we seek to get rid of consensus by generating per-part hypotheses, each of which has high localization accuracy for a target part. To this end, we propose a novel part-pair representation to model the configuration and appearance of an object.

With the part-pair representation, we break an object into part pairs and model the appearance and geometry of the object based on the part pairs. Specifically, we build detectors for each pair, thus obtaining a rich set of appearance models; we customize the strength of spatial constraints on parts by considering subsets of part pairs. The pipeline of part localization follows a bottom-up paradigm: (1) we first build context-aware part detectors by combining exemplar-based models with pair detectors; (2) then we generate part hypotheses from these part detectors; (3) finally, instead of reaching a consensus from multiple hypotheses, we explicitly compose promising configurations, where the highest-scoring one is expected to best fit the testing sample. We evaluate our method extensively on bird part localization and human pose estimation, where we achieve significant improvement over previous CoE-based methods.

6.1 Part-pair Representation

Unlike part-based models that treat an object as a collection of parts, part-pair representation breaks down the object into part pairs, which form a complete graph connecting any two parts. Under such representation, the modeling of shape and appearance is based on the part pairs.

6.1.1 Shape Modeling

Assuming an object X has n parts and x^i denotes the location of part i, then part-pair representation treats X as a set of $n(n-1)/2$ part pairs $\{(x^i, x^j)|i, j \in [1, n], i \neq j\}$. For each pair (i, j) of X, we record its center location $c^{i,j}$, orientation $\theta^{i,j}$, and length $l^{i,j}$. As any set of $n-1$ pairs that cover all the parts can reconstruct the global part configuration, most pairs seem to be redundant. In practice, such redundancy allows us to adjust the strength of geometric constraints as needed, which will be addressed in Sec. 6.2 and Sec. 6.3.

6.1.2 Appearance Modeling

We build pair detectors to model the appearance of each pair. They can be seen as specialized Poselet detectors [Bourdev *et al.*, 2010], targeting two parts simultaneously. These detectors cover different portion of an object at different scales, with possibly large overlap. For this reason, we have a rich representation of the object appearance.

Mixtures of Pair Detectors. To deal with rotation variations of the pairs, we discretize the rotation space in 15 different bins, with each bin corresponding to a span of 24 degrees. We then build

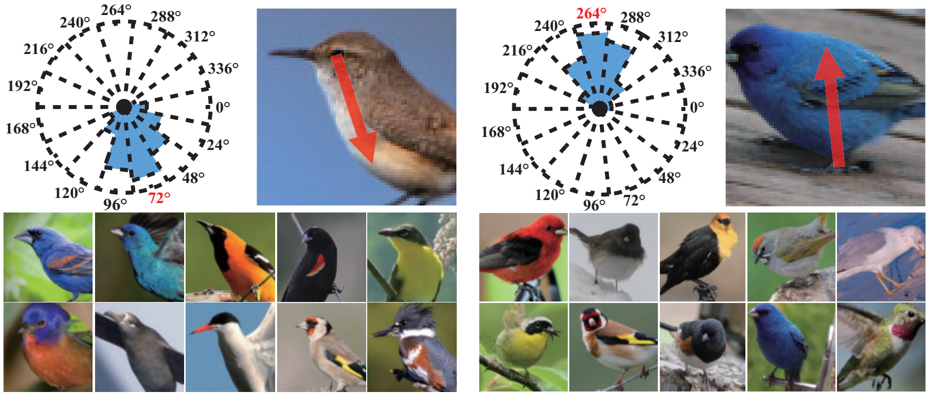


Figure 6.1: Normalized training samples. The left figure is for the pair (Left Eye, Belly), and the right figure corresponds to (Left Leg, Back). In each figure, sample frequencies over 15 orientations are visualized as blue sectors in the pie chart. The red arrow superimposed over the sample image indicates the target pair orientation.

one detector for each pair in each orientation[1]. For efficiency, we measure the sample frequencies in each orientation bin, and ignore the bins with frequencies lower than 1%. By doing so, we have 776 rather than $1,575$ detectors altogether for the bird dataset [Wah *et al.*, 2011] where the number of parts is 15.

Inspired by POOF [Berg and Belhumeur, 2013], we normalize the samples for each pair detector by rotating and rescaling them, so that they are aligned at the two corresponding parts. Please see Fig. 6.1 for some normalized examples. For rotation, the rotation angle is determined based on the difference between the original pair orientation and the center of the target orientation bin. For rescaling, we rescale the samples to predefined sizes, with care taken to handle very diverse pairs. For example, (Eye, Tail) pair is typically ten times larger than (Eye, Forehead) pair in an image, which entails different reference sizes for rescaling.

To automate the process of selecting the reference sizes, we first estimate the average length $\bar{l}$ for each pair from the training data. After that, we know the minimum and maximum average lengths $\bar{l}_{min}$ and $\bar{l}_{max}$ across all the pairs. Assuming that the normalized length is in the range $[\hat{l}_{min}, \hat{l}_{max}]$, we use a linear function $f(l)$ to map range $[\bar{l}_{min}, \bar{l}_{max}]$ to $[\hat{l}_{min}, \hat{l}_{max}]$. Therefore, the reference size

[1] We use non-linear detector to handle pose & appearance variations of samples in the same orientation bin.

for pair (i, j) is $f(\bar{l}^{i,j})$. We empirically set $\hat{l}_{min} = 24$ and $\hat{l}_{max} = 52$ to ensure reasonable quality of training samples.

Training and Testing. After normalization, we use toolbox [Dollár, 2009] to extract the first-order integral channel features within a bounding box (*i.e.*, feature window) that contains both parts inside. Note that the feature window is placed at the center of the target pair. We randomly generate up to $2,000$ rectangles to compute the integral channel features, and follow [Dollár *et al.*, 2012] to build a soft cascade detector with constant rejection thresholds.

A cascade has $T = mT_0$ weak classifiers, and each weak classifier is a depth-two decision tree. $m = 30$ is the number of rounds of bootstrapping. After each round, we mine up to 400 hard negatives, and increase the number of weak classifiers by $T_0 = 50$ to build an AdaBoost classifier. Instead of performing a rejection test at every weak classifier, we do it after building additional T_0 weak classifiers (to accumulate enough observations). Assuming the score of a sample s up to the kT_0-th weak classifier is $H_k(s) = \sum_{j \le kT_0} \alpha_j h_j(s)$ where $\alpha_j > 0$ and $h_j(s)$ is the binary output of the j-th weak classifier, then the threshold can be set as $\tau_k = b \sum_{j \le kT_0} \alpha_j$ ($b = 0.45$ in our experiment).

At the testing stage, we build an image pyramid with six scales per octave, and apply the pair detectors in a sliding-window fashion (with stride 4 pixels). To facilitate the subsequent procedures, we normalize the scores so that an early rejected sample will not be penalized too much. To do this, we use $\hat{H}_k(s) = \frac{H_k(s)}{\sum_{j \le kT_0} \alpha_j}$, so the normalized score $\hat{H}_k(s)$ is within the range $[0, 1]$ (early rejected samples will have scores below 0.45). Note that we do not apply Non-Maximum Suppression to the detection results; instead, we cache them as response maps for each pair detector at each scale.

6.2 Super Part Detector

Our method of part localization follows a bottom-up paradigm, and a very important step is to generate reliable estimation for each part. In detection and localization problems, the output from a single detector is generally very noisy. However, if the outputs from multiple related detectors are pooled together, the noise can be significantly suppressed. Part-pair representation has such capability when employing exemplar-based models. The idea is that there are multiple pairs sharing the same part, and exemplars specify which detectors should be used for the pairs. In previous CoE approaches, the basic element of an exemplar is part, and exemplars are used to dictate plausible global configuration of parts. In this work, however, the basic element of an exemplar is part pair, and exemplars dictate how the pairs constitute an object.

6.2.1 Part Response Maps

Given the detection response maps from the pair detectors, our goal is to generate the response maps for each part. The idea is similar to Hough Voting: a part pair activation votes for the positions of its related parts. Our method differs in that exemplars specify which orientations, and scales should be used for the voting. Assuming X_k is an exemplar being scaled to a certain size, we can obtain the response map for part i based on X_k as follows.

Let $R^{i,j}(x)$ denote the set of all the response maps for pair (i, j) where x is the pixel location in the test image, then exemplar X_k specifies a particular response map to use (at certain scale and orientation), which is denoted as $R_k^{i,j}(x)$. To get the response map for part i from $R_k^{i,j}(x)$, we simply shift $R_k^{i,j}(x)$ as illustrated in Fig. 6.2 (b):

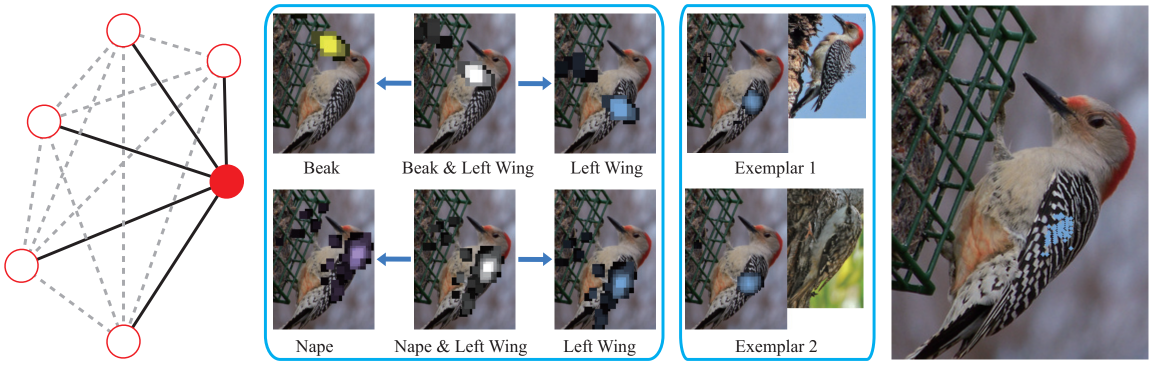
$$r_k^{i,j}(x) = R_k^{i,j}(x+o), \quad o = c_k^{i,j} - x_k^i, \tag{6.1}$$

where $r_k^{i,j}(x)$ is the resultant map; o is the offset between $c_k^{i,j}$ – the center location of pair (i, j), and x_k^i – the location of part i. In our implementation, we quantize the offset based on the discretization of pair rotations (please refer to Sec. 6.1.2). During testing, we cache the shifted response maps using the quantized offsets at each possible scale. Then, given an exemplar X_k, we can directly retrieve its corresponding map $r_k^{i,j}(x)$.

As exemplar X_k dictates all the visible pairs (*i.e.*, both parts of the pair should be visible) sharing part i, the part response map for part i is then estimated as

$$R_k^i(x) = \frac{1}{N_k^i} \sum_j r_k^{i,j}(x), \tag{6.2}$$

where N_k^i is the number of visible pairs sharing part i in X_k. Assuming there is a detector that directly generates such response map, then we name it as Super Part Detector, which is conditioned on a particular exemplar. Fig. 6.2 (c) shows two part response maps based on two exemplars.



(a) Part-pair graph (b) Shifted pair response maps (c) Part response maps (d) Candidate Detections

Figure 6.2: (a) shows the part-pair representation as a complete graph for an object with 6 parts. To build the super part detector for a part (solid circle), only the pairs sharing the part are used (solid lines). (b) illustrates the shifting of pair response maps. (c) shows the response maps for Left Wing conditioned on two exemplars. (d) shows the candidate detections of Left Wing.

