University of California

Los Angeles

An Automated Perceptual Learning Algorithm for Determining Structure-Based Visual Prototypes of Objects from Internet-Scale Data

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Lichao Chen

2015

Abstract of the Dissertation

An Automated Perceptual Learning Algorithm for Determining Structure-Based Visual Prototypes of Objects from Internet-Scale Data

by

Lichao Chen
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2015
Professor Vwani P. Roychowdhury, Chair

Object discovery and representation lies at the heart of computer vision, and therefore it has attracted widespread interest in the past several decades. Early efforts were largely based on single template models, bag-of-visual-word models, and part-based models. To represent the intra-class variety of the same type of object and address partial occlusion problem in images, more complex object representations, like attribute-based and part-based models, have been proposed. The advent of the Internet, however, enables one to obtain a comprehensive set of images describing the same object as viewed from different angles and perspectives, and its natural association with other objects. This opens up new opportunities and challenges: Given that for the first time we have millions of exemplars of an object embedded in its natural context, can one effectively mimic human-like cognition and build up prototypes (comprising parts, their different views, and their spatial relationships) for each object category? The well-known supervised approach relies heavily on well labeled image datasets and it (i) is still prohibitively hard for image labeling to catch up with the speed of image crawling, and (ii) does not lead to succinct prototype models for each category, which can then be used to locate object instances in a query. In my dissertation, we investigated the open problem of constructing part-based object representation models from very large scale image databases in an unsupervised manner.

To achieve this goal, we first define a network model from a full Bayesian setting. This augmented network model has spatial information in it, and is scale invariant throughout any image resolution variations in the learning set. This network model is able to find visual templates of the same part with dramatically different visual appearances, which, in existing models, have to be added manually or using text information from the Internet. We show that the global spatial structure of the underlying and unknown objects can be restored completely from the recorded pairwise relative position data. We also developed an approach to learn the graphical model in a completely unsupervised manner from a large set of unlabeled data, and the corresponding algorithm to do detection using the learned model. We also apply our algorithm to various crawled and archived datasets, show that our approach is computationally scalable and can construct part-based models much more efficiently than those presented in the recent computer vision literature.

The dissertation of Lichao Chen is approved.

Russel E. Caflisch

Alan J. Laub

Lieven Vandenberghe

Vwani P. Roychowdhury, Committee Chair

University of California, Los Angeles

2015

To my family, friends and teachers . . .

Table of Contents

List of Figures

List of Tables

Acknowledgments

First I would like to to express my sincere gratitude to my advisor Professor Vwani Roychowdhury. Thank you for your generous support and mentorship, which are invaluable to my career and life. This thesis would not have been possible without your help, support and patience through numerous discussions. Thank you for showing a model of an excellent scientist, visionary and information sleuth, which I will always be trying to achieve in my life. I am extremely grateful.

I also would like to acknowledge the members of my doctoral committee, Professor Lieven Vandenberghe, Professor Alan Laub and Professor Russell Caflisch for taking the time to serve on my doctoral committee, read my thesis, and provide advice on my work.

I am glad to have met many good friends in our Lab: Dr. Oscar Boykin, Dr. Qian Zhong, Dr. Chih-Li Wang, Dr. Nima Sarshar, Dr. Sudhir Kumar Singh, Dr. Joseph Kong, Dr. Zicong Zhou, Dr. Hai Qian, Dr. Roja Bandari, Misagh Falahi, and Eshan Ebrahimzadeh. I will always remember the interesting things and ideas we discussed together in these years, and I really appreciate your help and inspiration to make me a better scientist.

I also thank all of my friends who I met in UCLA: Jinchao Li, Jiadong Wang, Jianshu Chen, Ziwei Li, Juehua Yu, Linglin He, Chenni Qian, Hongjing Xia, Yuxi Zhao, Yan Jin, Lingyan Ruan, Hoshun Yang and many more. Thank you for sharing the journey with me throughout these years.

Last, and most importantly, my deepest appreciation goes to my parents, and my wife, Yizhe Bu: Thank you for your unconditional support, love, and for all the sacrifices that you have made for me. I would not be able to finish this work without your love and support.

Vita

2005-2009	B.S. (Computer Science), Fudan University, Shanghai, China.
2009-2010	M.S. (Electrical Engineering), UCLA.
2010–2012	Teaching Assistant, Electrical Engineering Department, UCLA.
2009–2015	Research Assistant, Electrical Engineering Department, UCLA.

CHAPTER 1

Introduction

1.1 Motivation from Cognitive and Neuroscience Findings

Visual object classification and recognition is of fundamental importance to almost all animals, and the evolutionary process has made the underlying systems highly sophisticated and refined, enabling abstractions and specificity at multiple levels of the perception hierarchy. Design of unsupervised, scalable, and accurate computer vision (CV) systems, inspired by principles gleaned from biological visual processing systems, has been a cherished goal since the inception of the field. Recent success of the Deep Neural Network (DNN) framework in computer vision can largely be attributed to its layered locally-connected architecture comprising sigmoidal elements (an abstraction of neurons), which mimics the organization of visual cortex [Ben09,Hin07,LBD89,ARK10,BCV13,Sch14]. In the layered cortex each processing stage copes with increasingly abstract representations [WAH92, WAH93, WAH97, RP99], enabled by extraction of features over increasingly-larger receptive fields. There is some computational evidence to suggest that DNNs exhibit similar automated extraction of features at various scales and resolutions [Ben09, Hin07]. The deep learning framework is undoubtedly a significant achievement, and seems to outperform more conventional classifiers driven by hand-crafted features (such as SIFT and HOG [Low99, BTV06, DT05, DTS06]) in several object classification and recognition tasks [BCV13, Hin07, Den13, DLH13]. In fact, the features it automatically discovers via its layered architecture are considered to be its primary advantage, and DNNs are increasingly being regarded as a framework for generating rich and relevant feature sets for objects.

It is, however, widely acknowledged that DNNs performance is nowhere comparable to that of biological systems and it seems to suffer from several limitations not observed in human visual systems, including (i) very high computing costs, often requiring a few thousand processors running for several days to train [GCC11,Ham93,ARK10,BCV13,Sch14], (ii) the need to avoid local minima during the training process, and (iii) the fact that for extracting the best performance out of DNNs, one must train it in a supervised manner, supported by a large training set with cropped and labeled datasets containing the object to be learned. The biological vision systems, on the other hand, are autonomous and unsupervised, can create models for objects based purely on familiarity and repeated visual exposures, can represent such learned objects at various scales and resolutions, and the underlying learning process is highly computationally efficient.

Clearly, several aspects of the potential synergy between biological and CV systems remain unexplored, and identification of characteristics of the human visual system that go *beyond multi-layered perceptron architectures* and that can be abstracted and *computationally integrated into a machine learning system* remains a topic of considerable ongoing interest. The primary goal of this thesis is to *identify* key organizational aspects of biological vision systems, and then *use them to design machine learning systems* for object representation and detection that display several desired properties such as, *unsupervised learning capability, computational scalability,and robustness to transformations and scenarios, such as scaling, occlusion and different views of the same three-dimensional object.* The immediate goal is not to emulate the exact granular hardware and feature generating building blocks of the visual system, such as neurons and their layered interconnections, but to incorporate more abstract aspects into a computational framework, such as how visual memory is structured, what role structure plays in detection and recognition, and the salient characteristics of the underlying learning process itself. The emphasis is on ensuring that the resulting computer vision system is computationally efficient, and exhibits robustness and performance metrics that are not achieved by the existing systems.

1.2 Key Inspirations from the Human and Other Mammalian Visual Systems

Increasing evidence from the fields of cognitive psychology, neuropsychology, and neurophysiology [LAM09, Kre07, LS96], indicates that the diversity of tasks that any biological recognition system must solve dictates that object recognition is not a single, general purpose process. Detailed data, based on both invasive and noninvasive probes, from a varied set of subjects including, normal adults, infants, animals, and brain-damaged patients reveal a complex interacting system, where functionalities such as classification of objects at a basic category level (commonly referred to as the *object detection/classification* problem in the CV community) and the identification of individual objects from a homogeneous object class (commonly referred to as the *recognition* problem in the CV community) utilize distinct pathways built on a shared set of features and primitives. For example, in certain patients with prosopagnosia, they retain the ability to identify classes of objects such as cars, phones, wallets etc. but cannot recognize their own items from members of the same class [BCP04, GGC08]. Among the many such properties, we next highlight the ones that we aim to incorporate in our CV system.

- **Many-to-Many relationships between stimuli and neuronal activities: No "grandmother neurons" tuned to specific objects.** It is being widely mentioned by DNN or DBN studies, that certain brain modules are devoted to recognizing objects of a particular class, which is comparable with the "best neurons" reportedly found at the output layer of the DNN [GCC11, BCV13]. In many early medical and neuroscience papers [GM04, Gro02], such selectivity was reported to be observed, and the existence of "grandmother" neurons was suggested based on such perceived observations. More recent work, however, for example by Liu et al. [LAM09], pointed out that such observations most likely do not imply the existence of specialized brain areas devoted to recognizing certain classes of objects. In particular, Liu et al. [LAM09] argues that (i) Most of the observed selectivity of neuronal response is for the particular category of human faces, which is of obvious evolutionary importance to us, and hence, it would not be surprising to have dedicated circuitry for such key objects. Very few object categories other than faces have been explored in the literature, and (ii) Most experiments in humans are limited in scope in the sense that access to only specific regions of the brain is available and one cannot explore the entirety of the brain regions responsible for vision and understanding. It is highly likely that many neurons get activated for the same stimulus. In fact, recent analyses suggest that each neuron is likely to respond to many different classes of objects and that each individual class may be represented also by many neurons [WKQ06, QRK05, Kre07].

- **Categories are represented by Prototypes obtained in an unsupervised manner from a large enough set of exemplars.** Several experiments, including one reported by Marsolek [Mar95], suggest that there are two separate visual systems in brain: one works at the category level, which is used to classify different instances as belonging to the same abstract category, the other is for distinguishing instances within the same category using visual details of the object. As a result, in the process of learning visually novel categories that are different to any known objects or concepts, studies in neuroscience suggests that, when the training set is small, humans tend to memorize visual details of every single exemplars, while when the training set is large enough, they will be able to detect stable correlational features, which occur most often through members of the same class, while maximizing differences from instances of other classes [HE73, PK68, LS96]. This kind of features are also referred to as features of high cue validity. It is also shown that such sets of features, which are shared by most of the category members, were found to be extracted and stored during the learning stage, instead of the recognition stage. Such features characterize most exemplars of an object class, and thereby form class representative prototypes for corresponding categories, which will be used for visual recognition and classification in the future.

- **Objects are encoded as a combination of "parts" and their spatial and**

geometric relationships. Electrophysiological findings show that a particular portion of the brain (Inferotemporal cortex, or IT) appears to meet all the machinery requirement for the formation of part-based object representations. Neurons in the IT responds selectively to stimulus from color, texture, simple structural primitives, to complex views, or even completed objects like faces [DG79, MK80, LS96]. To encode the object representation of different classes, these prototypes are likely to be decomposed into parts. These parts have stable spatial relationships to each other that remain invariant through many different views and are indexed according to their spatial relationships [AGH84]. In detection or classification tasks, parts and structural relationships among them are detected, indexed, and compared with the prototype. Exemplars are recognized as class members if and only if the structural information is close enough to that of prototype [Bie87, BC92, BG93].

- **Parts of objects are not necessarily "functional" parts but multiple different views of the functional parts and combinations.** As been claimed many times in deep learning literature [Hin07,Den13,BCV13], it is tempting to assume the neurons in the brain work in a hierarchical manner, from shape primitive, simple part, complicated pattern to complete object, each processing stage forms increasingly complicated representations. However, evidence from psychophysical and neurophysiological studies indicates that besides the system of recognizing objects by parts and their spatial interrelationships, there should be another system which may represent objects by combinations of multiple views, or aspects [Mar95]. For example, findings suggest that, as in humans, rat invariant recognition can flexibly rely on either view-invariant representations of distinctive object features or view-specific object representations, acquired through learning [RAA15]

  Moreover, the evidence from numerous studies shows that there are some categories of objects which are represented by a small portion of neurons with a complex configurational selectivity directly. These object cannot be reduced to selectivity of individual features or even constellations of such features [LS96]. As a result, the recognition of these object involves the second system, in which the holistic configuration, instead of individual features, is playing the dominant role. Wachsmuth's finding shows that about one fifth of the neurons studied in his experiment fired only to the whole body instead of any of the body parts alone [WOP94]. Oram and Perrett found that while many neurons respond to individual features of the head like eye, nose and lips, there are another population of neurons which can only be fired by co-occurrences of multiple parts [OP94]. Perrett also reported five types of such kinds of neurons, each of which is responsive to one of these stimuli: frontal face, profile face, back head, head up, and head down. In addition, there are also two subtypes that are responsive only to left profile or only to right profile face [PSP85, LS96].