6.2.2 Part Hypotheses

In this section, we will explain how to generate part hypotheses from the part response maps. As described in Sec. 6.2.1, different exemplars correspond to different super part detectors of part i. However, only the detectors from exemplars that match the testing sample at part i are needed (By "match at a part", we mean that the exemplar has similar configuration of parts in the neighborhood of the target part). Such detectors generally have high scores when firing at correct locations. Therefore, finding such detectors is equivalent to finding geometrically correct exemplars being placed at the correct locations.

A reasonable indicator about the goodness of an exemplar is the peak value of its corresponding part response map. Therefore, for part i, the score of X_k is computed as

$$S_k^i = \max_x R_k^i(x). \tag{6.3}$$

To search for good exemplars, a naive way is to go through all the training exemplars, rescale them to each possible scale, evaluate their scores with Eq. 6.3 and keep the top-scoring ones. This process can be made faster with a heuristic strategy: we estimate the upper bound of S_k^i at very low cost, and obtain an initial set of promising exemplars (*e.g.*, a few thousand). Then we use Eq. 6.3 to recompute their scores. The upper bound of S_k^i is computed as $\frac{1}{N_k^i}\sum_j \max_x r_k^{i,j}(x)$, where the addends can be reused to evaluate other upper bounds. In our experiment, we take the best 200 exemplars for each part, and extract up to five local maximas from each of their corresponding part response maps. The locations of the overall top-scoring 200 maximas then form the candidate part detections, as shown in Fig. 6.2 (d).

We have a by-product from the above procedure. As the candidate part detection indicates the image location to place exemplar X_k, we can also obtain the locations of other parts in the image. For instance, given X_k and image location x_0 for part i, the location of part j is estimated as $x_0 - x_k^i + x_k^j$. Therefore, we treat X_k and x_0 together as a part hypothesis. Fig. 6.3 shows several examples of part hypotheses, including both the target part and the inferred parts.

6.2.3 Analysis

The super part detector demonstrates one way of applying the part-pair representation, where a subset of up to $n - 1$ pairs are used to impose the geometric constraint (please see Fig. 6.2 (a)). As the subset of pairs form a star graph with the target part at the root, we call the target part as root part, and all the other parts as leaf parts. Because all the leaf parts are involved in building the super part detector of the root part, we achieve context-aware part detections, where the noise from the raw pair detectors is largely suppressed. Moreover, the quality of a super part detector is not sensitive to the displacement of distant leaf parts, as such displacement has limited effect on the pair feature (and thus the pair detection) due to generally large rescaling factors. Therefore, the strength of spatial constraints on the whole object is weaker than that in [Belhumeur *et al.*, 2011; Liu and Belhumeur, 2013]. In other words, exemplars that do not match the testing sample globally can still be used to localize a particular part. Such property makes the super part detector applicable in the case of insufficient training exemplars.

6.3 Predict Part Configuration

Recall that in Sec. 6.2.2, we obtain a set of hypotheses for each part. Each hypothesis consists of the part location, as well as the corresponding exemplar. To predict the global part configuration, we can apply the Consensus-of-Exemplars (CoE) approach using the part hypotheses. We also design an alternative approach to explicitly compose the global shape from the hypotheses.

6.3.1 CoE Approach

The idea is similar to [Liu and Belhumeur, 2013]: assuming we have exemplar X_k centered at position x_0 in the testing image, then we evaluate its overall score as

$$S_k = \frac{1}{N_k} \sum_{i,j} R_k^{i,j}(c_k^{i,j} + x_0) + \alpha \mathbf{bin}(V_k), \tag{6.4}$$

where N_k is the number of visible pairs, and $c_k^{i,j}$ is the relative location of pair (i, j) w.r.t the location of X_k. $\mathbf{bin}(\cdot)$ is the prior about the number of visible parts (denoted as V_k), and α is the weight parameter.

To predict the global configuration, we evaluate the overall scores of all the candidate exemplars (*i.e.*, the exemplars placed at the corresponding candidate part locations). Once we obtain the best 30 exemplars, we follow [Liu and Belhumeur, 2013] to predict the visibilities and locations of all the parts. In this case, the geometric constraint on the parts is much stronger than that for super part detectors, as all the pairs are involved in the evaluation, with each additional pair further imposing the constraints on its two end parts. Therefore, our part-pair representation is also applicable to the estimation of global configuration.

However, CoE approach is likely to fail if there are not enough training exemplars that are geometrically similar to the testing sample. In addition, CoE does not exploit the property of part hypotheses – having relatively more accurate estimation for the root parts than for the leaf parts. In other words, all the parts from an exemplar had better not be treated equally. Motivated by such observation, we attempt to explicitly compose the shape that fits the testing sample using multiple hypotheses of different parts.

6.3.2 Shape Composing

We aim to compose novel shapes (*i.e.*, part configuration) from the part hypotheses. Each visible part in the composed configuration should come from exactly one part hypothesis. The evaluation of these configurations is also based on Eq. 6.4, and only the highest scoring one is output as the prediction. As Eq. 6.4 generally produces high scores for correct and near-correct configurations, what we need is to compose geometrically correct configurations. To make the composing procedure tractable, we only employ a few part hypotheses (15 in our experiment) for each part. Tab. 6.1

shows that the top-scoring hypotheses already have very high accuracy, thus making such "list shortening" acceptable. Our shape composing follows two rules: (1) only consistent part hypotheses can be used to compose a shape; (2) shape composing should respect the property of part hypotheses as mentioned above. Before we describe the composing procedure, we first define consistent part hypotheses.

Consistent Part Hypotheses. To determine if two part hypotheses are consistent, we introduce the concept of uncertainty region (UR) for each part in a part hypothesis. Assuming we have a hypothesis with root part i, then the uncertainty region for part j is a circle with radius equal to a fraction (20% in our experiment) of the distance between part i and j (please see Fig. 6.3). If $i == j$, then the UR has zero radius. Given this definition, we claim that two part hypotheses rooted at two different parts agree on a part if the two corresponding URs for this part are close enough. The distance is measured as the distance between the two URs' centers divided by the larger radius. We require that two consistent hypotheses should agree on at least N parts, one of which must be a root part. In this way, the strictness of consistency is parametrized by N. In Fig. 6.3, we show two composing examples which use three consistent part hypotheses.

Composing Procedure. We design a procedure to progressively group hypotheses that are consistent. Although being heuristic, the method is capable of generating a good number of plausible shapes at reasonably low cost. (1) We start from a group with a single part hypothesis, then successively add another part hypothesis (sampled at random) that is consistent with more than a fraction r of current hypotheses. Note that before adding new part hypothesis, we copy and cache the current group. When the process terminates, we generate a large set of groups with different sizes. (2) Given a group, we retrieve its corresponding part hypotheses, which directly determine the locations of their corresponding root parts. For each of the other leaf parts, we determine its visibility based on voting (similar to Eq. 5.6). The location of a visible leaf part is taken from the part hypothesis with the smallest uncertainty region for the part, thus utilizing the general property of part hypothesis. (3) As the inferred locations of leaf parts are not as precise as root parts, we refine the leaf parts by replacing them with sufficiently close candidate part locations from other part hypotheses. The process is visualized in Fig. 6.3.

Analysis. Essentially, shape composing augment the set of available exemplars, towards the goal of geometrically matching the testing sample. We can adjust the process by tuning the param-

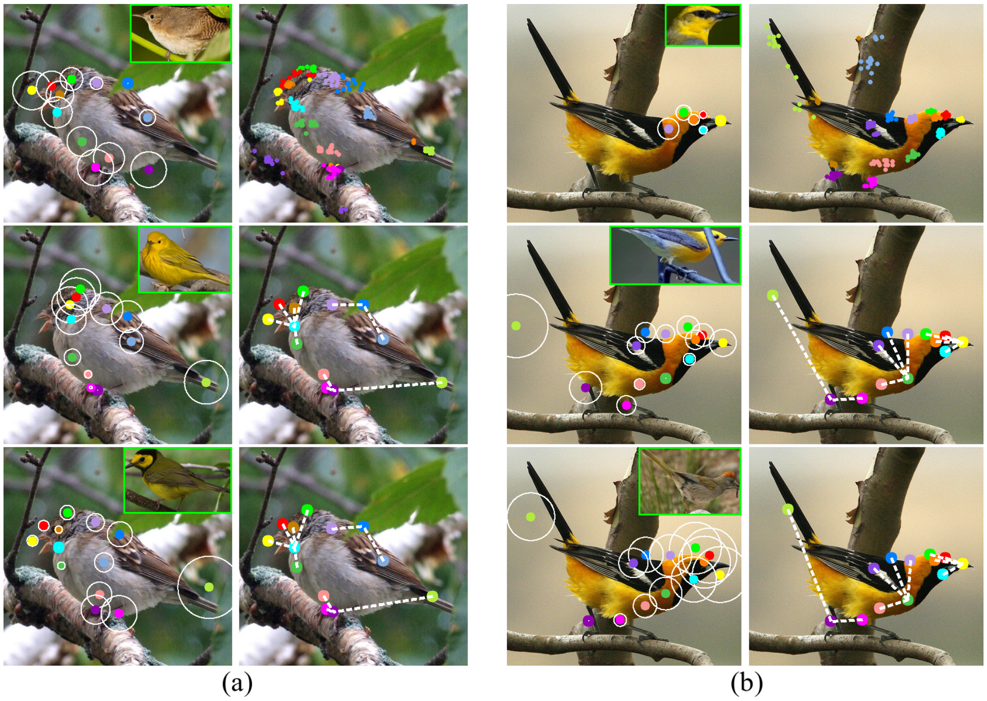


Figure 6.3: Two examples of shape composing. For both (a) and (b), the left column shows three part hypotheses, where the exemplars are shown at the top right corners, and the inferred parts are shown as colored dots. We mark the uncertainty regions of the inferred parts with white circles. The three images (from top to bottom) on the right column correspond to the candidate part detections, composed shape before refinement, and composed shape after refinement. The white dashed lines connect a root part with the associated leaf parts.

eters N and r. For instance, smaller N allows two consistent exemplars to be less similar, while smaller r allows more exemplars to contribute. In our experiment, we find $N = 4$, $r = 1$ gives the best result for the bird dataset, while $N = 3$, $r = 0.5$ works best for the human pose dataset.

6.4 Experiments and Results

We evaluate our part localization method on the bird dataset CUB-200-2011 [Wah *et al.*, 2011] and the human pose dataset LSP (Leeds Sports Poses) [Johnson and Everingham, 2010]. For all the experiments, we use the train/test split provided by the dataset. We withhold 15% of the training data as the validation set.

To evaluate the localization performance, we mainly use the PCP measure (Percentage of Correct Parts). For bird part localization, a correct part should be within 1.5 standard deviation of an MTurk worker's click from the ground-truth part location. For human pose estimation, although we localize the body joints, the PCP measure is based on the limb parts. A correct limb part should have both end joints within half of the part length from the ground-truth end joints.

6.4.1 Super Part Detector vs. Regular Part Detector

To have an idea about the advantage of super part detector over regular part detector, we compare their performance in localizing a single part. We also compare super part detector with individual pair detectors, showing the benefit of aggregating multiple relevant pair detectors. We conduct this experiment on the bird dataset. Note that we do not try to localize all the parts jointly, instead, we predict the location of each part independently.