- **Learning is perceptual: interaction with the same object in different configurations.** Imagine how infants learn novel concepts, for example, dogs. Are we feeding them with a bunch of dog images or showing them different kinds of dog with a label "positive" accompanied with other images and stuff with label "negative"? Apparently not! They don't even have the concepts "positive" or "negative". Actually, far before infants gain conceptual categorization abilities, they are already able to refer to all dogs as "wow-wow" through perceptual learning [EQ94]. Palmer and Rosch suggested that conceptual categories defined in human society actually have a perceptual basis, which are determined by the high cue features described in preceding discussion [PK68, PRC81]. Such studies in neuroscience show that the perceptual learning plays a more important role than the conceptual, linguistic labels, and the abstraction power we have, which makes classification/detection task possible, is not rooted on linguistic development but perceptual learning of correlated structure of the world. Similar result was also obtained from incidental task sequence learning studies [CM07].

- **Object detection and recognition process utilizes the same memory representation framework as of visual objects, i.e., it tries to find the parts and views (*it has already learned*) in the scene and then make a decision based on how well they fit together.** The recognition process in human brain is dramatically different from those *discriminative* approaches developed in the Computer Vision community. Instead, it has a close relation with memory representation [RP99]. Experiments in early studies have shown that once a category is learned, humans can recognize the prototype in a fast and accurate way, though the prototype is different from all the members that were directly presented to them, which makes the idea that the prototype forms the category's memorial representation theoretically appealing [PK68, FB71]. Recently, Electrophysiological findings also suggests that the medial temporal lobe (MTL), which is widely believed to be the area of visual perception, might also have a role in memory trace formation, consolidation and information retrieval [Kre07], including longer latencies of human MTL neurons than that of immediate visual object recognition [SKK07], extensive evidence from molecular experiments [Eic04], memory disruption caused by electrical stimulation in the MTL [HWS85].Moreover, Kreiman also shown that even the subject made an incorrect behavioral response, a statistical classifier can still tell whether a stimulus was familiar or not from response of a population of neurons, which is likely to mean that these neurons are not involved in the decision process but directly related to the actual memory representation [Kre07].

1.3 An Overview of the Structural Unsupervised Viewlets (SUV) Model

At the current stage of our scientific understanding, and especially given the capabilities and limitations of extant digital computers and computing hardware, we cannot hope to incorporate every aspect of the highly complex human or mammalian visual system. In this study, however, we retain the essence of most of the findings highlighted in the preceding section, and *design a CV framework* for object recognition that is *computationally tractable*, has a *rigorous statistical and mathematical foundation*, uses only *simple computational steps* and yet *can perform at par or better than most other highly-tuned and supervised CV algorithms*, as verified via performance on different test datasets. As reviewed in detail in Chapter 3, the CV community has a rich history of incorporating some of these features, but a unified framework that has exploited all the following characteristics does not seem to have been explored. We introduce a CV system that we refer to as the Structural Unsupervised Viewlets (SUV) Model, and different aspects of the model are outlined in the following.

1.3.1 Object Prototypes Based on Views and Spatial Relationships

The idea centers around the concept of a *prototype* of an object, as introduced in the preceding section. That is, for any object category with many examples, the biological memory system seems to build a prototype, or an average model which captures different parts and their combinations, along with their many configurations, and a flexible relative-location model that describes how they are spatially configured. Inspired by this abstraction, we build a Markovian Random Field (MRF) model for object representations (see Section 2.2), where

1. **Nodes** represent different views of parts and their various configurations. In such a prototype representation, each view might correspond to an actual functional part of the object, such as the head, arms and its different configurations, legs etc. of a human, or could be a combination of such parts configured in a particular manner, e.g. a half-body view of a model with arms akimbo. We refer to each node as a **viewlet**, and an object of interest could be represented by a collection of hundreds of such viewlets. From a computational representation perspective, each node is a collection of visual features, which characterize the viewlet represented by the node. The SUV model is by design agnostic as to the choice of the exact features, as long as they are discriminative enough and can be used as the basis of an accurate detector. For example, it could be features extracted from a DNN framework or a HOG or SIFT feature set. In order to show the power of our approach, we chose to represent each such viewlet as a rectangular patch of fixed dimensions, and use a high-dimensional HOG

feature set, as explained in Section 4.3. We note that a more sophisticated feature set could be, and perhaps should be, substituted for the HOG features, and we expect the performance to improve significantly. Our goal, however, is to show the power of the framework we have developed, and demonstrate that the system exhibits superior performance even with a relatively coarse set of visual features.

2. **Edges** represent conditional dependence, i.e., the relative location of a particular viewlet is determined (via a distribution) by the locations of the viewlets that it is connected to, and is statistically independent of the locations of other viewlets that it does not share any edge with. Thus, given the locations and scale information of the viewlet nodes connected to a particular viewlet, there is a tight distribution associated with the location and scale of the viewlet under consideration. If the distance distribution is jointly Gaussian, then the network represents the connectivity pattern of the inverse of the Covariance matrix, also referred to as the precision matrix, and our model becomes a Gaussian Markov Random Field model. The SUV model expects and exploits the fact that the MRF model is sparse and the number of edges in the MRF network grows linearly in the number of viewlets.

3. **An exclusivity relationship** among the viewlets or the nodes of the network, that allows the SUV model to succinctly represent *deformable objects*. In particular, such exclusivity relationships capture the fact that certain parts of the object category under consideration might be configured in multiple ways, and the visual appearances for these different configurations might be radically different. The corresponding viewlets then cannot occur together in the same image instance. For example, the arm could be positioned straight down or bent at the elbow. The corresponding viewlets (at multiple scale) will never occur simultaneously, but will share similar relative locations (i.e., similar edges and distributions) with respect to viewlets corresponding to other parts (e.g., viewlets that capture the head and legs). As explained in Section 4.4, this network model allows one to capture the so-called mixture models for representing object categories with deformable parts in one unified framework.

1.3.2 An Unsupervised Framework That Only Learns from Positive Examples

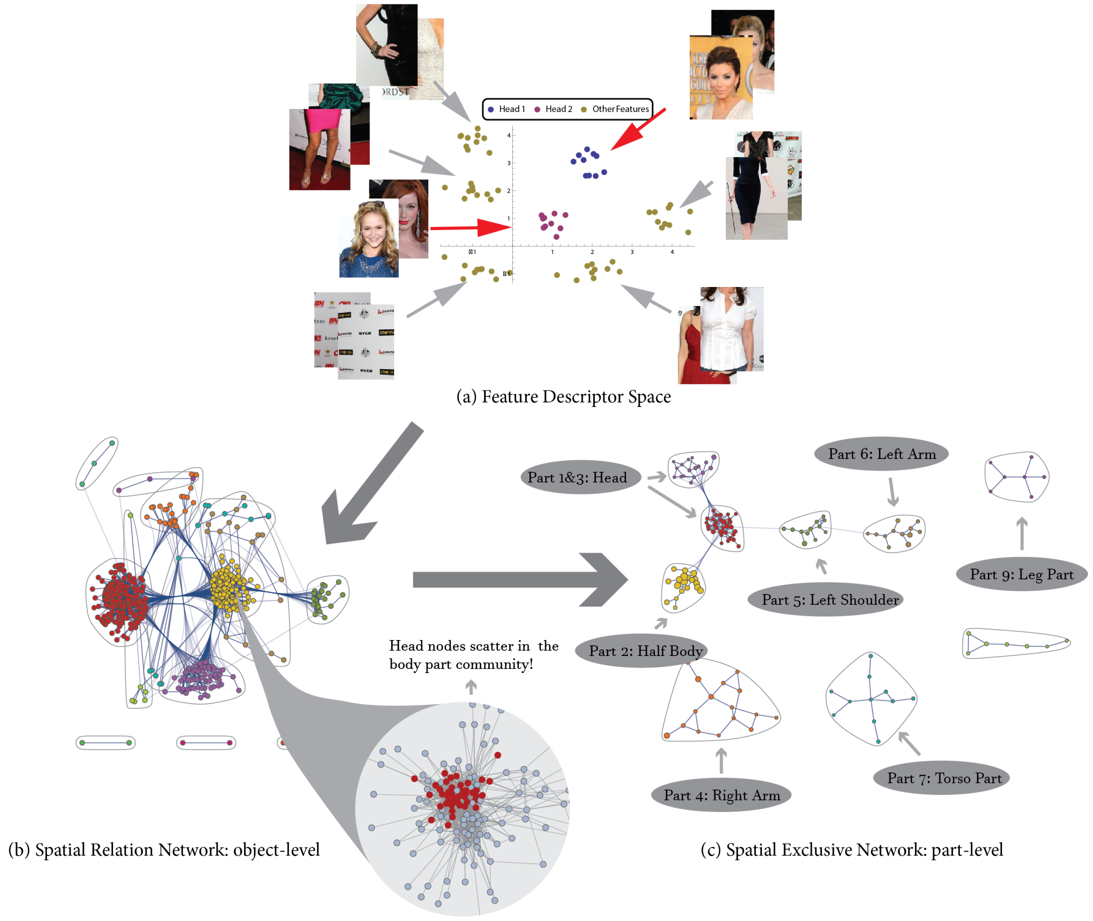
Most CV algorithms rely on discriminative learning, where positive examples of the object category to be learned are given along with a set of negative examples, comprising images that do not have instances of the object category to be learned. Such a supervised learning framework, while highly effective in most constrained use cases (e.g., where the negative training set is comprehensive), suffers from a number of well-known drawbacks including, (i) the need to already know the object before it has been discovered, the need for extensive manual labeling, (ii) the risk of over-fitting to the training dataset and not being able to generalize to new instances, and (iii) the difficulty in coming up with a bag of classifiers (for example, a comprehensive set of associated training sets) when the object category comprises exemplars or instances with very different visual appearances. While these are some many practical concerns with supervised methods, we avoid the supervised learning approach in this study primarily from an "aesthetic" perspective: The discriminative approach focuses more on what an object category is not, rather than what it is; objects are represented only in contrast to other objects rather than a model of its own. While it has proven successful, it requires specialized scripting and adopts a perspective that is difficult to scale. One of our goals is to show that it is possible to learn an object category in a totally unsupervised (that is from contextual information only) and computationally tractable manner, which necessarily implies that we do not rely on negative examples. As our results indicate, it is possible to do so for certain objects, and as long as one has high-enough resolution in the features space, it performs better than, or as well as, some of the best supervised systems currently being used in the space.

1.3.3 Perceptual Learning

As already noted, several experiments seem to confirm that infants are able to come up with the concept of categories of objects and categorical representations solely from the scenes they witness. In biological systems certain discrimination processes are involved in refining concepts by means of the repeated presentation of exemplars. Learning of the SUV models are inspired by this paradigm. Instead of learning the SUV model parameters from a labeled set of exemplars (as in say, providing a bounding box around human figures in the learning image set), we consider the following scenario: Given a large and completely unlabeled image corpus featuring multiple instances of an *unknown* but a specific set of *structured objects* (i.e., each object has several parts that have consistent and stable spatial relationships), how can one automatically discover and build composite abstract representations of the underlying unknown objects, their different parts (i.e., the groups of viewlets), and the relative spatial positions of the parts (i.e., how the "pieces" fit) within an object. Thus, the learning paradigm does not assume the knowledge of any known categories, and instead aims to discover object categories in an unsupervised manner. The models are to be learned not only in a non-discriminative manner, but also perceptually, i.e. by finding structures, both visual and spatial, that go together. The only requirement for such an approach to succeed is that the given image corpus should have enough number of instances of the object category to be learned that cover the different configurations.

1.3.4 From Models To Detection

In accordance with our neuroscience inspirations, the task of object detection in our SUV framework closely follows the model representation process itself. The detection algorithms first look for the viewlets that it can detect in the image under consideration, and then determine if the detected viewlets together comprise a reliable set of exemplars. That is, we look for the best groupings of the likely viewlets, where the quality of a grouping is determined by the likelihoods of how they are placed and their relative scales, as encoded in the SUV model. This heuristic search algorithm is agglomerative in nature rather than exhaustive, making it highly scalable. Moreover, this natural detection framework allows one to find multiple occurrences of objects in the same image efficiently.



(a) Feature Descriptor Space

(b) Spatial Relation Network: object-level

(c) Spatial Exclusive Network: part-level

Figure 1.1: The different steps in the SUV framework. See the description of Figure 4.2 for more details.

1.4 Outline of the Dissertation

This thesis targets the learning of structure-based visual prototypes from Internet-scale data. Our focus is on developing a neuroscience-inspired network-based framework that is able to learn object perceptually, i.e. in an unsupervised manner, and without using any kind of labeled negative/background data.

In **Chapter 2**, we present a network model along with learning and detection methodologies. Firstly, we construct a full Bayesian model, which is then simplified into a network that encodes each visual template as a node and their spatial dependencies as edges. The network also has a scale attribute for each node to capture the real world scale of the corresponding visual template. Based on this scale parameter on vertices, a relative spatial relation is derived on the edges, which is scale invariant and allows one to uniformly process images of different resolutions. Consequently, our model is scale and translation invariant. For global shape information modeling, instead of picking one part as a reference point to represent the absolute position of the rest, only the pairwise relative spatial information is required.

In **Chapter 3**, we briefly cover the related literature in learning and detection in computer vision, and then expand on work related to the part-based approaches, which we later use in Chapters 4 and 5 for comparison and benchmarking purposes. Considering the key usage in our framework of results from the complex network and community finding literature (mostly for unsupervised clustering), we also introduce the related concepts.