For regular part detector, we use the pose detectors built in [Liu and Belhumeur, 2013], where there are 200 detectors for each part. At the testing stage, the top five activations across all the pose detectors are output. As for pair detector, the activation of each pair detector casts a vote for its related parts. As such, for each part, we run all the relevant pair detectors (across the orientation), and obtain five highest-scoring votes. For super part detector, we use the top five candidate part detections obtained in Sec. 6.2.2. Note that we do not apply Non-Maximum Suppression, and the activations correspond to the local maximas in the response maps.

The PCPs for each part as well as the total PCP are listed in Tab. 6.1. We also report the top-5 accuracy, which measures the chance of at least one of the top five predictions being correct. From the comparison between pair detectors and pose detectors, we can see that their performance is on par with each other despite the different features and classifiers. However, after building the super part detector from the pair detectors, we achieve significant improvement on the part localization. This is reasonable as the super part detectors incorporate context information. What we want to emphasize is that by imposing geometric constraints at the stage of single part detection, we have high quality part hypotheses which make it promising to compose novel shapes.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
Part	23.4	23.8	31.1	28.1	35.0	28.8	11.5	17.3	18.3	29.4	10.0	34.7	23.9
Pair	27.2	28.4	39.7	31.8	21.4	28.4	5.3	14.5	13.2	38.6	17.9	44.7	25.1
SupP	**62.2**	**57.3**	**66.4**	**61.4**	**74.2**	**65.6**	**40.1**	**40.9**	**53.5**	**66.9**	**34.9**	**71.5**	**57.1**
Part-top5	49.1	47.2	56.2	55.7	62.3	51.1	23.9	37.9	43.9	53.5	26.6	59.1	46.7
Pair-top5	50.1	54.0	66.1	57.0	44.5	49.4	15.2	29.6	31.8	64.7	37.0	68.7	46.1
SupP-top5	**76.9**	**75.8**	**79.8**	**77.1**	**86.3**	**81.7**	**66.0**	**56.1**	**66.9**	**81.4**	**48.3**	**83.8**	**72.5**

Table 6.1: Comparison of different detectors in localizing individual parts. The super part detectors produce the most accurate part localization. From left to right, the parts are: Back, Beak, Belly, Breast, Crown, Forehead, Eye, Leg, Wing, Nape, Tail, and Throat.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
DPM	34.6	26.0	42.0	37.0	47.9	28.7	48.2	-	55.0	41.8	22.4	42.4	40.7
CoE-ext	62.1	49.0	69.0	67.0	72.9	58.5	55.7	40.7	71.6	70.8	40.2	70.8	59.7
Ours-rigid	59.7	59.0	69.5	67.3	**77.1**	**72.2**	67.9	39.9	69.7	75.2	34.7	76.7	63.1
Ours-flex	**64.5**	**61.2**	**71.7**	**70.5**	76.8	72.0	**70.0**	**45.0**	**74.4**	**79.3**	**46.2**	**80.0**	**66.7**

Table 6.2: Comparison of part localization results on CUB-200-2011. Our method outperforms state-of-the-art techniques on all the parts.

6.4.2 Bird Part Localization

We conduct experiments to predict the global part configuration, including part visibilities. As described in Sec. 6.3, we have two approaches. As the CoE-based approach literally matches the whole exemplar, we call it rigid method (Ours-rigid); the shape composing-based approach is then called flexible method (Ours-flex). We also compare with DPM implemented by [Branson *et al.*, 2013] and the extended CoE method (CoE-ext) [Liu and Belhumeur, 2013].

Tab. 6.2 shows the quantitative results. DPM [Branson *et al.*, 2013] has very low accuracy possibly for two reasons: there is too large intra-class variation to be captured by only a few DPM components (14 detectors per part); the first-order spatial constraints in DPM are not strong enough to suppress the detection noise. Although Ours-rigid does not outperform CoE-ext [Liu and Bel-

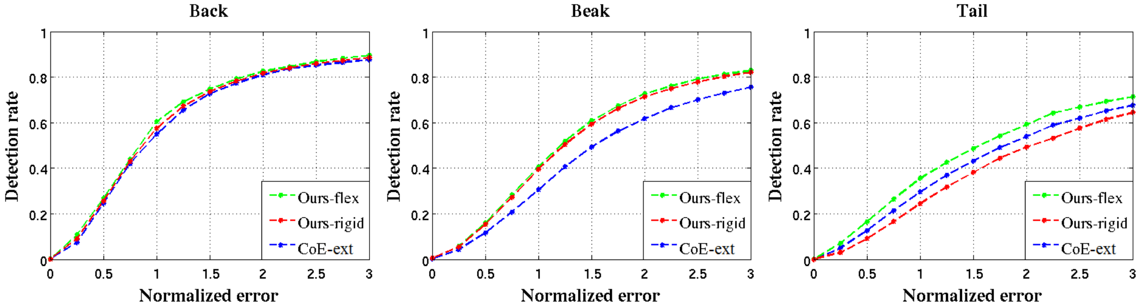


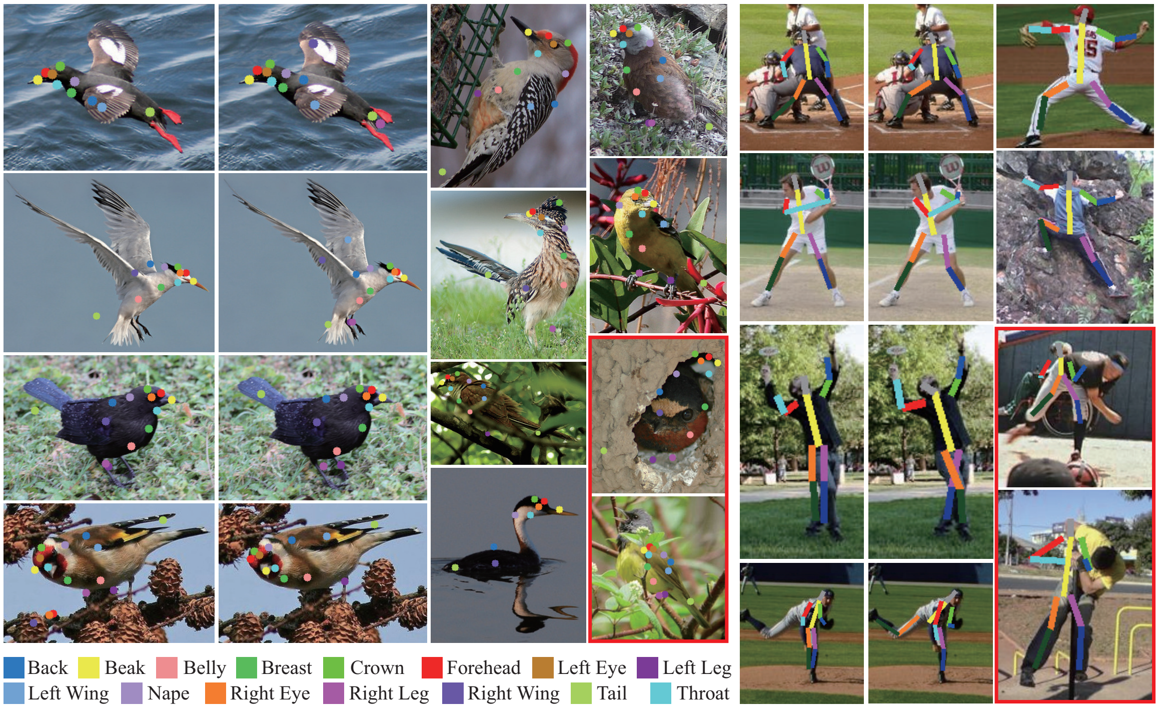
Figure 6.4: Detection rates of Back, Beak, and Tail given varying degrees of localization precision. 1.5 is the threshold for a correct detection. The normalized error is obtained by dividing the localization error with the standard deviation of an MTurk worker's click.

humeur, 2013] by a large margin, the improvement is still remarkable and should be attributed to our part-pair representation. As we do not use subcategory labels and the performance of raw pair detectors is not better than that of pose detectors, the improvement is due to the aggregation of a richer set of appearance models which largely suppress the false detections. Ours-flex further improves the overall PCP over Ours-rigid by about 3.6%. It clearly shows the benefit of composing new shapes, *i.e.*, augmenting the pool of exemplars to fit the testing image.

Fig. 6.4 shows similar comparisons on three representative parts (to generate the curves, we only consider the testing images with the target part visible, which is a subset of the whole testing set used for Tab. 6.2). For relatively rigid part like Back, all the three method have similar localization precision. For rigid part that has high standard of accuracy (*e.g.*, Beak), both Ours-flex and Ours-rigid beat extended CoE significantly. For more deformable part like Tail, the improvement of Ours-flex over Ours-rigid is evident. The reason is that the highest-scoring exemplars from rigid matching tend to fit the majority of parts well, but not the parts with rare or large deformations.

Fig. 6.5 (a) shows some qualitative results for bird part localization. We can see that Ours-rigid fails to accurately localize the parts with large deformation. Because the rigid matching strongly restricts the hypothesized configuration to be from the existing exemplars, Ours-rigid generally respects the spatial prior more than the particular testing image. This is problematic especially when we do not have exemplars that match the testing sample well. Ours-flex mitigates this issue by allowing multiple exemplars to complement with each other based on the particular testing sample.

Similar to [Liu and Belhumeur, 2013], we conduct bird species classification using the localized



(a)

(b)

Figure 6.5: (a) Qualitative results on CUB-200-2011 dataset. The color codes of the bird parts are at the bottom. (b) Qualitative results on LSP dataset. In both sub-figures, the first two columns compare Ours-rigid (left) with Ours-flex (right), the other columns show more examples from Ours-flex. Failures are denoted with red frames.

parts from our full method. On the whole dataset, the mAP (mean average precision) is 48.32% (4.19% improvement); on the 14-species subset, the mAP is 65.18% (2.76% improvement).

6.4.3 Human Pose Estimation

We apply our method to human pose estimation[2] using LSP dataset [Johnson and Everingham, 2010]. Similar to [Pishchulin *et al.*, 2013b], we use observer-centric (OC) annotations. The pair detectors are trained in the same way as those for bird dataset, and altogether we have 796 pair detectors. We also implement [Liu and Belhumeur, 2013] with only pose consistency enabled to make comparison.

[2] In shape composing, the root-leaf distance is measured in a geodesic way to account for articulated deformation.

PCP	Torso	Upper leg	Lower leg	Upper arm	Forearm	Head	Total
Strong-PS	**88.7**	**78.8**	**73.4**	**61.5**	**44.9**	**85.6**	**69.2**
Poselet-PS	87.5	75.7	68.0	54.2	33.9	78.1	62.9
CoE-ext	83.4	69.0	61.7	47.5	28.1	79.3	57.5
Ours-rigid	84.2	69.3	61.5	48.7	28.5	79.9	58.0
Ours-flex	87.6	76.4	69.7	55.4	37.6	82.0	64.8

Table 6.3: Comparison of part localization results on LSP dataset. Our flexible method generates reasonably good results.

The quantitative results are reported in Tab. 6.3. CoE-ext [Liu and Belhumeur, 2013] and Ours-rigid do not work well on human pose estimation. Compared with the bird dataset, the number of training samples is much smaller in LSP, and human body generally has larger articulated deformation. These factors make the Consensus-of-Exemplars approach less effective for human pose estimation. Also note that our rigid method only has marginal improvement over [Liu and Belhumeur, 2013]. One possible reason is that the images in the LSP dataset have already been normalized and cropped (unlike [Wah *et al.*, 2011]), making the effect of suppressing detection noise not prominent.

Tab. 6.3 also shows that Ours-flex significantly improves over Ours-rigid. Ours-flex also outperforms one state-of-the-art technique [Pishchulin *et al.*, 2013a]. Compared with the well-constructed method which employs appearance and geometric models that are tailored to human body [Pishchulin *et al.*, 2013b], our method still produces comparable results. The experiment demonstrates that our part-pair representation can be generalized to the categories with large articulated deformation.

Some qualitative results are shown in Fig. 6.5 (b). Similar to the comparison in Fig. 6.5 (a), Ours-flex achieves more accurate localization by balancing the shape prior from exemplars and the detector activations in the test image.

6.5 Discussion

We have proposed a novel part-pair representation that improves exemplar-based part localization. Exemplars under part-pair representation can impose customizable constraints on the part locations: if we only use the pairs sharing a single part, we can build high-quality exemplar-dependent part detectors which generate reliable and informative part hypotheses; if we allow subsets of pairs from different exemplars to collaboratively form a part configuration, then we achieve shape composing for novel configurations that potentially fit the testing sample better. Moreover, such representation naturally gives us a rich set of appearance models, such that we can score and rank the composed configurations reliably. We also eliminate the need for Consensus operation which may produce unreasonable result when hypotheses are noisy. Experimental results demonstrate that our method produces state-of-the-art results on bird part localization and promising results on human pose estimation.

Chapter 7

Hierarchical Exemplar-based Models

Exemplar-based models have achieved great success on localizing the parts of semi-rigid objects. However, their efficacy on highly articulated objects such as human body is yet to be explored. Inspired by hierarchical object representation and the recent application of Deep Convolutional Neural Networks (DCNNs) on human pose estimation, we would like to incorporate them into our exemplar-based approach. Specifically, we propose a discriminatively trained formulation, featuring multi-level exemplars. The formulation assumes independence between exemplars at different levels for flexibility; it also obtains strong spatial cues by inferring the spatial relations between parts at the same level. Overall, our method strikes a good balance between expressiveness and strength of exemplars, thus achieving better performance than previous exemplar-based approaches.

The basis of our method is the hierarchical representation of an object: starting from atomic parts at the lowest level, we gradually merge them into composite parts at higher levels until we have only one composite part (*i.e.*, the full object). Therefore, we obtain a tree structure, with the nodes representing parts at different levels of granularity. The spatial relations between the parts at each level are first inferred from the image, and then are used to score the exemplar-based models at an upper level. In this process, the strength of Deep Convolutional Neural Networks (DCNNs) and exemplars is exploited. Experimental results show that our method is both effective and generalizable as it achieves state-of-the-art results on two different benchmarks.

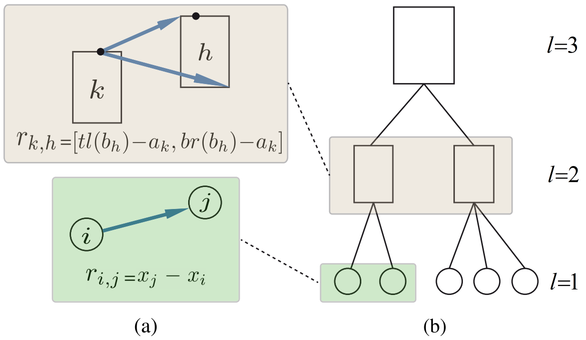


Figure 7.1: Hierarchical object representation. (a) shows the spatial relations between sibling parts. The black dots denotes the anchor points. (b) shows the tree structure of a hierarchy.

7.1 The Approach

Our method features a hierarchical representation of object. We will first describe the relevant notations and introduce hierarchical exemplars. Then we will explain our formulation of pose estimation. In the end, a comparison with relevant techniques will be addressed.

7.1.1 Hierarchical Representation

A hierarchical object (exemplar) contains two types of parts: *atomic part* and *composite part*. An atomic part i is at the finest level of granularity, and can be annotated as a keypoint with pixel location x_i (*e.g.*, elbow). A composite part k is the composite of its child parts (*e.g.*, arm = $\{$shoulder, elbow, wrist$\}$), and is denoted as a tight bounding box b_k containing all the child parts inside. As previous chapters, part configuration X is represented as the locations of atomic parts $[x_1, \ldots, x_N]$ where N is the total number of atomic parts.

Now, we define the *spatial relation* between parts of the same type. For atomic parts i and j, their offset $x_j - x_i$ characterizes the relation $r_{i,j}$ (*e.g.*, shoulder is 20 pixels above the elbow). For composite parts k and h, we first assign anchor points a_k and a_h to them. Anchor points are manually determined such that they are relatively rigid w.r.t the articulated deformation. Then we represent the relation $r_{k,h}$ as $[tl(b_h) - a_k, br(b_h) - a_k]$, where $tl(\cdot)$ and $br(\cdot)$ are the top-left and bottom-right corners of part bounding box (please see Fig. 7.1 (a)). Such definition is consistent in the sense that an atomic part is a degenerate bounding box.

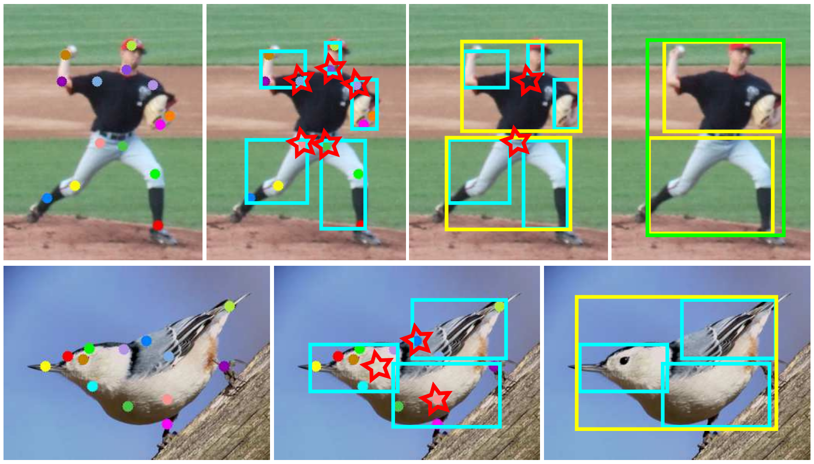


Figure 7.2: The instantiations of part hierarchy on human and bird. In each row, the part levels increase from the left to the right. Each figure shows the parts at the same level with the same color (except for the atomic parts). Immediate children are also plotted for level 2 and above. The stars mark the anchor points for the composite parts.

The hierarchical representation follows a tree structure, as shown in Fig. 7.1 (b). The root denotes the whole object, and each leaf denotes an atomic part. Each internal node k at level $l > 1$ corresponds to a composite part $k^{(l)}$ – the union of its immediate children denoted as $C(k^{(l)})$. The degree of the tree is not bounded, and the structure of the tree depends on the particular object category. A general rule is: geometrically neighboring parts (at the same level) can form a part at the upper level if their spatial relations are sufficiently captured by the training data. Fig. 7.2 shows the instantiation of the hierarchy for two different categories: human and bird. As the bird body is relatively more rigid than the human body, the degrees of bird's internal nodes can be larger, resulting in fewer levels.

The exemplars in previous works [Belhumeur *et al.*, 2011; Liu and Belhumeur, 2013] correspond to a depth-2 hierarchy, where all the atomic parts are the children of the unique composite part (i.e., root part). As a result, each exemplar models the relations between all the atomic parts, making the ensemble of training exemplars not capable of capturing unseen poses. Our hierarchical exemplars, however, adapt to a hierarchy with larger depth which gives us multiple composite parts. By treating each composite part as a standalone object, we have exemplars that model the spatial relations of its child parts (which are referred to as the pose of composite part). We use

$\mathcal{M} = \{\mathcal{M}_k^{(l)}\}|_{l>1}$ to denote the set of exemplar-based models for all the composite parts, where k is the index, and l denotes the level. By design, the hierarchical exemplars cover a spectrum of granularity with proper decomposition of the object, which dramatically improves the expressiveness of exemplars. Note that the depth had better not go too large, as we still want to make use of the strength of exemplars in constraining the configurations of more than two parts.

7.1.2 Formulation

We define an energy function to score a configuration X given an image I and the spatial models $\mathcal{M}$ as

$$S(X|I, \mathcal{M}) = U(X|I) + R(X|I, \mathcal{M}) + w_0, \tag{7.1}$$

where $U(\cdot)$ is the appearance term, $R(\cdot)$ is the spatial term, and w_0 is the bias parameter. Our goal is to find the best configuration $X^* = \arg\max_X S(X|I, \mathcal{M})$.

Appearance Terms: $U(X|I)$ is a weighted combination of the detection scores for each atomic part:

$$U(X|I) = \sum_{i=1}^{N} w_i \, \varphi\left(i|I\left(x_i, s^{(1)}\left(X\right)\right)\right), \tag{7.2}$$

where w_i is the weight parameter, $\varphi(\cdot)$ scores the presence of a part i at the location x_i based on the local image patch (Eq. 7.9), and $s^{(1)}(X)$ denotes the scaling factor to normalize X's atomic parts. As we do not know the scale and location of the target object in the test image (a practical scenario), sliding-window detection over scales is used.

Spatial Terms: We design multi-level spatial terms to evaluate the part relations. Assuming there are L levels in the object hierarchy, and there are n_l parts at the l-th level, then $R(X|I, \mathcal{M})$ is defined as

$$R(X|I, \mathcal{M}) = \sum_{l=2}^{L} \sum_{k=1}^{n_l} \Psi\left(p_k^{(l)}|b_k^{(l)}, I, \mathcal{M}_k^{(l)}\right), \tag{7.3}$$

where $b_k^{(l)}$ denotes the bounding box of part $k^{(l)}$, $p_k^{(l)}$ denotes the pose of $k^{(l)}$, $\mathcal{M}_k^{(l)}$ denotes the corresponding spatial models, and $\Psi(\cdot)$ scores $p_k^{(l)}$ based on both appearance and spatial models. Note that $p_k^{(l)}$ is defined as the spatial relations between the child parts of $k^{(l)}$.

We now elaborate the derivation of $\Psi(\cdot)$. Using exemplar-based models, we can assume $\mathcal{M}_k^{(l)}$ contains T exemplars $\{X_i\}|_{i=1,\ldots,T}$, each of which dictates a particular pose p_i (*e.g.*, an example of raised arm). Here, we drop the subscript k and superscript l for simplicity. With these in hand, we evaluate $\Psi(\cdot)$ as the combination of two terms:

$$\Psi\left(p_k^{(l)}|b_k^{(l)}, I, \mathcal{M}_k^{(l)}\right) = \alpha_k^{(l)}\phi\left(p_o|I\left(b_k^{(l)}, s^{(l-1)}(X)\right)\right) + \beta_k^{(l)}\psi\left(p_k^{(l)}, p_o\right), \tag{7.4}$$

where $\alpha_k^{(l)}$ and $\beta_k^{(l)}$ are the weight parameters, p_o (corresponding to exemplar X_o) is the pose that best fits $p_k^{(l)}$, $\phi(\cdot)$ evaluates the likelihood of pose p_o being present in the image region at $b_k^{(l)}$ (Eq. 7.10, 7.11), $s^{(l-1)}$ indicates that the relevant image patches also need to be resized as Eq. 7.2, $\psi(\cdot)$ measures the similarity between $p_k^{(l)}$ and p_o as

$$\psi(p_k^{(l)}, p_o) = -\min_t ||\vec{X}_k^{(l)} - t(\vec{X}_o)||^2, \tag{7.5}$$

where t denotes the operation of similarity transformation (the rotation angle is constrained), $\vec{X}_k^{(l)}$ denotes the vectorized locations of parts $C(k^{(l)})$ in X, and $\vec{X}_o$ denotes the vectorized X_o.