In **Chapter 4**, we address the human prototype learning problem, which is considered to be one of the most challenging problems in the CV community. The human body has several deformable parts and depending on the point and angle of view, the 2D images can have a wide variability in visual features. Moreover, with the advent of the social media giants such as Facebook and Twitter, there is an added interest in being able to create accurate and predictive CV models of the human body. Firstly, we create a high quality celebrity image set in which the frontal faces are not always present. We then implement our graphical model in a fully unsupervised way on this celebrity image set, and the corresponding algorithms to do detection using the learned model as described in **Chapter 2**. With this network, we can find structurally similar nodes [KFH08, BGH04], and because of our augmented edge with relative spatial information, the spatial similarity was also verified. This enables us to find k-partite like sub-graphs in our network, which establish the connection between visual templates of the same part (like arm, torso, head, etc.) with dramatically different appearances or visual features. Traditionally, such hidden connections between common parts and their different configurations, have either been added manually, or inferred using text information from the Internet [KHE12]. We also show that the global spatial information

Table 1.1: Torso detection performance

	Approaches		
	DPM ([FMR08])	Poselet ([BMB09])	Ours
True Positive	1239	3115	2810
False Positive	5263	1678	108
Coverage	38.3%	96.3%	86.9%
Precision	19.1%	65.0%	96.3%

can be restored completely from the recorded pairwise data. In the experiment section, instead of the network-based grouping strategy described in Section 2.3, we developed a simplified anchor-based grouping strategy, and use the network-deduced transform rules to infer the position of head/torso from detected parts. First we show that for the task of face detection our model, while learned without using any annotated data, significantly outperforms the Viola-Jones face detector, which is commonly used in the CV community and is the result of supervised and highly-tuned training efforts. Second, we consider the much more challenging task of torso detection, and show that we perform much better than a state-of-the-art part-based approach called the Deformable Part Model (DPM) and have comparable result with the so-called poselet approach (which was trained in a heavily supervised way and uses a bag of templates), as shown in Table 1.1.

In **Chapter 5**, we further study the model by learning multiple objects simultaneously from a data set referred to as the Caltech-101 dataset [FFP06]. In this implementation, we implement the network-based grouping described in Section 2.3 for detection. The network-based grouping eliminates the need of picking anchor nodes and extracting transform rules, which makes the model a fully automated approach. What's more, it addresses the multi-object-in-one-image problem natively, and also offers a reliable confidence measure of the predication (number of semantic parts covered by the group). To learn models from the 4 categories (airplane, car, face, motorbike), depending on whether a shared dictionary is used or separate ones are used, we design 2 different cases: In the shared dictionary experiment, we use patches from all categories to construct an 800-word visual dictionary, based on which the network for each category is learned as described in **Chapter 2**. Then, SVM classifiers are trained using the confidence scores from learned model as inputs. The resulting classifier outperforms a part-based approach introduced by Fergus et al. [FPZ07] and described in some detail in Chapter 3. Even though the approach in [FPZ07] incorporates elements of discriminative training (e.g. while learning the model for category A, they use images from other categories F,M, and C as negative samples), the inherent flexibility of our model in capturing structure more than compensates for the lack of discriminative information. In the separate dictionary experiment, we demonstrate that even when the images shown to our algorithm are beyond experience (i.e. it is fed with regions in the feature space not covered by its visual words), the structure information alone can still be extracted and abstracted to form discriminative prototype representations. The resulting detectors outperform the counterpart work [FPZ03] significantly, as show in Table 1.2.

Finally, Chapter 6 outlines some potential extensions of the work presented in the thesis. The approach introduced in this thesis, and particularly Chapter 2, can be generalized in several directions allowing one to build effective models for highly-deformable objects and also improve both accuracy and coverage of detection.

Table 1.2: Confusion table for separated dictionary model on the Caltech-101 dataset

	Our Model				Fergus et al. [FPZ03]			
Query Image	F	M	A	C	F	M	A	C
Face	0.98	0.069	0.215	0.252	0.964	0.33	0.32	-
Motorbike	0	0.95	0.370	0.237	0.50	0.925	0.51	-
Airplane	0	0.007	0.665	0.025	0.63	0.64	0.902	-
Car	0	0	0.002	0.600	-	-	-	-

CHAPTER 2

Methodology

Any visual system for recognizing object categories has three interacting parts to it, namely, a flexible model for representation, efficient algorithms to learn the model parameters, and detection algorithms that analyze a given image to locate objects. We already provided a descriptive overview of all the three parts for our system in the introductory chapter and here we provide a rigorous exposition.

2.1 The Structural Unsupervised Viewlets (SUV) Model

The first challenge is coming up with models that are flexible and yet precise enough to capture the "essence" of a category, i.e. what is common to the objects that belong to it, and can accommodate object variability, e.g. presence/absence of distinctive parts such as mustache and glasses, variability in overall shape, changing appearance due to lighting conditions, viewpoint etc.

2.1.1 Markov Random Fields and a Scale and Translation Invariant Network Model

The model comprises *viewlets* (as introduced in Chapter 1) which are visually distinct views of different parts and their configurations that are representative of exemplars from the object category. Each viewlet is thus represented by a unimodal distribution in an associated *appearance* feature space. Thus, a viewlet node V_i is associated with an appearance feature vector random variable A_i which is drawn from a distribution $\mathcal{N}(\mu_f, \Sigma_f)$ in $\mathcal{R}^{|f|}$, where $|f|$ is the dimension of the feature space. The specific feature sets used in the thesis are described in Sections 4.6, 5.2.2 and 5.2.3

To model the spatial relationships among the viewlets in a distributed and translation invariant manner, we look at the relative difference in locations of viewlets in a pairwise manner, i.e., we model $X_i - X_j$ for pairs of viewlets V_i and V_j. Here X_i and X_j represent the location parameters of the respective viewlets. For example, in this thesis each viewlet V_i is represented by a rectangular patch of fixed width w, and fixed height h along with a relative scale parameter of S_i (again a random variable). In this context, X_i represents the coordinates, (x_i, y_i), of the top-left corner of the rectangular patch. The best way to visualize the relative scale parameter, S_i, is to imagine an overall scale parameter s for an exemplar in a given image; s determines the overall size of the object, as measured in pixels, and as rendered in the particular image under consideration. In such a scenario, the viewlet V_i is expected to have a width of $s_i^{(x)} = w * S_i * s$ pixels and a height of $s_i^{(y)} = h * S_i * s$ pixels; correspondingly the viewlet V_j is expected to have a width of $s_j^{(x)} = w * S_j * s$ pixels and a height of $s_j^{(y)} = h * S_j * s$ pixels. This pairwise relative distance and scale model has multiple advantages, and in particular avoids the use of a single landmark viewlet V_1 and then calculating all the relative positions and scales with respect to X_1 and s_1. Clearly, having such a “star” dependence on a single node makes the modeling as well as detection and learning processes less robust.

Next, to make a part-based approach truly scale-invariant, we must handle not only the scaling of each individual part, but also how the relative scales influence relative positions of pairs of parts or viewlets. During detection, there is a deviation of the detected position from the true position of the viewlet. This deviation or noise is usually caused by the local variance tolerance of different features which are usually computed from a fixed-size patch that is obtained by sub-sampling a larger region in the original image. Thus, the noise in detection and estimation of the pixel position of a viewlet gets multiplied by its scale. Considering a simple example, given the ground-truth of viewlets (x_1, y_1, S_1), (x_2, y_2, S_2), (x_3, y_3, S_3), the detected X-axis values can be represented as $x_i^d = x_i + N_i \quad (i = 1, 2, 3)$, where N_i is a random noise term with its norm or standard deviation being proportional to the actual width in pixels, $s_i^{(x)}$. So when we are using the same instrument to measure the relative distance, $x_i - x_j$, the standard deviation of the absolute error $|N_i - N_j|$ is proportional to the sum $s_i^{(x)} + s_j^{(x)}$. Next, since each relevant pair contributes to the overall likelihood or probability function of the entire object instance, it is desirable to normalize the measured distances (in pixel values) by their overall variances so that the contributions are independent of scale and hence, comparable. This point is further demonstrated in Figure 2.1. As a result, for each interaction, we are using the scale of both viewlets to do the normalization, i.e., using $\left(\dfrac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}}, \dfrac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}}\right)$ as the metric for a scale and translation invariant distance measure.

We are now ready to define the distributions that characterize the variations allowed for in the exemplars belonging to the object category. In order to do so, we adapt the well-known spring model for our purposes, where each pair of viewlets V_i and V_j is connected via a spring of stiffness parameter $c_{ij} \geq 0$. Thus, if the zero-stress normalized separation between the viewlets is μ_{ij}, then by Hooke's law, the potential energy of the spring corresponding to locations x_i and x_j is given as $\left(c_{ij}\left|\dfrac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}} - \mu_{ij}\right|^2\right)$. Further, assuming an isotropic spring model, where the displacements along the X-axis and Y-axis are treated separately and independently, the total potential function of the given configuration is given by $P = P(\mathbf{X}) + P(\mathbf{Y})$, where

$$P(\mathbf{X}) = \frac{1}{Z^{(x)}} e^{-f_x(\mathbf{x})} \tag{2.1}$$

$$f_x(\mathbf{x}) = \sum_{i \neq j} c_{ij}^{x} \left(\frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}} - \mu_{ij}^{x}\right)^2 \tag{2.2}$$

$$P(\mathbf{Y}) = \frac{1}{Z^{(y)}} e^{-f_y(\mathbf{y})} \tag{2.3}$$

$$f_y(\mathbf{y}) = \sum_{i \neq j} c_{ij}^{(y)} \left(\frac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}} - \mu_{ij}^{(y)}\right)^2 \tag{2.4}$$

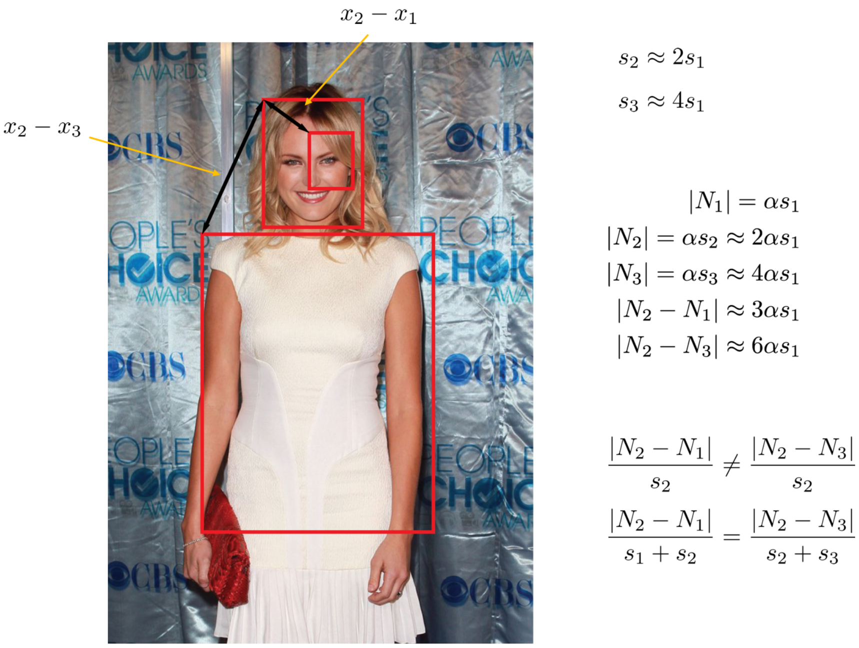

$x_2 - x_1$
$x_2 - x_3$
$s_2 \approx 2s_1$
$s_3 \approx 4s_1$
$|N_1| = \alpha s_1$
$|N_2| = \alpha s_2 \approx 2\alpha s_1$
$|N_3| = \alpha s_3 \approx 4\alpha s_1$
$|N_2 - N_1| \approx 3\alpha s_1$
$|N_2 - N_3| \approx 6\alpha s_1$
$\frac{|N_2 - N_1|}{s_2} \neq \frac{|N_2 - N_3|}{s_2}$
$\frac{|N_2 - N_1|}{s_1 + s_2} = \frac{|N_2 - N_3|}{s_2 + s_3}$

Figure 2.1: Relative distance normalization between 2 different pairs: *Head* ↔ *Eye*, *Head* ↔ *Torso*.

and Z^x and Z^y are the corresponding normalization terms and are also often referred to as the partition functions. For scale $\mathbf{s}$, we note that it is a multiplicative factor, and we take its logarithm and define an analogous potential function,

$$P(\mathbf{S}) = \frac{1}{Z^{(s)}} e^{-f_s(\mathbf{s})} \tag{2.5}$$

$$f_s(\mathbf{s}) = \sum_{i \neq j} c_{ij}^{(s)} (\log \frac{s_i}{s_j} - \mu_{ij}^{(s)})^2 \tag{2.6}$$

Note that if we consider $f_x(x)$, $f_y(y)$, $f_s(s)$ as the quadratic-form potential functions (as in the above equations), then we can regard our model as a *fully connected Gaussian Markov random field*, where each node has a corresponding part in the real system (the mapping can be many-to-one, since a part can have multiple appearances in our model), and instead of specifying the Covariance matrix, Σ, the network model specifies the Precision matrix $\Lambda = \Sigma^{-1}$ via the edges of the network. We start with the X-axis displacement potential function, and will later generalize the result to Y and S. The precision matrix $\Lambda = \Sigma^{-1}$ can be easily calculated from the c_{ij} values, associated with edge, simply by matching the coefficients of the respective product terms $x_i * x_j$. We note that the precision matrix Λ as defined in our model is singular, which is easy to see because of the translation invariance nature of the potential function. For easy deduction, we must fix one value to reduce the degree of freedom to $M - 1$, and without loss of generality, assuming $X_M = 0$, we have,

$$\Lambda_{ii} = \sum_{j=1,j\neq i}^{M-1} \frac{c_{ij}}{(s_i^{(x)}+s_j^{(x)})^2} + \frac{c_{iM}}{(s_i^{(x)}+s_M^{(x)})^2} \tag{2.7}$$

$$\Lambda_{ij} = -\frac{c_{ij}}{(s_i^{(x)}+s_j^{(x)})^2} \quad i \neq j \tag{2.8}$$

It is worthwhile to point out certain properties of the model we have introduced so far: (i) The precision matrix we have defined is an M-matrix, that is all the off-diagonal entries are non-positive and it is diagonally dominant, that is the row sum is positive, and (ii) If $c_{ij} = 0$ then we know from the properties of the multi-variate Gaussian distribution that the corresponding location variables are conditionally independent. That is, given all the x_k's such that $c_{ik} > 0$, x_i is statistically independent of x_j. Hence, by using a spring model we are specifying the Precision matrix directly, and constraining the covariance matrix to be a non-negative matrix (inverse of an M-matrix is a positive matrix).

The number of parameters we need to specify to define the SUV model is the number of non-zero elements in $C^{(x)}, C^{(y)}, C^{(s)}$, where the C's represent the sets of corresponding c_{ij}s. A full multivariate Gaussian model having $\Theta(n^2)$ parameters is still technically difficult to learn [FPZ05]. However, it is easy to see that parts in a real object can not be statistically fully connected with each other. Instead, the true direct interactions tend to be sparse, while indirect ones are conditionally independent given the nodes between them. We expect that the average number of direct connections for a viewlet does not grow with the total number of viewlets. Such direct interactions, combined with node set V, form a network $G(V, E)$, and hence we expect the total number of edges to be $\Theta(n)$ instead of $\Theta(n^2)$. Thus in the learning stage, to determine the edge set E, we must find a sparse maximum likelihood solution, that is, the best $C^{(x)}, C^{(y)}, C^{(s)}$ which optimize the likelihood Equation 2.1, Equation 2.3 and Equation 2.5.

2.1.2 A Probabilistic Interpretation

Our model shares similar basic elements as that of the widely used spring model [FH05], and the probabilistic model introduced in [BWP98, FPZ03, FPZ04, FPZ07, FFP03, FFP06], that is, the model of a object consists of two kinds of information, the appearances of parts, and the structure how parts are organized. However, we make some essential changes to extend them.

In order to learn in a full unsupervised way, we are not keep the model only to object or POI(Point of Interests) and describing background separately. Instead, we break entire images into patches and quantize a vocabulary V of K visual words from all of them. Apparently not all of these visual words are related to the object we interested in. Suppose there is a set $V_f \subset V$, which includes n_f visual templates, which are all about a certain category of object, while the rest $n_b (= K - n_f)$ templates $V_b = V_f^c$, are mainly describing background. Given an image of N patches, each visual template has a chance to be detected in it. In previous part-based approaches, each part is usually modeled using a unimodal representation, and one entry in the presence indicator vector h. To model the general nature of the appearance variety, we allow multiple appearances, so the each part can have more than one visual word. As a result, h becomes a visual word indicator. For a word v_i, $h_i \in \{0, 1, ..., N\}$ is a variable such that $h_i = 0$ while the visual template is missing from image, and $h_i \in \{1, ..., N\}$ while the template is found on a particular location. For probability calculation convenience, as done in [FPZ07], we also define a pure presence indicator $d = sign(h)$, such that, $d_i = 1$ if and only if v_i is present. For all patches in the image, we also define the position X, Y, and the scale S(patch width or height, since they have a fixed ratio), such that, for a visual template v_i, if detected, h_i will be non-zero, and the position and scale will be $X(h_i)$, $Y(h_i)$, and $S(h_i)$. Suppose our model have parameter set θ, give the latent variable h we just defined, we have,

$$
\begin{aligned}
& P(A, X, Y, S|\theta) \\
& \quad = \sum_{h \in \{0,1,..,N\}^{n_f}} P(A, X, Y, S|h, \theta) P(h|\theta) \\
& \quad = \sum_{h \in \{0,1,..,N\}^{n_f}} P(h|\theta) \\
& \quad \times P(A(h), X(h), Y(h), S(h)|h, \theta) \qquad (2.9)
\end{aligned}
$$

Here, we only care object related patches, while background terms $A(h_c)$, $X(h_c)$, $Y(h_c)$, $S(h_c)$ are dropped. We are doing this for 3 reasons:

- We think the penalty of lacking essential part has already modeled by $p(h|\theta)$, so it is redundant to have it in the spatial related term again.
- By dropping $A(h_c)$, $X(h_c)$, $Y(h_c)$, $S(h_c)$ at the first place, we avoid creating a background model to be used as the denominator to cancel it, like in [FPZ07], which requires a separate background set and it not quite *unsupervised.*
- As indicated in [FE73], modeling background and noise is difficult to be precise and comprehensive, and without it the detection problem can be regarded as finding the best linear embedding, which can be easily expressed as an optimization problem with clearly defined cost function.

As a result, we are only focusing on Term 2.9 in the following discussion. For convenience, we drop the subset indices (h) notation in $A(h)$, $X(h)$, $Y(h)$, $S(h)$ and representing them simply by A, X, Y, S in the remaining of this section for conciseness.

Because X and Y depend on S, we have,

$$
\begin{aligned}
P(h, X, Y, S) &= P(X, Y, S|h) P(h) \\
&= P(X, Y|S, h) P(S|h) P(h) \qquad (2.10)
\end{aligned}
$$

Obviously, X, Y are independent, we get,

$$P(X, Y|S, h) = P(X|S, h)P(Y|S, h) \tag{2.11}$$

Now, let us turn to $P(h)$, while calculating probability of h without considering X, Y and S, we don't care about the detail patch assignment until looking into the geometric information, leaving only the absence/presence information. Thus, d is a sufficient statistics for h here, as seen in the following equation,

$$P(h|\theta) = P(d|\theta) \tag{2.12}$$

Here, we have some difficulties to advance more. Firstly, because of the translation invariant nature of object on a image, given a (Δ_X, Δ_Y), we have,

$$P(X, Y, S) = P(X + \Delta_X, Y + \Delta_Y, S) \tag{2.13}$$

Similarly, for the scale parameter, given a scaling factor α, we have

$$P(X, Y, S) = P(X, Y, \alpha S) \tag{2.14}$$

From Equation 2.13 and Equation 2.14, we can see that $P(\theta)$ is multi-modal, and is technically difficult to learn from various images in which objects are at different locations and of various scales. To address this problem, Fergus [FPZ07] demonstrated an approach that using one *landmark part* as a reference parts, the scales of all remaining parts can be replaced with the ratios with scale of reference node, and the locations can be represented as the relative locations to the reference node normalized by its scale. For instance, the relative X values are defined by,

$$X'^T = \begin{pmatrix} \frac{X_2 - X_1}{S_1} & \frac{X_3 - X_1}{S_1} & \dots & \frac{X_M - X_1}{S_1} \end{pmatrix} \tag{2.15}$$

and now the shape is modeled as a joint Gaussian distribution

$$X' \sim \mathcal{N}(\mu', \Sigma') \tag{2.16}$$

$$\mu_i' = \mathbb{E}[X_i] - \mathbb{E}[X_1] \quad i > 1 \tag{2.17}$$

$$\text{Cov}[X_i', X_j'] = \mathbb{E}[(X_i' - \mathbb{E}[X_i'])(X_j' - \mathbb{E}[X_j'])] \tag{2.18}$$

Although it made the model theoretically scale-invariant and translation-invariant. There are still a few questions remaining unanswered. Firstly, the performance are highly coupled with the detection of the reference node (or landmark node); moreover, having K parts results in $O(K^2)$ parameters to learn, which prevents the use of numerous object visual templates.

Instead of centralized shape and scale representation regarding to a single landmark part, we developed a *distributed* approaches. To begin with, let us write the PDF in another form,

$$P(\mathbf{X}) = \frac{1}{Z} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{2.19}$$

Define $f(x)$ such that,

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{2.20}$$

In [FPZ03], given the scales of all parts, $\mathbf{s}$, $f(\mathbf{x})$ is represented by the best approximation in function family $g(\mathbf{x})$,

$$g(\mathbf{x}) = \sum_{i,j \neq 1} c_{ij} \left(\frac{x_i - x_1}{s_1} - \mu_i'\right)\left(\frac{x_j - x_1}{s_1} - \mu_j'\right) \tag{2.21}$$

From which it is clear to see the value is highly volatile to the landmark node (x_1, y_1, s_1) errors.

2.2 Learning the SUV Model

For learning, we consider the following scenario: Given a large and completely unlabeled image corpus featuring multiple instances of an *unknown* but a specific set of *structured objects* (i.e., each object has several parts that have consistent and stable spatial relationships), how can one automatically discover and build SUV models for the underlying but unknown object classes. Thus, in our setup the learning paradigm should not assume the knowledge of any known categories, and instead it aims to discover object categories in an unsupervised manner. That is, the models are to be learned not only in a non-discriminative manner, but also perceptually, by finding structures, both visual and spatial, that go together and define object categories. The only requirement for such an approach to succeed is that the given image corpus should have enough number of exemplars of the unknown object category to be learned and that such exemplars should cover the range of different configurations that are required to learn the variations within the object category.

2.2.1 Learning Vocabulary and Appearance Features of Viewlets

In the first step we randomly sample all images (utilizing a scale pyramid) in the learning set using a fixed-size rectangular patch, then convert all patches into image feature vectors, and then extract a visual vocabulary out of them using a clustering algorithm. While the specific implementation details can be found in Sections 4.2 and 5.2, it suffices to mention here that each visual word is a distribution in the feature space and represents a *potential viewlet* in the object models to be extracted from the learning set. For example, in Chapter 4 we use a dictionary of 1086 visual words, but not all of them correspond to viewlets that are part of the object class “Human”; only about one hundred of the potential viewlets are part of the actual object model.

2.2.2 Learning SRN (Spatial Relation Network)

In Section 2.1, we have simplified our model into the form of a sparse network, which can be determined by an edge set E, and the related parameters $\{C^{(x)}, C^{(y)}, C^{(s)}, \mu^{(s)}, \mu^{(x)}, \mu^{(y)}\}$. As a result, the learning process becomes much easier to handle.

2.2.2.1 A Maximum Likelihood (ML) framework

In this section we illustrate the ML formalism in the context of the X-axis potential function. First note that the partition function $Z^{(x)}$ in Equation 2.1 equals the normalization term used in Gaussian distributions and is proportional to $|\Lambda|^{-1/2}$, where $|\Lambda|$ is the determinant of the Precision Matrix Λ. By defining a random variable $Z_{ij} = \frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}}$, we write Equation 2.1 in log-likelihood form.

$$\log P(X) = \text{const} + \log |\Lambda| - \sum_{i \neq j} c_{ij} \operatorname{Var}(Z_{ij}), \tag{2.22}$$

where $\operatorname{Var}(Z_{ij})$ is the empirically observed variance of the random variable Z_{ij}. To maximize $\log P(X)$ while setting as many c_{ij}'s to zero as possible, we reverse the sign to get a minimization problem and add an $L - 1$ regularization term to obtain:

$$G(X) = -\log |\Lambda| + \sum_{i \neq j} c_{ij} (\operatorname{Var}(Z_{ij}) + \lambda), \tag{2.23}$$

where $\lambda > 0$ is the regularization parameter. The ML estimates of c_{ij}'s can be obtained by minimizing $G(X)$, subject to the constraint $c_{ij} \geq 0$. Note that this a convex optimization problem [BV04], and while the optimum values can be solved for numerically for any given data set, our objective here is to explore the properties that the optimal c_{ij}'s must satisfy so that we can find approximate solutions that are intuitive and easy to compute.