As multi-scale image cues and multi-level spatial models are both involved, $\Psi(\cdot)$ covers part relations at different levels of granularity. For instance, at a fine scale (small l), it evaluates whether the arm is folded; at a coarse scale (large l), it evaluates whether the person is seated.

7.1.3 Comparisons with Related Methods

We make independence assumption on the spatial models in Eq. 7.3, which can benefit articulated pose estimation. The reason lies in that it gives us a collection of spatial models that can better handle rare poses. For instance, our models allow a person to exhibit arbitrarily plausible poses at either arm as long as the spatial relations between the two arms are plausible. With such assumption, our formulation still captures the part relations thoroughly and precisely: the relations between sibling parts are encoded explicitly in Eq. 7.4; the relations between parent and child parts are implicitly enforced (the same X is referred to across the levels).

Below, we address the differences between our method and relevant techniques, such as image dependent spatial relations [Chen and Yuille, 2014; Sapp and Taskar, 2013; Pishchulin *et al.*, 2013a] and hierarchical models [Sun and Savarese, 2011; Wang *et al.*, 2011; Wang and Li, 2013]:

- Unlike [Chen and Yuille, 2014; Sapp and Taskar, 2013], our method infers from the image not only the spatial relations between atomic parts (*e.g.*, elbow and shoulder), but also the spatial relations between composite parts (*e.g.*, upper body and lower body).

- Unlike [Sapp and Taskar, 2013; Pishchulin *et al.*, 2013a], we do not conduct the selection of spatial models upfront as errors in this step are hard to correct afterwards. Instead, our selection of spatial models is based on the configuration under evaluation (the second term in Eq. 7.4), which avoids pruning the state space too aggressively.
- Unlike [Sun and Savarese, 2011; Wang *et al.*, 2011; Wang and Li, 2013], our method directly optimizes on the atomic part locations, avoiding the interference from localizing the composite parts. Also, we turn to exemplars to constrain the part relations, rather than using piece-wisely stitched "spring models".

7.2 Inference

The optimization of Eq. 7.1 does not conform to general message passing framework due to the dependency of p_o on X (Eq. 7.4) and the interactions between variables x_i across multiple levels (Eq. 7.3). Therefore, we propose an algorithm (Algorithm 1) which simplifies the evaluation of Eq. 7.3 with hypothesized parts. Although being approximate, the algorithm is efficient and yields good results. In the following sections, we explain two major components of the algorithm.

7.2.1 Hypothesize and Test

The first component is *Hypothesize and Test*, which leverages a RANSAC-like procedure of exemplar matching. For this, we rewrite Eq. 7.3 in a recursive form which scores the sub-tree rooted at $b_k^{(l)}$ ($l \geq 2$):

$$f(b_k^{(l)}) = \sum_{j \in C(k^{(l)})} f(b_j^{(l-1)}) + \Psi\left(p_k^{(l)} | b_k^{(l)}, I, \mathcal{M}_k^{(l)}\right). \tag{7.6}$$

Note that $f(b_k^{(1)}) = 0$ for any k. By comparing Eq. 7.6 with Eq. 7.3, we can see that $f(b_1^{(L)}) = R(X|I, \mathcal{M})$.

Hypothesize and Test is conducted in a bottom-up manner: (1) Given the hypothesized locations of all the parts at level $l-1$ (each part has multiple hypotheses), transform the exemplars at level l to the test image with similarity transformation such that each exemplar's child parts align with two randomly selected hypotheses of atomic parts (if $l = 2$), or up to two hypotheses of composite parts (if $l > 2$). (2) The geometrically well-aligned exemplars generate hypotheses for the parts at level

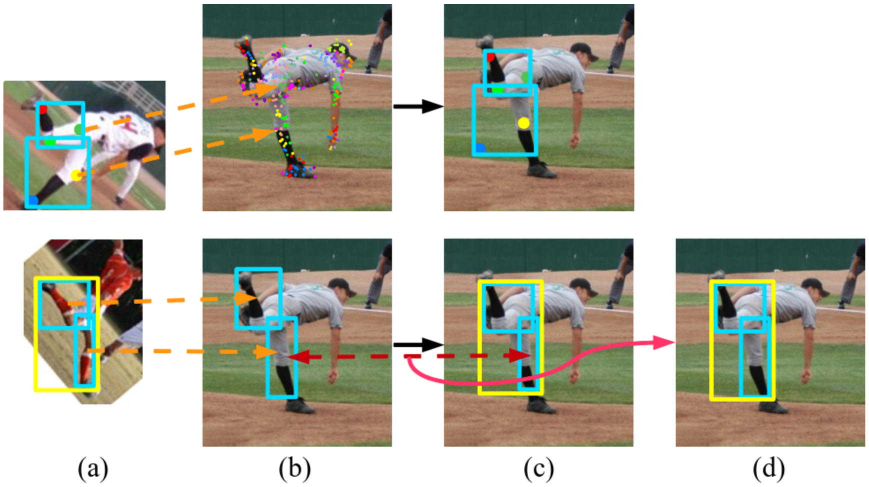


Figure 7.3: Generate hypotheses using exemplars. The first row corresponds to $l = 2$ and the second row corresponds to $l = 3$. (a) Two training exemplars. (b) The test image overlaid with hypotheses at level $l - 1$. (c) Part hypotheses at level l which are inherited from the exemplars. (d) Augmented hypothesis after swapping the hypotheses of child part.

l. Each hypothesis carries from exemplar the object size, the corresponding sub-tree, as well as the pose p_o for each node in the sub-tree. (3) Augment the hypotheses of $k^{(l)}$ (if $l > 2$) by replacing their sub-trees with geometrically close hypotheses at level $l - 1$. (4) Evaluate all the hypotheses at level l using Eq. 7.6 and keep the top-scoring ones. (5) Increment l and go to step (1). Fig. 7.3 shows examples of the first three steps.

7.2.2 Backtrack

The second component of the algorithm is *Backtrack*. Assuming we have a hypothesis of the root part $b_1^{(L)}$, we can trace down its tree constructed in Sec. 7.2.1, which naturally gives us p_o's (in Eq. 7.4) for each composite parts, as well as the hypothesized locations for the atomic parts $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N]$.

The next step is to re-score $\hat{X}$ by obtaining its refined configuration $\hat{X}^*$. For this purpose, we define $g(\cdot)$ to approximate $S(\cdot)$ in Eq. 7.1:

$$g(\hat{X}|I, b_1^{(L)}) = U(\hat{X}|I) + \sum_{k=1}^{n_2} \beta_k^{(2)} \psi(p_k^{(2)}, p_o) + D, \tag{7.7}$$

Algorithm 1: Inference Procedure for Pose Estimation

```
Input: Multi-level exemplars $\{\mathcal{M}^{(l)}\}|_{l=2,\ldots,L}$;
       Multi-level appearance models $\{\mathcal{C}^{(l)}\}|_{l=1,\ldots,L-1}$;
       Test image $I$;
       Maximum number of hypotheses per part Z;
Output: The optimal configuration $X^*$;
$hypo^{(1)} \leftarrow$ top $Z$ local maximas from $\mathcal{C}^{(1)}(I)$, $l \leftarrow 2$;
while $l \leq L$ do
    $hypo^{(l)} \leftarrow$ randomly align $\mathcal{M}^{(l)}$ with $hypo^{(l-1)}$;
    Augment $hypo^{(l)}$ if $l > 2$;
    Evaluate $hypo^{(l)}$ using Eq. 7.6 and $\mathcal{C}^{(l-1)}(I)$;
    $hypo^{(l)} \leftarrow$ top-scoring $Z$ $hypo^{(l)}$;
    $l \leftarrow l + 1$;
Refine and re-score $hypo^{(L)}$ through backtrack;
$X^* \leftarrow$ highest-scoring $\hat{X}^*$;
return $X^*$;
```

where $D = f(b_1^{(L)}) + w_0$. Such approximation assumes $s(\hat{X})$, p_o and $b_k^{(l)}$ for $l > 2$ change little during the refinement (which mainly changes the atomic part locations). After plugging Eq. 7.2 and Eq. 7.5 into Eq. 7.7, we can solve each atomic part independently as

$$\hat{x}_i^* = \arg\max_{x_i \in \mathcal{R}(\hat{x}_i)} w_i\, \varphi(i|I(x_i, s^{(1)}(\hat{X}))) - \beta_k^{(2)} ||x_i - \hat{x}_i||^2. \tag{7.8}$$

where $\mathcal{R}(\hat{x}_i)$ denotes the search region of part i. We define the search region as a circle with radius equal to 15% of the average side length of $b_1^{(L)}$. We evaluate Eq. 7.8 for all the pixel locations inside the circle, which gives us the highest-scoring location. In the end, we obtain the refined configuration $\hat{X}^* = [\hat{x}_1^*, \hat{x}_2^*, \ldots, \hat{x}_N^*]$ with updated score $g(\hat{X}^*|I, b_1^{(L)})$.

7.3 Model Learning

In this section, we describe how we learn the appearance models in Eq. 7.1 (i.e., $\varphi(\cdot)$ and $\phi(\cdot)$), as well as how we learn the weight parameters $\mathbf{w}$ (i.e., w_i, $\alpha_k^{(l)}$, and $\beta_k^{(l)}$).

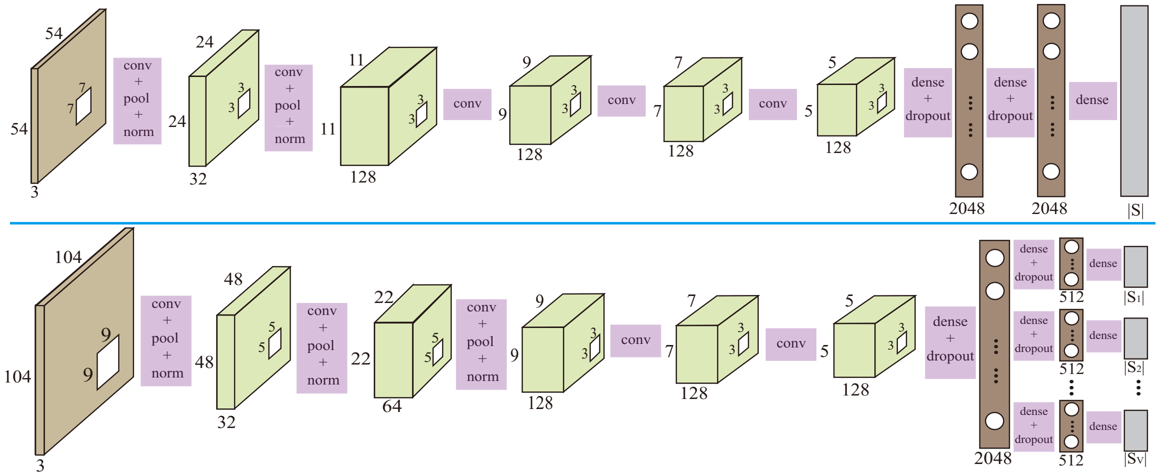


Figure 7.4: TOP: The architecture of DCNN-based atomic part detector. It consists of five convolutional layers, two max-pooling layers and three fully-connected layers. The output dimension is |S|. BOTTOM: The architecture of DNN-based models for composite parts. It consists of five convolutional layers, three max-pooling layers and three fully-connected layers. The last two layers branch out, with each branch targeting the possible spatial relations of one composite part to its predefined reference part.