In particular, the optimal c^*_{ij}'s must satisfy the KKT conditions which for the constrained optimization problem (i.e. $c_{ij} \geq 0$) [BV04] are the following:

$$\frac{\partial G(X)}{\partial c_{ij}} = 0 \qquad \text{if } c^*_{ij} > 0 \tag{2.24}$$

$$\frac{\partial G(X)}{\partial c_{ij}} > 0 \qquad \text{if } c^*_{ij} = 0 \tag{2.25}$$

Hence, for all $c^*_{ij} > 0$ they must satisfy the following set of equations:

$$\frac{\partial G(X)}{\partial c_{ij}} = -\frac{\partial \log |\Lambda|}{\partial c_{ij}} + (\operatorname{Var}(Z_{ij}) + \lambda) = 0 \ . \tag{2.26}$$

Or equivalently,

$$\frac{\partial \log |\Lambda|}{\partial c_{ij}} = \operatorname{Var}(Z_{ij}) + \lambda \ . \tag{2.27}$$

Next we use the following property of the derivative of a determinant:

$$\frac{\partial |A|}{\partial \alpha} = |A| \operatorname{Tr}\left(A^{-1} \frac{\partial A}{\partial \alpha} \right) \ . \tag{2.28}$$

Note that the inverse of the Precision Matrix is the Covariance matrix, i.e. $\Lambda^{-1} = \Sigma$, and $\frac{\partial \Lambda}{\partial c_{ij}} = \frac{1}{(s_i^{(x)} + s_j^{(x)})^2}(e_i - e_j)(e_i - e_j)^T$, where e_i is the indicator column vector, where all entries equal 0 except the i^{th} entry, which equals 1. Substituting these identities we can further simplify the left-hand side of Equation 2.27 as follows:

$$\frac{\partial \log |\Lambda|}{\partial c_{ij}} = \frac{1}{|\Lambda|} \frac{\partial |\Lambda|}{\partial c_{ij}} \tag{2.29}$$

$$= \frac{1}{|\Lambda|} \times |\Lambda| \operatorname{Tr}\left(\Sigma \frac{\partial \Lambda}{\partial c_{ij}} \right) \tag{2.30}$$

$$= \frac{\Sigma_{ii} + \Sigma_{jj} - 2\Sigma_{ij}}{(s_i^{(x)} + s_j^{(x)})^2} . \tag{2.31}$$

Substituting this in Equation 2.27, we get the conditions for optimality:

$$\frac{\Sigma_{ii} + \Sigma_{jj} - 2\Sigma_{ij}}{(s_i^{(x)} + s_j^{(x)})^2} = \operatorname{Var}(Z_{ij}) + \lambda \ . \tag{2.32}$$

We next use the above optimality equation to derive a bound on the optimal c_{ij}^*'s, and for that purpose we use the determinant version of the Schur Complement: Let c be a column vector and r a row vector of appropriate dimensions, then

$$|(X + cr)| = |X|(1 + rX^{-1}c) \ . \tag{2.33}$$

Next we observe the following, (i) $\Lambda_{-c_{ij}^*} = \Lambda - c_{ij}^* \frac{(e_i - e_j)(e_i - e_j)^T}{(s_i^{(x)} + s_j^{(x)})^2}$ is another M-matrix where c_{ij}^* has been set to 0; (ii) Thus $|\Lambda_{-c_{ij}^*}| \geq 0$, and we already know that $|\Lambda| > 0$. Hence, substituting $X = \Lambda$, $c = -c_{ij}^* \frac{(e_i - e_j)}{(s_i^{(x)} + s_j^{(x)})}$, and $r = \frac{(e_i - e_j)^T}{(s_i^{(x)} + s_j^{(x)})}$ in Equation 2.33, we get

$$0 \leq |\Lambda_{-c_{ij}^*}| = |\Lambda - c_{ij}^*(e_i - e_j)(e_i - e_j)^T \frac{1}{(s_i^{(x)} + s_j^{(x)})^2}| \tag{2.34}$$

$$= |\Lambda| \left(1 - c_{ij}^* \frac{(e_i - e_j)^T}{(s_i^{(x)} + s_j^{(x)})} \Sigma \frac{(e_i - e_j)}{(s_i^{(x)} + s_j^{(x)})} \right) \tag{2.35}$$

$$= |\Lambda| \left(1 - c_{ij}^* \frac{\Sigma_{ii} + \Sigma_{jj} - 2\Sigma_{ij}}{(s_i^{(x)} + s_j^{(x)})^2} \right) . \tag{2.36}$$

Now using the fact that $|\Lambda| > 0$ and using the optimality condition in Equation 2.32, we get $(1 - c^*_{ij}(\mathrm{Var}(Z_{ij}) + \lambda)) \geq 0$ or equivalently:

$$c^*_{ij} \leq \frac{1}{\mathrm{Var}(Z_{ij}) + \lambda} \ . \tag{2.37}$$

Thus, the above bound on the optimal stiffness parameter connecting the locations of two viewlets, c^*_{ij}, decreases monotonically with increases in both the observed variance, $\mathrm{Var}(Z_{ij})$, and the sparsity parameter, λ. This makes intuitive sense, as two different viewlets that correspond to parts that are not directly linked by a stiff joint or some such structure in the physical object (hence, there are intermediate parts connecting them, and the location of one could be predicted accurately, given the locations of these intermediate parts), will tend to have a higher variance in their relative locations. Thus, a higher normalized variance in the relative locations of pairs of viewlets is a good measure of their statistical conditional independence or correspondingly, a lower stiffness in the spring connecting the underlying parts.

The above observation inspires us to use the bound in Equation 2.37 to determine sparsity in our GMRF model. By definition, we have $c^*_{ij} > 0$ and $\mathrm{Var}(Z_{ij}) > 0$, and we get

$$\frac{1}{c^*_{ij}} - \lambda > \mathrm{Var}(Z_{ij}) \ . \tag{2.38}$$

Thus, if we say that all those edges for which the optimal c_{ij} is guaranteed to be less than say a value c, will be removed from the network, then it implies from the above equation that all edges with empirical $\mathrm{Var}(Z_{ij}) < \frac{1}{c} - \lambda$ should be disconnected. Thus, we have derived a simple threshold rule on the pairwise variances, and by lowering the threshold (that is, by increasing the sparsity parameter λ) we get increasingly sparse GMRF models. Similarly, we can get upper bound for $c^{(y)}_{ij}$, and $c^{(s)}_{ij}$.

2.2.2.2 Implementation of the ML framework

The above ML framework can be implemented as follows: we first go back to the original image corpus (e.g., in the celebrity case, the approximately 9000 strong image learning set used in Section 4.6) and detect in each image the visual words that appear in it. That is, given an image, we first perform a dense scan (using the scaling pyramid so that we capture viewlets that have inherently larger scale), with a fixed-size sliding window, and assign a visual word to each resulting patch using a k-Nearest-Neighbor (kNN) algorithm (See Section 5.2 for a particular implementation). Note that the kNN algorithm is non-discriminative in the sense that no negative image patches have been used to train it, and it uses only the exemplar patches that were cropped from the learning set to determine the set of visual words. Then, for each pair of patches (p_i, p_j), in the image I, and the corresponding pair of assigned visual words (v_i, v_j), we count it as a co-occurrence of the visual word pair v_i and v_j on image I. As noted below the location and scale parameters of the viewlets are also noted.

This step is repeated for every image in the learning set, resulting in a co-occurrence count O_{ij} for every pair of visual words $(v_i,\ v_j)$. We account for potential detection errors and rare statistically insignificant co-occurrences by setting a threshold t, and only pairs with $O_{ij} > t$ are considered for the next step, where we utilize stable spatial properties.

If the pair of visual words describes patterns on the same real world object, the spatial relation between them should be more stable (by stability we mean low variance after normalization of their relative locations) and consistent through all co-occurrences (e.g., a "head" and an "arm" visual word pair) than between pairs that are related at an indirect level or not related at all. Consequently, when a sparsity directed threshold is applied on the variance, the surviving pairs are always better than those that got rejected. We next describe a method to compute spatial relationships between visual words.

Co-occurrences can be represented by pairs of image crops, each of which has its own determined position and size in the original image. Since all image patches in our dataset are cropped from an original image, given a bitmap image, B, and a patch of it, P, P is a sub-matrix of B, and can be determined by 4 parameters: the origin (we use top left corner)

(x, y) of the crop, the width $s^{(x)}$, and the height $s^{(y)}$, all in pixels.

$$Z_{ij}^{(s)} = \frac{S_j}{S_i} = \frac{s_j^{(x)}}{s_i^{(x)}} = \frac{s_j^{(y)}}{s_i^{(y)}} \tag{2.39}$$

$$Z_{ij}^{(x)} = \frac{(x_j - x_i)}{(s_i^{(x)} + s_j^{(x)})} \tag{2.40}$$

$$Z_{ij}^{(y)} = \frac{(y_j - y_i)}{(s_i^{(y)} + s_j^{(y)})} \tag{2.41}$$

Now, we are ready to estimate the statistical stability of any pair of visual word by computing variances of $Z_{ij}^{(x)}$, $Z_{ij}^{(y)}$, $Z_{ij}^{(s)}$ through all co-occurrences. In accord with Equation 2.5, $\log s$ is used to be compute variance. Here is the definition of the sum of variances V(volatility),

$$V = \mathrm{Var}(Z_{ij}^{x}) + \mathrm{Var}(Z_{ij}^{(y)}) + \mathrm{Var}(\log Z_{ij}^{(s)}) \tag{2.42}$$

To retain conditional dependence edges in the underlying GMRF model we are estimating, we check the sum of variances of the preceding three variables for all pairs of visual words, and only those with a sum less than our threshold are kept as edges.

Now, we can set up a threshold of V to keep pairs which have stable spatial relations. What's more, thresholds could be adjusted according to the required sparsity in graph. Section 5.2 details the choices of thresholds for different datasets. We refer to the network, comprising nodes that are not isolated and stable edges that survive the threshold criteria, as the *Spatial Relationship Network* or the SRN.

Note that we started off with all the visual words in the dictionary as the candidate nodes for the SRN, and then we let the ML estimation process to determine a sparse network that corresponds to the GMRF that best models the structure and appearance of the underlying object(s) that are in the learning image set. This is done in a completely unsupervised manner. From the perspective of noise reduction and model compression, the construction of the SRN enables us to achieve the following objectives:

- *Distilling Only the Objects From the Background Scenes:* By processing the learning image set, the final SRN will retain only those viewlets that correspond to an underlying object model. This is where the advantages of observing a large set of images, where the same object is captured under various different conditions and configurations, become most apparent: The visual words corresponding to the non-object categories will not have the consistent spatial structure as the visual words for the object that is consistently present in the majority of the images in the image corpus. Thus, our unsupervised methodology is able to create a network, in which only visual words or viewlets belonging to the same kind of objects will be connected. And each relatively densely connected community in the SRN will be about a certain kind of object. In Section 5.2 we usually feed images containing a single target object in majority of them while training a model, which results in one single giant connected component in the final network that models the unknown object in the corpus.

- *Eliminating Structurally Noisy and Non-Discriminative Visual Words*: In our unsupervised methodology, we do a K-means partitioning of the feature space to determine the visual words. However, due to both the limitations of the feature set we use and the inherent characteristics of the K-means or any other clustering algorithm, many of the visual words will not be visually uniform. Thus, during the kNN classification process, these visual words will be found in many locations in the images, and will not retain a consistent spatial relationship with other visual words, as required by our model. Similarly, some of the visual words might be visually uniform, but may not have enough discriminative features so that during kNN classification a diverse set of visual patches will be mapped to these visual words. Again this will imply that during the SRN construction, such nodes will lose most its edges. These are some of the robust features of our overall methodology.

For example, for the human body modeling application as described in Chapter 4 by applying community finding algorithms on the network learned from celebrity images, we get graph communities which consist of all human body part nodes. When we check visual words in one such community, we find that while the constituent visual words are far apart in the feature space, all of the visual words in the community relate to different representations of the multiple human body parts, i.e. *these visual words truly constitute viewlets as defined in our model.* The construction of SRN thus enabled us to break the links between celebrity body part nodes from all other scene related visual word nodes(like carpet, advertisement, etc.).

However, although we separated foreground/object nodes from that of the background ones, they are still mixed together as a large community, with head nodes scattered throughout. For example, two head nodes (i.e., two viewlets representing the same part, the head) don't necessarily have a better chance to be connected. In fact, sometimes it might be even impossible, if these two head nodes are about different kinds of heads, such as long hair versus short hair, because let alone the existence of stable spatial relationships, they even won't have a sufficient number of co-occurrences to qualify for a potential edge to connect them.

2.2.3 Extracting the Parts Using SEN (Spatial Exclusion Network)

Recall that our object model is an augmented MRF model, where in addition to the conditional dependence modeling, we also have the aspect of mutual exclusivity: Viewlets V_i and V_j could represent the same physical part of an object but in different configurations, resulting in very different visual appearance. In such a situation, V_i and V_j will never co-occur in an image and therefore will never have an edge in the MRF. Such exclusivity relationships are explicitly added to our generative model so that when sampling an image we do not pick viewlets that are mutually exclusive. Now we address estimation of the underlying parts (hence, the mutual exclusivity relationships) from the learning set and provide a complete learning of the entire model that we introduced in . In particular, we want to construct a Spatial Exclusion Network (SEN) from the SRN, where viewlet nodes corresponding to the same part are clustered into the same network community. Each such community thus represents a part of the object that can take on different configurations (for example, a bent vs a straight arm) or may have very different views (for example the hood of a car can have very different appearances based on the viewpoint of the observer and the make of the car).

In order to do derive the SEN, we must first define what a "configurable part" is. A dual way of looking at what constitutes a configurable part would be as follows: Two or more viewlets that are replaceable in making up the whole object, or equivalently, two or more viewlets that are *mutually exclusive*, **and** have *identical geometrical relationships* with other pieces (representing other parts). Thus, if two head nodes are visually different (See Figure 2.2) then the chances that they co-occur sufficiently many times and they have a stable spatial relationship are very small, and there is no edge between these two nodes in the SRN. However, these head nodes will have almost identical geometrical relationships with other viewlets such as those corresponding to arms or legs. Thus, as shown in Figure 2.2, *we would observe a wedge* with respect to a third node (C), when two head nodes are of different types (like A and D in figure). Because of how we automatically extract the viewlets, there is, however, another scenario to consider so that we can group all viewlets that correspond to the same part. This scenario arises when the viewlet nodes are only slightly shifted versions of each other. In such a situation, because of the features we are using, they constitute different visual words, but they share a very stable edge (that is have a low variance in their relative locations) in the SRN between them; for example, nodes A and B in the figure. These viewlets again will have almost the same geometrical relationships with viewlets corresponding to other parts of the object, and hence would *form a triangle* with a third node (C).

Using the above principles, we now compute the Spatial Exclusive Network (SEN) from the SRN as follows: For every pair of nodes A, B, we first determine if they share at least two other nodes C and E, such that the spatial relationships $A \leftrightarrow C$, $B \leftrightarrow C$ are almost identical (i.e., the difference is within a small threshold) and the spatial relationships $A \leftrightarrow E$, $B \leftrightarrow E$ are also almost identical. We add a third node to suppress noise. If a pair of nodes A, B satisfies the above condition, we add an edge between the two to construct the SEN.

For example, for the human body modeling application as described in Chapter 4, after applying community finding algorithms on the SEN (as shown in Figure 4.2(c)), we observed that the network is further dissembled into small components, and each of them has a well-

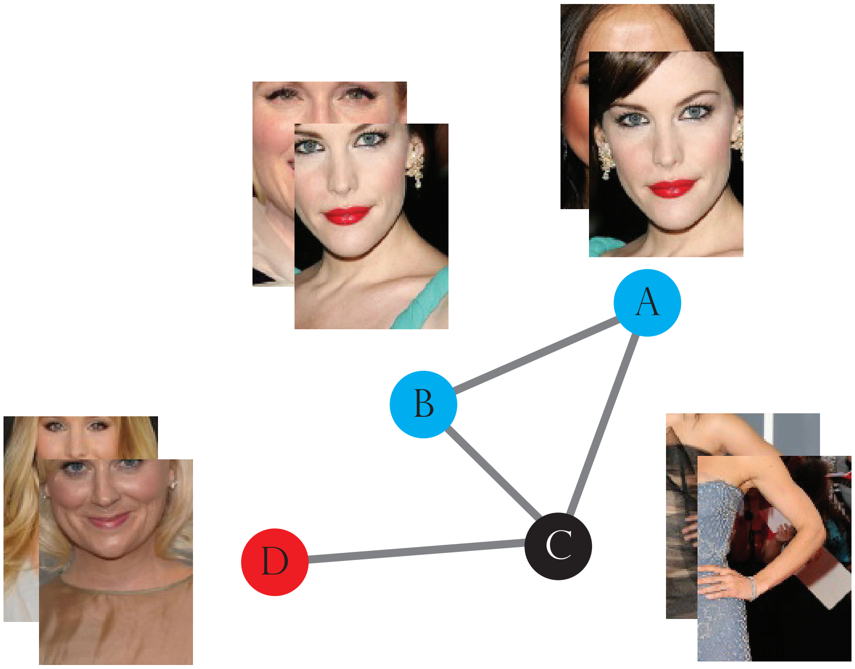


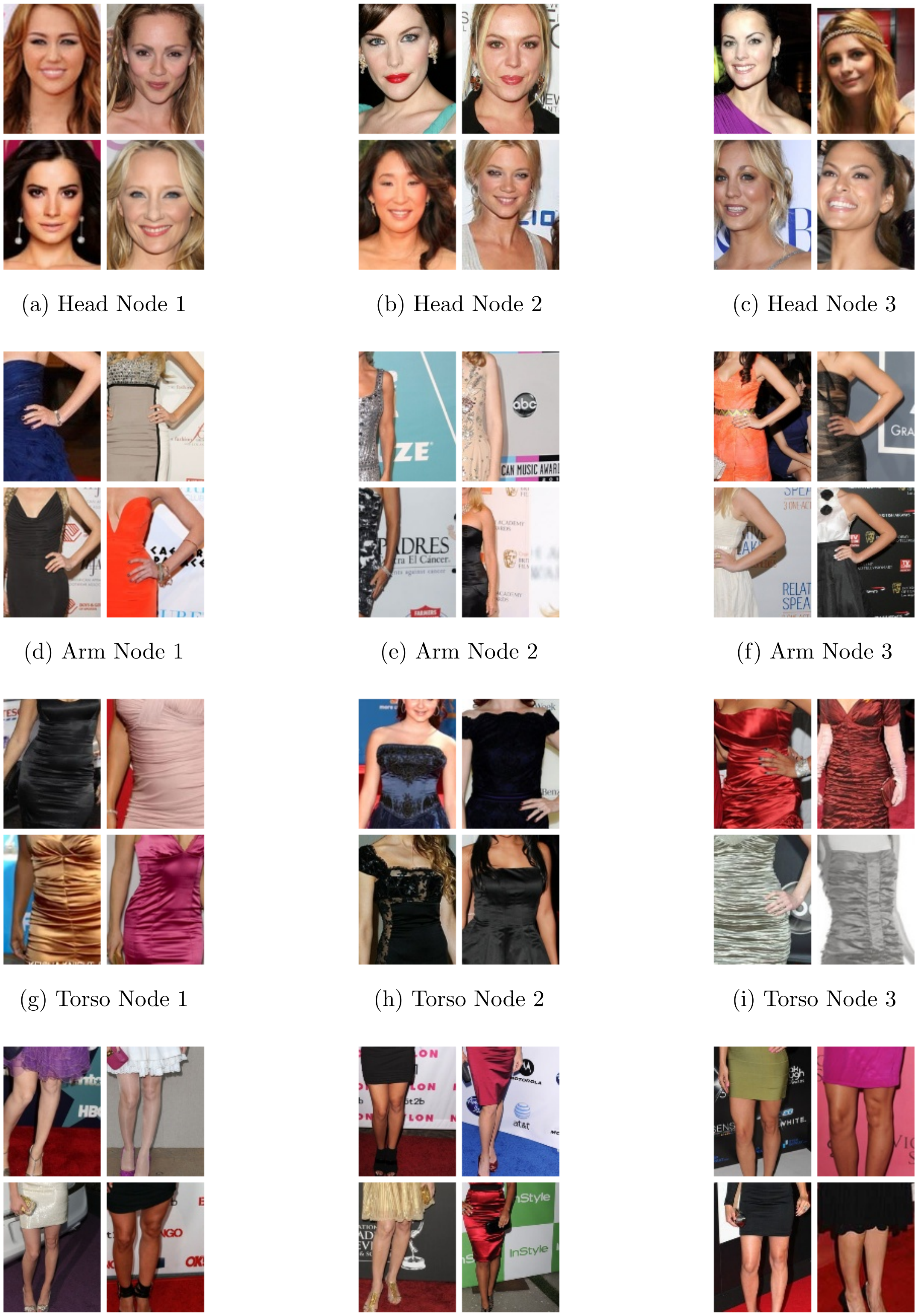
Figure 2.2: Wedge(B,D,C) and triangle(A,B,C) structures in the SRN.

defined semantic meaning. Each community corresponds to a distinct human body part as also labeled in Figure 4.2(c).

Figure 2.4 further validates our part-finding results. As explained in Section 2.2.4 we can use the SRN to compute global positions of every visual word and as one can see, the viewlets or visual words corresponding to each part (as determined from the SEN) occupy distinct regions in the virtual 2-D space, almost defining a human body contour.

2.2.4 Global Information Reconstruction

In preceding sections, we have demonstrated that we can model in an unsupervised manner how the entire object system is assembled out of many pieces and parts. In the SRN, edges represent stable spatial relations with relative size and position information of endpoint visual words. Though all spatial information is stored locally in a distributed manner, we develop an algorithm to assemble these pieces and parts to restore the desired global properties of



(a) Head Node 1

(b) Head Node 2

(c) Head Node 3

(d) Arm Node 1

(e) Arm Node 2

(f) Arm Node 3

(g) Torso Node 1

(h) Torso Node 2

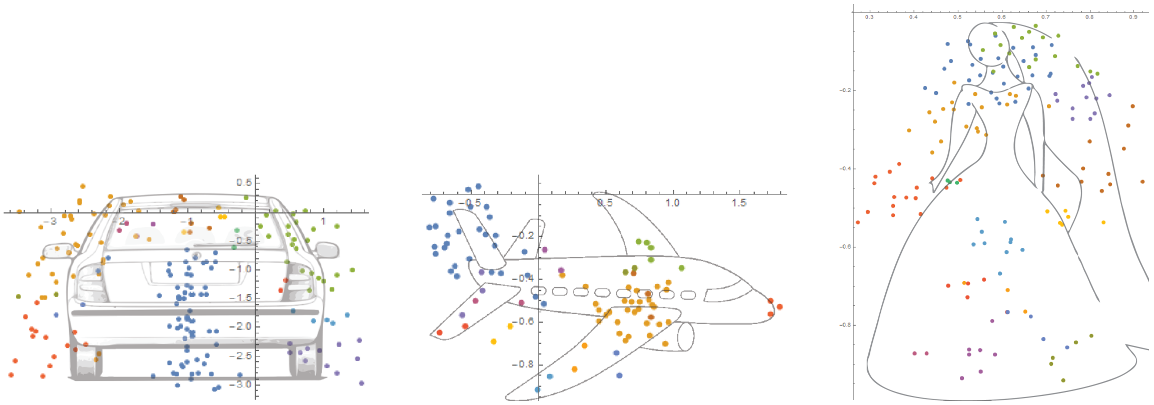
(i) Torso Node 3

(j) Leg Node 1

(k) Leg Node 2

(l) Leg Node 3

Figure 2.3: Different views of three body parts discovered by the Spatial Exclusion Network computed in Chapter 4.



(a) Caltech-101 Rear Car (b) Caltech-101 Airplane (c) Celebrity

Figure 2.4: Examples in global structure reconstruction.

the underlying object system.

For a densely connected component in this network, to reconstruct the structure of the entire system, we decide to take advantage of the pairwise local information to calculate a set of global locations for all the viewlets in the SRN network. This task is in some ways similar to that of some previous works, like Multi-Dimensional Scaling (MDS) [Kru64]. We derive an iterative approach to calculate the position and size of each viewlet using the relative positions and size of its neighbors by solving optimizing problems as shown in Equation 2.43 and Equation 2.44. The approach is summarized in Section 4.5, Algorithm 1.

$$\min \sum_{(i,j)\in E} c_{ij}^{(x)}\left(\frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}} - \delta_{i,j}^{(x)}\right)^2 + c_{ij}^{(y)}\left(\frac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}} - \delta_{i,j}^{(y)}\right)^2 \tag{2.43}$$

$$\min \sum_{(i,j)\in E} c_{ij}^{(s)}(\log s_i - \log s_j - \delta_{i,j}^{(s)})^2 \tag{2.44}$$

From Algorithm 1, we get global position assignments for all nodes in the largest component of the Spatial Relation Network. We plot all nodes using the global coordinates we got for 3 different objects, as shown in Figure 2.4.

From Figure 2.4, we notice that the community partition of the Spatial Exclusive Network (SEN) is highly correlated with the global spatial value we derived from the SRN.

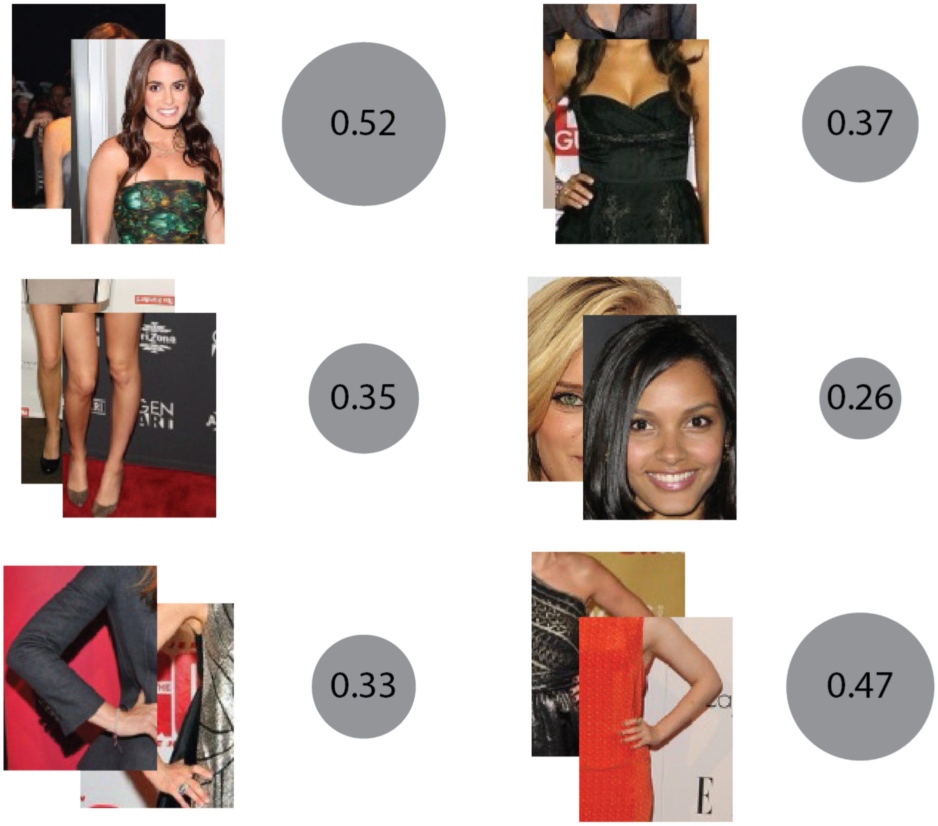


Figure 2.5: Global scale values for some body viewlet.