7.3.1 Relations Between Atomic Parts

We follow the method of [Chen and Yuille, 2014] to infer the spatial relations between atomic parts. Specifically, we design a DCNN-based multi-class classifier using Caffe [Jia *et al.*, 2014]. The architecture is shown in the first row of Fig. 7.4. Each value in the output corresponds to $p(i, m_{i,j}|I(x, s^{(1)}(X)))$, which is the likelihood of seeing an atomic part i with a certain spatial relation (type $m_{i,j}$) to its predefined neighbor j, at location x. If $i = 0$, then $m_{i,j} \in \{0\}$, indicating the background; if $i \in \{1, \ldots, N\}$, then $m_{i,j} \in \{1, \ldots, T_{i,j}\}$. By marginalization, we can derive $\varphi(\cdot)$ and $\phi(\cdot)$ as

$$\varphi(i|I(x, s)) = \log(p(i|I(x, s))). \tag{7.9}$$

$$\phi(m_o|I(b_k^{(2)}, s)) = \sum_{i \in C(k^{(2)})} \log(p(m_{i,j}|i, I(x_i, s))). \tag{7.10}$$

Note that superscript $^{(1)}$ and X are dropped for clarity, i and j are siblings. To define type $m_{i,j}$, during training, we discretize the orientations of $r_{i,j}$ into $T_{i,j}$ (*e.g.*, 12) uniform bins, and $m_{i,j}$

indicates a particular bin. The Training samples are then labeled as $(i, m_{i,j})$, and the image patches are centered at the target parts.

7.3.2 Relations Between Composite Parts

We build another DCNN-based model to infer the spatial relations between composite parts, as shown in the second row of Fig. 7.4, the architecture differs from that for atomic parts in multiple aspects. First, as the model targets composite parts which have coarser levels of granularity, the network has a larger receptive field. Second, as there are relatively fewer composite parts than atomic parts, we let all the composite parts share the features in the first few layers. Third, as the composite parts have different granularity with possibly significant overlap with each other, the DCNN branches out to handle them separately.

Assuming the i-th branch corresponds to part i at level $l-1$ (Note that $l > 2$), then the branch has $|\mathrm{S}_i|$-dim output with each value being $p(m_{i,j}|i, I(a_i, s^{(l-1)}(X)))$ based on the image patch centered at the anchor point a_i. Assuming the parent of part i is part $k^{(l)}$, then $\phi(\cdot)$ (if $l > 2$) is evaluated as

$$\phi(p_o|I(b_k^{(l)}, s)) = \sum_{i \in C(k^{(l)})} \log(p(m_{i,j}|i, I(a_i, s))). \tag{7.11}$$

Note that superscript $^{(l-1)}$ and X are dropped for clarity. To train this model, we cluster the relation vector $r_{i,j}$ into $T_{i,j}$ (*e.g.*, 24) clusters (types) for part i, and the training samples are labeled accordingly.

7.3.3 Weight Parameters

Eq. 7.1 can be written as a dot product $\langle \mathbf{w}, \Phi(X, I, \mathcal{M}) \rangle$. Given a training sample (X, I), we compute $\Phi(X, I, \mathcal{M})$ as its feature. Each training sample also has a binary label, indicating if the configuration X is correct. Therefore, we build a binary max-margin classifier [Tsochantaridis *et al.*, 2004] to estimate $\mathbf{w}$, with non-negativity constraints imposed. To avoid over-fitting, the training is conducted on a held-out validation set that was not used to train the DCNNs.

Before training, we augment the positive samples by randomly perturbing their part locations as long as they are reasonably close to the ground-truth locations. To generate the negative samples, we randomly place the configurations of positive samples at the incorrect regions in the training images, with Gaussian noise added to the part locations.

7.4 Experiments and Results

We evaluate our method extensively on multiple benchmarks, and conduct diagnostic experiments to show the effect of different components in our method.

7.4.1 Human Pose Estimation on LSP Dataset

The Leeds Sports Pose (LSP) dataset [Johnson and Everingham, 2010] includes $1,000$ images for training and $1,000$ images for testing, where each image is annotated with 14 joint locations. We augment the training data by left-right flipping, and rotation through 360°. We use observer-centric (OC) annotations to have fair comparisons with the majority of existing methods. To measure the performance, we use Percentage of Correct Parts (PCP). In PCP measure, a “part” is defined as a line segment connecting two neighboring joints. If both of the segment endpoints (joints) lie within 50% of the length of the ground-truth annotated endpoints, then the part is correct.

In this experiment, we build a hierarchy of four levels for human body. The first level contains the atomic body joints; the second level has five composite parts (Head, Right arm, Left arm, Right leg, and Left leg); the third level has two composite parts (Head&Arms and Legs); the fourth level corresponds to the whole body. To gain an understanding of the effect of the two components of our inference algorithm, we evaluate our full method (which will be referred to as “Ours-full”), and a variant of our method (which will be referred to as “Ours-partial”, and “Ours-no-HIER”). Ours-full corresponds to the whole inference algorithm; Ours-partial only conducts the first part of the inference algorithm, then obtains the best root hypothesis based on Eq. 7.6, and outputs the locations of its atomic parts; Ours-no-HIER only uses full-body exemplars (after augmentation) as the spatial models.

The quantitative results of our method as well as its counterparts are listed in Tab. 7.1. Ours-full generally outperforms the state-of-the-art methods on all the parts. The improvement over IDPR [Chen and Yuille, 2014] demonstrates the effect of reasoning multi-level spatial reasoning. We expect to see even larger improvement if we augment the annotations with midway points between joints as [Chen and Yuille, 2014] does. We also experiment with person-centric (PC) annotations

Method	Torso	ULeg	LLeg	UArm	LArm	Head	Avg
Strong-PS	88.7	78.8	73.4	61.5	44.9	85.6	69.2
PoseMachine	88.1	78.9	73.4	62.3	39.1	80.9	67.6
IDPR	92.7	82.9	77.0	69.2	**55.4**	87.8	75.0
Ours-partial	89.2	79.5	73.6	65.8	50.3	85.6	71.3
Ours-no-HIER	85.4	75.3	66.7	54.9	37.5	82.5	63.7
Ours-full	93.5	**84.4**	**78.3**	**71.4**	55.2	**88.6**	**76.1**
Ours-full (PC)	**93.7**	82.2	76.0	68.6	53.2	88.3	74.2

Table 7.1: Comparison of pose estimation results (PCP) on LSP dataset. Our method achieves the best overall performance.

on the same image set, where the accuracy drops slightly. Ours-full achieves improvement over Ours-partial and Ours-no-HIER by a large margin, which demonstrates the benefits of *backtrack* (higher precision) and hierarchical exemplars (more expressive models). Note that Ours-partial already outperforms Strong-PS [Pishchulin *et al.*, 2013b] and PoseMachine [Ramakrishna *et al.*, 2014], which should be partly attributed to the use of DCNN models.

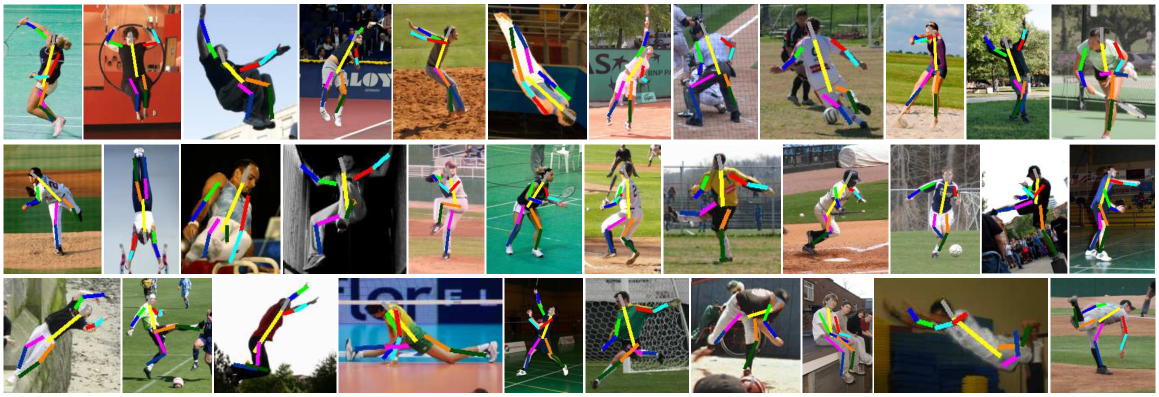


Figure 7.5: Qualitative results of human pose estimation on LSP dataset (OC annotations).

Fig. 7.5 shows some testing examples, which are selected with high diversity in the poses. We can see that our method achieves accurate localization for most of the body joints, even in the case of large articulated deformation.

7.4.2 Human Pose Estimation on LSP Extended Dataset

To have fair comparisons with [Toshev and Szegedy, 2014; Tompson *et al.*, 2014], we train and test our models on LSP extended dataset using PC annotations. Altogether, we have $11,000$ training images and 1000 testing images. As the quality of the annotations for the additional training images varies a lot, we manually filter out about 20% of them. We also augment the training data through flipping and rotation.

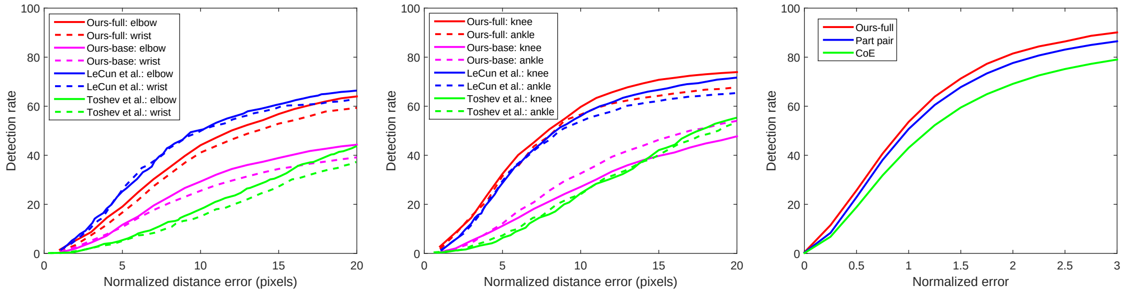


Figure 7.6: Detection rate vs. normalized error curves. LEFT, MIDDLE: arm (elbow and wrist) and leg (knee and ankle) detection on the LSP dataset. RIGHT: Average part detection on the CUB-200-2011 bird dataset.

We use Percentage of Detected Joints (PDJ) to evaluate the performance, which provides an informative view of the localization precision. In this experiment, we evaluate the baseline of our method (referred to as "Ours-base") by only using the first term in Eq. 7.1. It is equivalent to localizing the parts independently. In Fig. 7.6, we plot the detection rate vs. normalized error curves for different methods. From the curves, we can see that Ours-base already achieves better accuracy than [Toshev and Szegedy, 2014] except for Knee. It demonstrates that a detector that scores the part appearance is more effective than a regressor that predicts the part offset. Ours-full achieves significant improvement over Ours-base by incorporating the multi-level spatial models. Our method is also comparable to [Tompson *et al.*, 2014] which enjoys the benefit of jointly learning appearance models and spatial context. [Tompson *et al.*, 2014] has higher accuracy on the lower arms, while we have better results on the lower legs. Also note that [Tompson *et al.*, 2014] requires delicate implementation of a sophisticated network architecture, while our method allows the use of off-the-shelf DCNN models.