Furthermore, the global positions of the nodes in these communities of the SEN mirrored the object in real world. Our ability to reverse engineer human body structure demonstrates that we are successfully identifying the semantic meaning of images by leveraging network communities instead of purely hard knowledge encoding (manual tagging, specific features, etc.)

Using the same algorithm, we can extract a global scale value for each of the meaningful visual words. Some examples of these values are listed in Figure 2.5 with corresponding images. We can see that while the nodes from some communities of the SEN are all sharing similar scale values, which are in good accordance with our understanding of the related parts.

2.3 Detection

Once we learned model θ as we described in Section 2.2, we can use the model for object detection. In a traditional detection approach, for a given image, the saliency discovery algorithm will first be applied, and a list of Points of Interest are detected. Parts must be placed on these candidate positions to form different configurations. Among these configurations, the best fitting is picked alongside the corresponding h. That is, given appearance vector set A, and the corresponding spatial information X, Y, S, which consists of feature of all sampled patches in image, we are looking for a h, which picks subset of A, $A(h)$, as the best location and configuration of the desired object.

$$h_{opt} = \arg\max_{h} P(h|A, X, Y, S, \theta) \tag{2.45}$$

Applying Bayesian rule and the conditional relation defined in Equation 2.10 and Equation 2.12, we get,

$$\begin{aligned} P(h|A, X, Y, S, \theta) &\propto P(A, X, Y, S|\theta, h)P(h|\theta) \\ &= P(A|h, \theta) \\ &\times P(X, Y|S, h, \theta)P(S|h, \theta) \\ &\times P(d|\theta) \end{aligned} \tag{2.46}$$

To find h_{opt}, a straightforward approach is to search exhaustively in h space. However, for each v_i, the corresponding $h_i \in \{0, ..., N\}$ can have $N + 1$ possible values, combined with n_f foreground visual words, the search space is $O(N^{n_f})$, which is a hopeless task. Several pruning approaches have been proposed, like using A* search([FPZ07]), simplify the structure (chain like dependency, tree-like dependency [FH05, FGM10], k-fan structure [CFH05]) and use dynamic programming, the later ones results in a time complexity $O(N^2 n_f)$. However, it's still not good enough for large N and n_f, which lets these approaches to use a restricted set of Point of Interest's (Though in unsupervised learning, dense sampling is generally believed to have better performance than discovered interesting points [TLB10]), and using only a few unimodal parts (less than 20 parts were used in [FPZ07]).

We develop an agglomerate approach, which shares the characteristics of A* like pruning and optimal substructure in dynamic programming. To begin with, given $d = sign(h)$ as defined in Section 2.1, we have $P(h|\theta) = P(d|\theta)$. Let's divide Equation 2.46 into three parts, the part absence/presence setting $P(d|\theta)$, the global structure $P(X, Y|S, h, \theta)P(S|h, \theta)$, and the local appearance $P(A|h, \theta)$.

Firstly, while a full set of part patches are required to calculate the global term, the local term can be factored as,

$$\log P(A|h, \theta) = \sum_{i=1}^{n_f} \log P_i(A(h_i)|\theta) \tag{2.47}$$

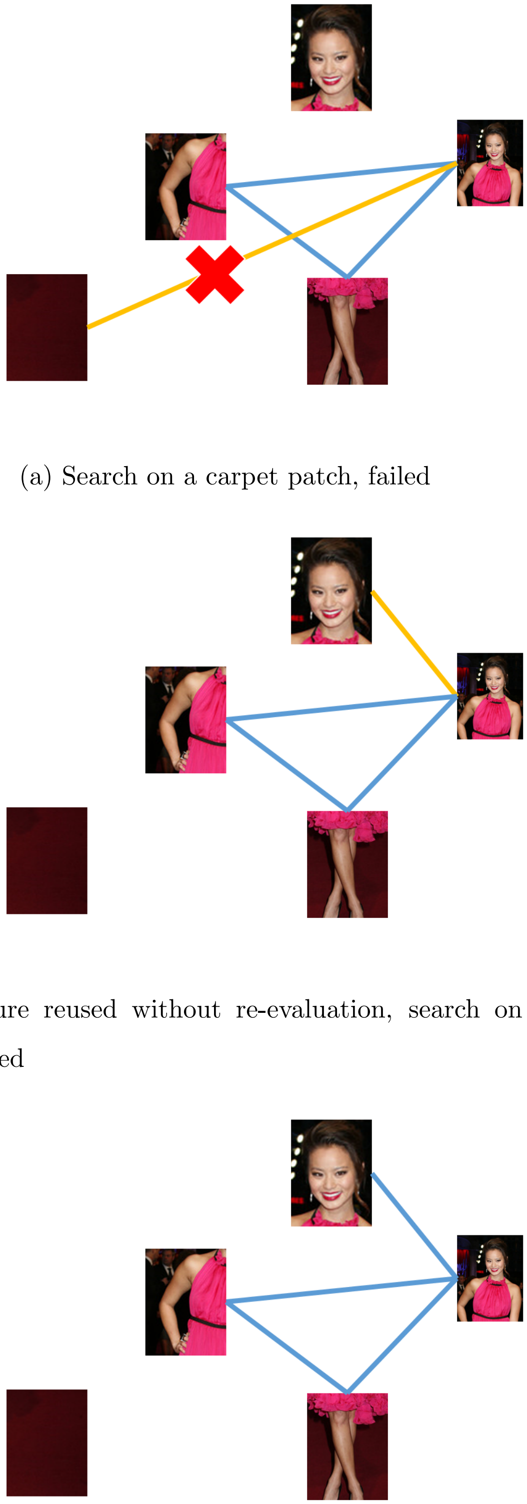
Then, for patch $P_{j,j\in\{1,...,N\}}$, when $h_k = j$, $P_k(A(j)|\theta)$ is the probability to be assigned as part k, and can be evaluated before full search space enumeration and stored in a table. After obtaining this table, we can prune heavily by dropping sub-spaces which have one or more extremely unlikely matches in them. For instance, In our celebrity model, i_{head} is a head template, and as defined, $h_{i_{head}}$ is the index of the corresponding patch. Now, if we also have $P_{j_{leg}}$, which is a patch of leg template. As a result, with any meaningful feature and distance metric $\{A, P_k\}$, $P_{i_{head}}(A(j_{leg})|\theta)$ will be so small that all $(N+1)^{n_f-1}$ h's have $h_{i_{head}} = j_{leg}$ in it can be dropped without further inspection. Moreover, if feature vectors of patches have been discretized using a K word vocabulary, letting $V_k \subset \{1, 2, ..., N\}$ denotes the index set of all patches in current image which are classified as visual word v_k, we can further enforce that any h_i can only be picked from patches detected as one single word v_{f_i}. Now we have reduced the search space from $(N+1)^{n_f}$ to $\prod_{i=1}^{n_f} |V_{f_i}|$. Considering that $\sum_{i=1}^{K} |V_i| = N$, the pruned space is much more compacted.

Secondly, to further boost the performance, we need to look into the global term. Taking natural logarithm on both sides, we have,

$$
\begin{aligned}
&\log P(X, Y|S, h, \theta) + \log P(S|h, \theta) \\
&= \sum_{e_{i,j} \in E} \left(c_{ij}^{(x)} \left(\frac{x_i - x_j}{s_i^{(x)} + s_j^{(x)}} - \mu_{ij}^{(x)} \right)^2 \right. \\
&+ c_{ij}^{(y)} \left(\frac{y_i - y_j}{s_i^{(y)} + s_j^{(y)}} - \mu_{ij}^{(y)} \right)^2 \\
&\left. + c_{ij}^{(s)} (\log \frac{s_i}{s_j} - \mu_{ij}^{(s)})^2 \right) \qquad (2.48)
\end{aligned}
$$

From Equation 2.48, we can see that though individual nodes are not independent, edges are highly separated and optimal substructure can be reused without repeatedly evaluating. However, in our probabilistic setting, we don't have a clearly defined score to evaluate a *portion* of the solution. To address this problem, we must regard the detection process as finding the best embedding of desired object in the image, and both Equation 2.48 and Equation 2.47 are terms of the cost function. Therefore, using similar technique as that of Kruskal's Algorithm, each foreground patch can be regarded as the simplest (maybe also most unlikely) solutions which only have one node in the group. We check all node pairs and add edges between those the spatial relations of which are close enough to the modes of the model ($\mu_{ij}^{(x),(y),(s)}$) according to the model threshold. By doing so, we are merging small graphs into big ones, and eventually there will be one component left, which is the solution (when multiple objects are presented in the image, there can be more than one components left).

Thirdly, we need also to define $P(d|\theta)$, which is not accounted for in our learning stage. A straightforward solution will be using a Beta-Binomial setting, so the probability only depends on number of detected foreground words. However, this approach ignores the divergence in the patch group. For instance, in human detection, a group with one head patch and one torso patch should be regarded as a better one than another group with 2 head patches. As a result, defining part as the communities discovered in SEN (see Section 2.2.3), we reduce viewlet presence/absence indicator d into a part one d_p. Assuming all parts are equal, we have sufficient statistics, the L1-norm of d_p, $||d_p||_1$, Therefore, $P(d_p|\theta) = P(||d_p||_1|\theta)$. When



(a) Search on a carpet patch, failed

(b) Substructure reused without re-evaluation, search on a head patch, succeeded

(c) group merged

Figure 2.6: Optimal-substructure in patch grouping.

a part is present, we are seeking for at least 2 patches of related viewlets, more would be better, but will not help as much as the first 2, to model this behavior, a sigmoid function is used. In summary, we have,

$$||d_p||_1 \sim \sum_{i=1}^{n_p} \frac{1}{1+e^{-(n_{p_i}-1)}} \tag{2.49}$$

$$\begin{aligned} P(d|\theta) &= P(d|d_p)P(d_p|\theta) \\ &= P(d|d_p)P(||d_p||_1|\theta) \\ &\sim P(d|d_p)\frac{1}{1+e^{-(||d_p||_1-1)}} \end{aligned} \tag{2.50}$$

Furthermore, we can bring in some approximation here by just counting the number of part with more than 2 patches detected, which binarized n_{p_i} values. Therefore, $P(d|d_p) \sim uniform$, and we have,

$$P'(d|\theta) \sim \frac{1}{1+e^{-(||d_p||_1-1)}} \tag{2.51}$$

2.4 Summary

In this chapter we have presented some fundamentals of the SUV model and then outlined the analytical approaches that can be used to solve the corresponding learning and detection algorithms. In Chapters 4 and 5 we provide further details and variations of the learning and detection algorithms that we have developed herein.

CHAPTER 3

Related Work

3.1 Introduction

Object discovery and localization lies at the heart of computer vision, and has attracted a large amount of research interest. However, it remains a difficult task due to several challenges. The first is the limitation of computer image representation, which is inconsistent to spatial bias and color, illumination condition, white balance fluctuations; the second is the variable object appearances of objects from the same category, and the “deformable” nature of many objects.

To address the first problem, several attempts have been made to deliver representative and robust visual representations, or, “features”, which represent image regions as feature vectors. Feature vectors usually preserve contour, texture and patterns, color and intensity information. Detection algorithms rely on feature vectors and spatial relations between them to make decisions. As a result, these features are of crucial importance and must be representative and separable in the feature space. We will discuss a variety of different feature descriptors in **Section 3.2**.

With extracted features, numerous strategies can be applied to detect objects:

- Simple features directly encode the entire object as a **rigid template** of feature vectors. General machine learning techniques like Principal Component Analysis [SL90], template matching, k-Nearest-Neighbors, or discriminative approaches like Support Vector Machines (SVM), Logistic Regression, etc., and even generative approaches like Bayesian approaches with the Expectation-Maximization can be then used as holistic classifiers to detect object as a whole, like pedestrian detection [DT05, DTS06].

- However, in more complex cases, objects cannot be simply represented by a general template. To account for such objects, **attributed-based** and **part-based approaches** have been densely studied. In such approaches, an object class is usually represented in a factored way. With factored representations, the detection can be done by recognizing similar image patterns/features that commonly appear in instances of this class and the spatial relationship between them.
  - Early work of factored models were based on a bag-of-visual-word model, upon which various existing approaches from text-based data mining were applied, from simple K-Means clustering, spectral approaches like kernel PCA and Laplacian Eigenmap, to more complex ones.
  - Among these methods, probabilistic models have been intensively studied. Sivic proposed a method based on Probabilistic Latent Semantic Analysis. Later, Latent Dirichlet Allocation (LDA) [RES06, BNJ12], which had been widely used in the text topic modeling field, was also adapted. A further extension, Spatial Latent Dirichlet Allocation, was proposed by X. Wang to account for spatial information [WG07].
  - Supervised approaches rely heavily on well labeled image datasets. However, there are usually far more categories to be labeled than that of text. Moreover, because of the intrinsic nature of bitmap image representations, the intra-cluster variances are considerably large. As a result, it is difficult to annotate and maintain a comprehensive dataset. Therefore, current applications of supervised approaches on Internet-scale datasets are far from ideal. As a result, numerous studies have been done on demonstrating unsupervised approaches on object discovery [FPZ04, GD06, KHK11]. Tuytelaars and Lampert et al. have a comprehensive survey on these approaches [TLB10].
  - To discover relations between parts, different approaches have been developed,

Marius Leordeanu and Martial Hebert proposed a spectral approach for finding persistent feature pairs [LH05]. A structural similarity discovery approach was proposed by Blondel [BGH04]. To represent the structure of the entire system and settings of each part, there are considerably more parameters, which makes the learning process even more challenging. In order to address this problem, some approaches with simplified structure were proposed. Tree models, which simplify the fully joint relation between parts into a tree [FH05], or tree-like Bayesian networks [CFH05], were densely studied. Felzenszwalb and McAllester also proposed a weakly-discriminative deformable part model approach [FMR08,FGM10,GFM11]. Lubomir and Jitendra proposed a 2-layer supervised approach to detect and localize people using "Poselet" [BMB09]. What's more, in order to have the shape and structure of the parts-based system properly described throughout the variance of training image resolutions, a scale-invariant unsupervised approach was proposed by Fergus et al. [FPZ03,FPZ07] and Fei-fei et al. [FFP03,FP05,FFP06], in which one particular part is used as a reference point to have the locations and scales of other parts normalized and represented.