Method	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
CoE-ext	62.1	49.0	69.0	67.0	72.9	58.5	55.7	40.7	71.6	70.8	40.2	70.8	59.7
Part-pair	64.5	61.2	71.7	70.5	76.8	72.0	70.0	45.0	74.4	79.3	46.2	80.0	66.7
DCNN-CoE	64.7	63.1	74.2	71.6	76.3	72.9	69.1	48.1	72.5	82.0	46.8	81.5	67.5
Ours-partial	65.1	64.2	74.6	72.4	77.1	73.8	70.4	48.6	73.1	82.5	48.3	82.2	68.3
Ours-full	**67.3**	**65.6**	**75.9**	**74.4**	**78.8**	**75.3**	**72.7**	**50.7**	**75.3**	**84.7**	**49.9**	**84.2**	**70.2**

Table 7.2: Comparison of part localization results on the CUB-200-2011 bird dataset. Our method outperforms the previous methods by a large margin. From left to right, the parts are: Back, Beak, Belly, Breast, Crown, Forehead, Eye, Leg, Wing, Nape, Tail, Throat, and Total.

7.4.3 Bird Part Localization

We also evaluate our method on the CUB-200-2011 bird dataset, which contains $5,994$ images for training and $5,794$ images for testing. Each image is annotated with image locations for 15 parts. We also augment the training data through flipping and rotation. As birds are less articulated than humans, we design a three-level hierarchy for birds. The first level contains the atomic parts; the second level has three composite parts (Head, Belly&Legs, and Back&Tail); the third level corresponds to the whole bird. Although we did not prove that the manually-designed hierarchy is optimal, we empirically find that it facilitates the prediction of coarse-level part relations.

We use both PCP and PDJ to measure performance. In the bird dataset, a correct part detection should be within 1.5 standard deviation of an MTurk worker's click from the ground-truth location. For a semi-rigid object such as bird, directly applying exemplar-based models can produce very good results. Therefore, we replace the part detectors in [Liu and Belhumeur, 2013] with our DCNN-based detector (which target the atomic parts), obtaining an enhanced CoE method (which will be referred to as "DCNN-CoE").

We compare the results of different methods in Tab. 7.2, including CoE-ext [Liu and Belhumeur, 2013] and Part-pair [Liu *et al.*, 2014]. First, DCNN-CoE outperforms CoE-ext significantly, demonstrating that DCNN is much more powerful than the conventional classification model (*e.g.*, SVM). DCNN-CoE also outperforms Part-pair with much less overhead, thanks to the efficiency of multi-class detector. Using our new method, the localization accuracy is further improved. Ours-partial improves slightly over DCNN-CoE, which is reasonable as Ours-partial is essentially multi-level

DCNNs plus hierarchical exemplars, and the flexibility of hierarchical exemplars has limited benefit for semi-rigid objects. Also note that Ours-partial uses an incomplete scoring function. By considering the full scoring function, Ours-full achieves the best results on all the parts. The rightmost plot in Fig. 7.6 shows the detection rate vs. normalized error curves. The localization error is normalized by the standard deviation of an MTurk worker's click for the part. From the curves, we can see that our method outperforms the previous state of the art in both high-precision and low-precision regions.

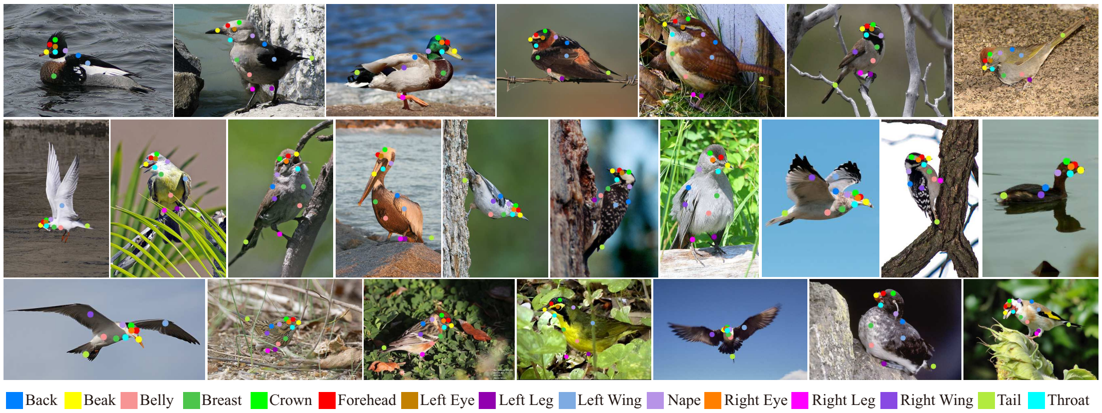


Figure 7.7: Qualitative results of part localization on CUB-200-2011 bird dataset. The color codes are shown at the bottom.

Some qualitative results are shown in Fig. 7.7. From the examples, we can see that our method is capable of capturing a wide range of poses, shapes and viewpoints. In addition, our method localizes the bird parts with very high precision.

7.5 Discussion

In this chapter, we propose a novel approach for articulated pose estimation. The approach exploits the part relations at different levels of granularity through multi-scale DCNN-based models and hierarchical exemplar-based models. By incorporating DCNN-based appearance models in the spatial terms, our method couples spatial relations with image cues, thus better capturing the interactions between the parts than otherwise. By introducing hierarchy in the exemplar-based models, we enjoy much more expressive spatial models even if the training data are limited. In addition, we propose an efficient algorithm to infer “good-enough” part configurations from a sophisticated formulation. These efforts together enable us to achieve state-of-the-art results on different datasets, which demonstrates the effectiveness and generalization ability of our method.

Chapter 8

Conclusions

8.1 Thesis Summary

In this thesis, we have shown how to build automatic visual systems that perform fine-grained classification using part-based method. The core of the system is to extract features with part-level correspondences across different instances. By conducting extensive experiments on different categories and comparing with other approaches without using parts, we observe significant gains in the classification performance. It agrees with our intuition that subtle differences between similar subcategories generally lie in the local regions at or around parts. If we extract features from these regions and maintain the part-level correspondence in the feature vectors, we make it easier for classifiers to learn discriminative features that best differentiate subcategories. In building the fine-grained visual systems, we have made available two datasets with part labels, facilitating vision community to further explore part-based visual recognition.

We have also demonstrated the potential of exemplar-based models in localizing object parts. In applying them to different categories, from relatively rigid face to highly deformable human body, we gain insight about their pros and cons. These findings motivated us to explore richer appearance models and sophisticated object representations that better capture the interactions between parts. By combining exemplars with these newly designed components, exemplar-based models are significantly enhanced and generalized. More importantly, such improvement further advances the state of the art in object part localization. Therefore, our work makes progress towards the goal of enabling machines to perceive the presence and configuration of objects.

8.2 Future Directions

As we mentioned earlier, part localization is to build the non-linear mapping from appearance space to pose space. Therefore, completely solving this problem requires the understanding of how these two spaces look like and how they are related to each other. The ideas and methods presented in this thesis are just initial attempts. There are many things to explore regarding exemplars and part localization. First, exemplars in our work only encode the geometric information of the object. It is interesting to see if geometric exemplars have counterparts in appearance space. In other words, exemplars may carry the relations between part appearance besides spatial relations. For instance, the pattern at one part should co-exist with certain pattern at another part. Second, we use non-parametric exemplars to represent the pose manifold as a discrete set. However, we still lack insight about what the manifold looks like. For instance, we don't know what is the best way to measure the distance between two configurations, especially when some parts are not visible due to self-occlusion. Otherwise, we may be able to build low-dimensional embedding for the exemplars, facilitating the mapping from image cues to exemplars that are applicable (*i.e.*, ruling out incorrect exemplars). Third, a follow-up question is whether we can design more informative part annotations than just keypoint locations. It should be helpful to know the support of part. Last, current methods including ours estimate the part locations in a feed-forward manner. We feel some feedback from currently estimated results may help us correct the errors or adjust the inference incrementally. In the following two sections, we briefly discuss potential directions that differ from current methodologies for object part localization. No matter what techniques these directions will lead us to, we expect to have machines perform vision tasks in in a more human-like way.

8.2.1 Sequential Part Localization

Although we try to detect all the parts on an object, they do not necessarily exhibit equal levels of difficulty. For instance, in bird part localization, head parts are generally easier to detect than legs and tail; in human pose estimation, head is also easier to detect than other parts. This observation implies a sequential part localization, where easier (more reliable) parts are detected first, then the other more difficult parts based on the easy ones. One possible method of estimating the difficulty is: build regular part detectors, and evaluate them on validation set. The error rates then indicate the levels of difficulty. As this is just a statistical estimation, the ranking of difficulty may not hold for a particular image. Therefore, an adaptive strategy is needed so that the sequence of parts being detected also respects the image cues. In some sense, the strategy is equivalent to a cascade which needs to be designed and learned in a principled way. [Alexe *et al.*, 2012; Chen *et al.*, 2014] have relevant ideas, but they do not particularly study the problem of part localization or the sequence of detections.

8.2.2 Holistic Pose Estimation

Most of existing methods treat parts on an object as independent entities, and build separate models for them. As such, they naturally follow a bottom-up paradigm where individual parts are detected, and then combined to predict the final configuration. Although this idea is straightforward, it may fall in the trap of noisy part detections from the beginning. As we discussed before, there are ambiguities in the local part appearance, making it hard to build part detectors that are reliable. However, if incorrect regions that contain misleading patterns are excluded based on some analysis of the global or surrounding image content, the imperfect part detectors will have higher chances of locating the parts correctly.

DeepPose [Toshev and Szegedy, 2014] has attempted to approach the problem by building DCNN-based part regressors. It takes as input the image of the full body, but its loss function does not involve the interactions between parts explicitly. Therefore, the individual parts are inferred independently throughout the method. At the least, we would like to have some intermediate representations of part configuration, such that the mapping from raw image to their activations and the mapping from the activations to individual part locations leverage enough contextual image cues. To obtain such representations, we may need to go back to understanding the pose manifold mentioned above. Poselets [Bourdev *et al.*, 2010] provide one choice of the intermediate representation, but they are built in a greedy and ad-hoc way without supervision from a global objective. In addition, getting the part locations from Poselet activations just involves simple voting strategy.

With the development of deep neural networks, we expect to see more works in this direction that predict objects, parts, and pose based on fully analysed image cues.

Bibliography

[Alexe *et al.*, 2012] Bogdan Alexe, Nicolas Heess, Yee W. Teh, and Vittorio Ferrari. Searching for objects driven by context. *Proc. NIPS*, 2012.

[Amberg and Vetter, 2011] Brian Amberg and Thomas Vetter. Optimal landmark detection using shape models and branch and bound. *Proc. ICCV*, 2011.

[Arca *et al.*, 2006] Stefano Arca, Paola Campadelli, and Raffaella Lanzarotti. A face recognition system based on automatically determined facial fiducial points. *Pattern Recognition*, 39(3):432–443, 2006.

[Belhumeur *et al.*, 2008] Peter N. Belhumeur, Daozheng Chen, Steven Feiner, David W. Jacobs, W. John Kress, Haibin Ling, Ida Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, and Ling Zhang. Searching the worlds herbaria: A system for visual identification of plant species. *Proc. ECCV*, pages 116–129, 2008.

[Belhumeur *et al.*, 2011] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *Proc. CVPR*, 2011.

[Berg and Belhumeur, 2012] Thomas Berg and Peter N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. *Proc. BMVC*, 2012.

[Berg and Belhumeur, 2013] Thomas Berg and Peter N. Belhumeur. POOF: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. *Proc. CVPR*, 2013.

[Blaschko and Lampert, 2009] Matthew B. Blaschko and Christoph H. Lampert. Object localization with global and local context kernels. *Proc. BMVC*, 2009.

[Bourdev and Malik, 2009] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. *Proc. ICCV*, 2009.

[Bourdev *et al.*, 2010] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. *Proc. ECCV*, 2010.

[Branson *et al.*, 2010] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. Visual recognition with humans in the loop. *Proc. ECCV*, 2010.

[Branson *et al.*, 2011] Steve Branson, Pietro Perona, and Serge Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. *Proc. ICCV*, 2011.

[Branson *et al.*, 2013] Steve Branson, Oscar Beijbom, and Serge Belongie. Efficient large-scale structured learning. *Proc. CVPR*, 2013.

[Branson *et al.*, 2014] Steve Branson, Grant Van Horn, Pietro Perona, and Serge Belongie. Improved bird species recognition using pose normalized deep convolutional nets. *Proc. BMVC*, 2014.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Cao *et al.*, 2012] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *Proc. CVPR*, 2012.

[Cevikalp and Triggs, 2012] Hakan Cevikalp and Bill Triggs. Efficient object detection using cascades of nearest convex model classifiers. *Proc. CVPR*, 2012.

[Chai *et al.*, 2013] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. *Proc. ICCV*, 2013.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[Chen and Yuille, 2014] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. *Proc. NIPS*, 2014.

[Chen *et al.*, 2014] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun3, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. *Proc. CVPR*, 2014.

[Cootes *et al.*, 2001] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE TPAMI*, 2001.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Cristinacce and Cootes, 2006] David Cristinacce and Timothy F. Cootes. Feature detection and tracking with constrained local models. *Proc. BMVC*, 2006.

[Csurka *et al.*, 2004] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *Proc. CVPR*, 2005.

[Dantone *et al.*, 2012] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc V. Gool. Real-time facial feature detection using conditional regression forests. *Proc. CVPR*, 2012.

[Desai *et al.*, 2011] Chaitanya Desai, Deva Ramanan, and Fowlkes Charless C. Discriminative models for multi-class object layout. *IJCV*, 95(1):1–12, 2011.

[Divvala *et al.*, 2012] Santosh K. Divvala, Alexei A. Efros, and Martial Hebert. How important are deformable parts in the deformable parts model? *Parts and Attributes Workshop, ECCV*, 2012.

[Dollár *et al.*, 2009] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. *Proc. BMVC*, 2009.

[Dollár *et al.*, 2010] Piotr Dollár, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. *Proc. BMVC*, 2010.

[Dollár *et al.*, 2012] Piotr Dollár, Ron Appel, and Wolf Kienzle. Crosstalk cascades for frame-rate pedestrian detection. *Proc. ECCV*, 2012.

[Dollár, 2009] Piotr Dollár. Piotr's Image and Video Matlab Toolbox (PMT). `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`, 2009.

[Everingham *et al.*, 2006] Mark Everingham, Josef Sivic, and Andrew Zisserman. hello! my name is... buffy automatic naming of characters in tv video. *Proc. BMVC*, 2006.

[Farrell *et al.*, 2011] Ryan Farrell, Om Oza, Ning Zhang, Vlad I. Morariu, Trevor Darrell, and Larry S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. *Proc. ICCV*, 2011.

[Felzenszwalb and Huttenlocher, 2005] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[Felzenszwalb *et al.*, 2010a] Pedro F. Felzenszwalb, Ross B. Girshick, and David McAllester. Cascade object detection with deformable part models. *Proc. CVPR*, 2010.

[Felzenszwalb *et al.*, 2010b] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE TPAMI*, 2010.

[Fidler *et al.*, 2013] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. *Proc. CVPR*, 2013.

[Gall *et al.*, 2011] Juergen Gall, Nima Razavi, and Luc V. Gool. An introduction to random forests for multi-class object detection. *Theoretical Foundations of Computer Vision*, pages 243–263, 2011.

[Gao *et al.*, 2011] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. *Proc. CVPR*, 2011.

[Gavves *et al.*, 2013] Efstratios Gavves, Basura Fernando, Cees G. M. Snoek, Arnold Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. *Proc. ICCV*, 2013.

[Gehler and Nowozin, 2009] Peter Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. *Proc. CVPR*, 2009.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. CVPR*, 2014.

[Gould *et al.*, 2009] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. *Proc. NIPS*, 2009.

[Grauman and Darrell, 2005] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. *Proc. ICCV*, 2005.

[Hillel and Weinshall, 2006] Aharon B. Hillel and Daphna Weinshall. Subordinate class recognition using relational object models. *Proc. NIPS*, 2006.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *Proc. MM*, 2014.

[Johnson and Everingham, 2010] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. *Proc. BMVC*, 2010.

[Jurie and Triggs, 2005] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. *Proc. ICCV*, 2005.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Proc. NIPS*, 2012.

[Kumar *et al.*, 2009] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. *Proc. ICCV*, 2009.

[Kumar *et al.*, 2012] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. J. Kress, Ida Lopez, and João V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. *Proc. ECCV*, 2012.

[Lampert *et al.*, 2008] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. *Proc. CVPR*, 2008.

[Lazebnik *et al.*, 2006] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proc. CVPR*, 2006.

[Li *et al.*, 2011] Congcong Li, Devi Parikh, and Tsuhan Chen. Extracting adaptive contextual cues from unlabeled regions. *Proc. ICCV*, 2011.

[Liu and Belhumeur, 2013] Jiongxin Liu and Peter N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. *Proc. ICCV*, 2013.

[Liu *et al.*, 2014] Jiongxin Liu, Yinxiao Li, and Peter N. Belhumeur. Part-pair representation for part localization. *Proc. ECCV*, 2014.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Malisiewicz *et al.*, 2011] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. *Proc. ICCV*, 2011.

[Matthews and Baker, 2004] Iain Matthews and Simon Baker. Active appearance models revisited. *IJCV*, 2004.

[Milborrow and Nicolls, 2008] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. *Proc. ECCV*, 2008.

[Nilsback and Zisserman, 2008] Maria E. Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *Proc. ICVGIP*, pages 722–729, 2008.

[Parkhi *et al.*, 2012] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. *Proc. CVPR*, 2012.

[Pishchulin *et al.*, 2013a] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. *Proc. CVPR*, 2013.

[Pishchulin *et al.*, 2013b] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Strong appearance and expressive spatial models for human pose estimation. *Proc. ICCV*, 2013.

[Ramakrishna *et al.*, 2014] Varun Ramakrishna, Daniel Munoz, Martial Hebert, J. Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. *Proc. ECCV*, 2014.

[Ramanan, 2006] Deva Ramanan. Learning to parse images of articulated bodies. *Proc. NIPS*, 2006.

[Ren and Ramanan, 2013] Xiaofeng Ren and Deva Ramanan. Histograms of sparse codes for object detection. *Proc. CVPR*, 2013.

[Rowley *et al.*, 1998] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE TPAMI*, 20:23–38, 1998.

[Sadeghi and Farhadi, 2011] Mohammad A. Sadeghi and Ali Farhadi. Recognition using visual phrases. *Proc. CVPR*, 2011.

[Sapp and Taskar, 2013] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. *Proc. CVPR*, 2013.

[Saragih *et al.*, 2009] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Face alignment through subspace constrained mean-shifts. *Proc. ICCV*, 2009.

[Sermanet *et al.*, 2014] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *Proc. ICLR*, 2014.

[Song *et al.*, 2011] Zheng Song, Qiang Chen, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. *Proc. CVPR*, 2011.

[Su and Jurie, 2011] Yu Su and Frédéric Jurie. Visual word disambiguation by semantic contexts. *Proc. ICCV*, 2011.

[Sun and Savarese, 2011] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. *Proc. ICCV*, 2011.

[Sun *et al.*, 2013] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. *Proc. CVPR*, 2013.

[Szegedy *et al.*, 2013] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. *Proc. NIPS*, 2013.

[Tompson *et al.*, 2014] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Proc. NIPS*, 2014.

[Toshev and Szegedy, 2014] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *Proc. CVPR*, 2014.

[Tsochantaridis *et al.*, 2004] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *Proc. ICML*, 2004.

[Vedaldi and Fulkerson, 2008] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[Vedaldi *et al.*, 2009] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. *Proc. ICCV*, pages 606–613, 2009.

[Vidaldi and Zisserman, 2011] Andrea Vidaldi and Andrew Zisserman. Image classification practical. `http://www.robots.ox.ac.uk/~vgg/share/practical-image-classification.htm`, 2011.

[Viola and Jones, 2001] Paul Viola and Michael Jones. Robust real-time object detection. *IJCV*, 57(2):137–154, 2001.

[Vondrick *et al.*, 2013] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. *Proc. ICCV*, 2013.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Computation & Neural Systems Technical Report*, CNS-TR-2011-001, 2011.

[Wallraven and Caputo, 2003] Christian Wallraven and Barbara Caputo. Recognition with local features: the kernel recipe. *Proc. ICCV*, 2003.

[Wang and Li, 2013] Fang Wang and Yi Li. Learning visual symbols for parsing human poses in images. *Proc. IJCAI*, 2013.

[Wang *et al.*, 2009] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. *Proc. CVPR*, pages 3360–3367, 2009.

[Wang *et al.*, 2011] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. *Proc. CVPR*, 2011.

[Xie *et al.*, 2013] Lingxi Xie, Qi Tian, Richang Hong, Shuicheng Yan, and Bo Zhang. Hierarchical part matching for fine-grained visual categorization. *Proc. ICCV*, 2013.

[Yang and Ramanan, 2011] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. *Proc. CVPR*, 2011.

[Yao *et al.*, 2011] Bangpeng Yao, Aditya Khosla, and Li Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. *Proc. CVPR*, 2011.

[Yao *et al.*, 2012] Bangpeng Yao, Gary Bradski, and Li Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. *Proc. CVPR*, 2012.

[Yin *et al.*, 2011] Qi Yin, Xiaoou Tang, and Jian Sun. An associate-predict model for face recognition. *Proc. CVPR*, 2011.

[Zhang and Viola, 2007] Cha Zhang and Paul Viola. Multiple-instance pruning for learning efficient cascade detectors. *Proc. NIPS*, 2007.

[Zhang *et al.*, 2012] Ning Zhang, Ryan Farrell, and Trever Darrell. Pose pooling kernels for sub-category recognition. *Proc. CVPR*, 2012.

[Zhang *et al.*, 2013] Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. *Proc. ICCV*, 2013.

[Zhang *et al.*, 2014a] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. *Proc. ECCV*, 2014.

[Zhang *et al.*, 2014b] Ning Zhang, Manohar Paluri, MarcAurelio Ranzato, Trevor Darrell, and Lubomir Bourdev. Panda: Pose aligned networks for deep attribute modeling. *Proc. CVPR*, 2014.

[Zhou *et al.*, 2010] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. *Proc. ECCV*, 2010.

[Zhou *et al.*, 2013] Feng Zhou, Jonathan Brandt, and Zhe Lin. Exemplar-based graph matching for robust facial landmark localization. *Proc. ICCV*, 2013.

[Zhu and Ramanan, 2012] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. *Proc. CVPR*, 2012.

[Zhu *et al.*, 2010] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. *Proc. CVPR*, 2010.