Among these studies, there are three most prominent ones:

The weakly unsupervised Bayesian generative approach by Fergus and Fei-fei et al. and its variants [FPZ03,FFP03,FPZ04,FZP04,FPZ05,FP05,FFP06,FFP10], which has a predetermined number of parts and use a Bayesian graphical model to represent the spatial relationship between parts, and use the Expectation-Maximization algorithm to learn the probability distribution function parameters;

The Deformable Part Model by Felzenszwalb et al. and its variants [FH05, CFH05, FGM10, FGM09, GFM11, CML14], use Histogram of Oriented Gradient(HOG) feature as the descriptor, and a constellation-structured cost function to control spatial relationship;

poselet proposed by Bourdev et al. [BMB09, BMB10], and its variants [GHG14, ZFD12, ZDG14,ZPT15], which provides a sophisticated way to find discriminative image crops which are also representative. These approaches will be discussed in greater detail in **Section 3.3**.

Although these methods can represent deformable parts along with pairwise spatial information, the spatial information representation and system structure modeling are far from optimal. Constellation-based approaches or star-models are highly dependent on a root or landmark feature, which makes the model's performance prone to landmark/root node detection error. Meanwhile, they fail to capture the real world relations among parts. Moreover, the EM and variational approaches used in these models are known to be slow when the number of parameters to be learned is high, which makes it even more computationally impractical to work with complex high-dimensional visual descriptors or Internet-scale datasets.

Network models are highly flexible and representative, which makes it an ideal tool for exploring real world systems and their mechanisms. Moreover, it is probably the most well studied model and there are numerous delicate algorithms to reveal structure from the topology of networks [TWH03,NG04,New06,BGH04,For10]. Community finding algorithms are widely used in our framework as an unsupervised clustering approach. Therefore, we also introduce Complex Network and community finding studies in **Section 3.4**.

3.2 Features

There are mainly two general classes of features: sparse features and dense features.

- **Sparse Features** In sparse approaches, instead of converting the entire image in to a set of feature vectors, another stage of saliency region discovery (usually blob detectors [AKB08], like the Laplacian of Gaussians (LoG),the Difference of Gaussians (DoG)) is placed in front of feature extraction. With this extra stage, only these regions, or so-called points of interest defined by the saliency information filter, are encoded into descriptors (e.g. Scale-Invariant Feature Transform (SIFT) [Low99], Speeded Up Robust Features (SURF) [BTV06], shape context [BMP01,BMP02,BM00]). The main advantages of these approaches include: (i) the resulting data is much smaller size because except the highly discriminative regions, majority portion of the image is dropped, and (ii) the descriptors are usually resistant to location shift, noise, and other kinds of variation.

- **Dense Features** In dense approaches, images are swept with a fixed step and all regions are converted into feature vectors. Such approaches usually require more storage space. However, they are not necessarily slower than their sparse counterparts, since even in the sparse approaches, it is still necessary to scan entire image for saliency discovery. Although preserve an excessive amount of information is usually regarded as a drawback, it is of critical importance in unsupervised learning [TLB10]. Among features that are usually used in dense sampling scenarios, the most significant is probably Histogram of Oriented Gradients (HOG) [DT05]. This is based on occurrences of quantized gradient orientation in patches of an image and uses local contrast normalization for consistent representation.

3.3 Part-Based Approaches

3.3.1 An Unsupervised Approach: Fergus' Scale-Invariant Bayesian Generative Model

The Bayesian generative approach and graphical models are generally accepted as a powerful machine learning technique in representing complex systems, and are widely adapted in text data mining, like Probabilistic Latent Semantic Analysis (PLSA) [Hof99], Latent Dirichlet Analysis (LDA) [BNJ12], etc. With proper feature representation, the same approaches can be applied to image data mining. The first part-based Bayesian model was proposed by Burl and Weber. In their approach, the EM algorithm was used to learn the graphical model for image classification and category discovery [BWP98, WWP00]. Fergus et al. proposed a vastly improved model, which explicitly accounts for shape, scale and appearance [FPZ03, FPZ07].

In Fergus's approach, an object has a fixed and predetermined number of parts. The appearances of parts are modeled as Gaussian distributions in the feature vector the space with to-be-learned means and variance values. Among these parts, one of them is marked as the root part. Upon determining root part, shape is represented by all other parts relative to the root part. Similarly, the scale of each part is also normalized by comparing it to the absolute scale of the root part. Therefore, the entire generative pipeline is formed by chaining these random variables by conditional probability. As a result, for an image with an appearance matrix A, define a latent variable position vector X, a scale vector S and a part presence indicator h, given the object(foreground) parameters, the likelihood can be written as Equation 3.1.

$$\begin{aligned}
&P(A, X, S|\theta_{fg}) \\
&\quad = \sum_{h \in H} P(A, X, S, h|\theta_{fg}) \\
&\quad = \sum_{h \in H} P(A|X, S, h, \theta) \\
&\quad \times P(X|S, h, \theta_{fg}) P(S|h, \theta_{fg}) P(h|\theta_{fg})
\end{aligned} \tag{3.1}$$

Similarly, if there is no object in the image, given the background parameter set θ_{bg}, the likelihood can be written as Equation 3.2.

$$\begin{aligned}
&P(A, X, S|\theta_{bg}) \\
&\quad = \sum_{h_0 \in H} P(A, X, S, h_0|\theta_{bg}) \\
&\quad = \sum_{h_0 \in H} P(A|X, S, h_0, \theta) \\
&\quad \times P(X|S, h_0, \theta_{bg}) P(S|h_0, \theta_{bg}) P(h_0|\theta_{bg})
\end{aligned} \tag{3.2}$$

By calculating the posterior ratio R, as shown in Equation 3.3 and comparing it to a threshold

T, detection decisions can be made.

$$\begin{aligned} R &= \frac{P(\text{Object}|A,X,S)}{P(\text{No object}|A,X,S)} \\ &= \frac{P(A,X,S|\text{Object})P(\text{Object})}{P(A,X,S|\text{No object})P(\text{No object})} \\ &\approx \frac{P(A,X,S|\theta_{fg})P(\text{Object})}{P(A,X,S|\theta_{bg})P(\text{No object})} \end{aligned} \tag{3.3}$$

As seen in Equation 3.1, to make the model scale and translation invariant, the scale latent variable S is the parent node of the position latent variable X. Each of these likelihood functions has a fixed form which mostly consists of Gaussian, Poisson and geometric PDF's. Object model is learned using the Expectation-Maximization (EM) algorithm, detecting parts and their configurations, and estimating the parameters of the above densities from detected configurations alternatively until convergence.

Fergus's approach addresses many problems of object recognition, including learning without supervision, modeling appearance explicitly, scale-invariance, etc. However, it is still unsatisfactory due to the following unsolved problems: one is that the part number is predetermined and fixed, which is restricted when the part number is unknown; Moreover, for n parts, there are $O(n^2)$ parameters to learn, in order to control the running time and keep the algorithm computationally feasible, the part number must remain small; Another problem is that the EM algorithm is inherently slow and inefficient, especially when there are multiple large covariance matrices to learn; Last, a constellation model is highly dependent on the accurate detection of the root/reference part, and is prune to root part detection error or bias.

3.3.2 A Weakly Supervised Approach: the Deformable Part Model

Unlike the Unsupervised Bayesian Model introduced in the preceding section, the Deformable Part Model (DPM) is a discriminative model introduced in 2007 by Felzenszwalb et al. which sought to address the performance gap between part-based models and rigid templates

[FMR08,FGM10]. The model is trained in a supervised way. However, the authors refer to it as "semi-supervised" because although a bounding box is required for every positive object, parts will be inferred by the model later in the learning process and are not required to be labeled before training. The system outperforms the by-then best results in the PASCAL VOC 2007 challenge in ten out of twenty object categories. A new approach for training was also proposed, in which a generalized SVM with latent variables representing part positions was used to learn parts from weakly-labeled data.

The Deformable Part Model shares essentially the same idea as the pictorial structure model in earlier studies [FH05, BWP98]. It includes a global object template and a few part templates organized as a star-structured model, which captures spatial relationships between parts and the root. The appearance is represented by Histogram of Oriented Gradient (HOG) features from [DT05, DTS06] with some parameter change to reduce the descriptor complexity. Principal Component Analysis (PCA) was then applied to further reduce dimension. By applying PCA at cell-level instead of patch-level. The image cell structure is preserved, which eliminates the necessity to re-evaluate the descriptors for each small shift.

In the detection stage, a dense set of possible positions and scales will be searched. Matching here is more difficult because multiple parts must be placed while the global criteria is evaluated. The tree/star model allows a dynamic programming search strategy which is more efficient. In each candidate configuration, a score will be calculated based on the appearance similarity estimated by a filter of each template, as shown in Equation 3.4,

$$\sum_{x',y'} F[x',y'] \cdot G[x+x', y+y'] \tag{3.4}$$

and the geometric relation deformation represented by cost function, as in Equation 3.5,

$$f(p_0, \cdots, p_n) = \sum_{i=0}^{n} F_i' \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b \tag{3.5}$$

where,

$$(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i) \tag{3.6}$$

$$\phi_d(dx, dy) = (dx, dy, dx^2, dy^2) \tag{3.7}$$

To simplify the model, the interactions between overlapping parts are not modeled. The authors claimed that, despite potential benefits, it is unlikely to be a problem in a discriminative approach.

In the learning stage, DPM proposed a latent SVM to train a part-based discriminative model: While the bounding boxes of objects are given, the bounding box of each part is not. Instead, the position of part and model parameters were estimated together with the Expectation-Maximization (EM) algorithm to minimize the error rate on the training images.

The Deformable Part Model, while reducing the performance gap with rigid template models, is still far from perfect. Unlike Fergus's approaches [FPZ03, FPZ07], DPM needs supervised training, which require each object in the training images to be labeled with a bounding box. However, it might not always be feasible in Internet-scale learning, considering how easy it is today to download millions of images in one day. Moreover, similar to the preceding section, the EM algorithm becomes slow when parameter number is large. Especially in latent SVM training, each possible configuration must be evaluated, which further impacts the performance. Last, the model inherently lacks flexibility and representative power. Lack of interactions between parts makes the root-leave connections the only geometrical constraints. The predetermined number of parts and even the area of them, makes the use case even more limited.

3.3.3 A Strongly Supervised Approach: Poselet

"Poselet", a term which suggests that it describes a part of one's pose under a given viewpoint, was proposed as a new notion of part by Bourdev et al. [BMB09]. Each poselet has a corresponding rectangular patch of a given person on a given image, and also corresponds to a point in the "configuration space" of human pose. A good poselet is claimed to have two most desired properties:

- It is separable in the feature space, which means the members of the same part are tightly clustered in the feature space and relatively distant from members of other classes, which leads to easy detection and classification for traditional computer vision algorithms.

- It has clear semantic meaning, which, in the authors' words, "should be easy to localize the 3D configuration of the person conditioned on the detection of a poselet."

With such kind of "poselet", a two-layer classification/regression model for detecting and localizing object at part level, was proposed [BMB09, BMB10]. Poselet model has a similar structure to deformable part model [FM06, FMR08]. The first layer consists of individual poselet classifiers which detect part patterns in the image. The second layer applies a max-margin optimization approach, and makes decisions based on the detection output of the first layer and the spatial interactions between them. One main difference from [FM06] is that, in Felzenszwalb's approach, parts were discovered automatically as in the preceding section, while in poselet they are trained in an outright supervised manner. To facilitate the extra annotation work, a new keypoint-based annotating approach was proposed to find good poselets efficiently. The overall procedure is described as follows.

Poselet Generating Firstly, the configuration space is defined as a 2D space with a set of keypoints (like left eye, right shoulder, nose, left elbow, etc.). With this model, each concrete example in images can have its keypoints translated into the configuration space. The distance of 2 examples s, r can be defined as,

$$\sum_i w_s(i)||x_s(i) - x_r(i)||^2(1 + h_{s,r}(i)) \tag{3.8}$$

where $x_s(i) = [x, y, z]$ are the normalized coordinates, $w_s(i)$ are Gaussian function centered at the patch center, and The $h_{s,r}(i)$ is visibility mismatch penalty.

To generate a large population of poselet candidates, a rectangle window is used to sweep over a training image with human annotations. At each position, least-square is used to find close matches from every other annotated human images in the training set regarding to a threshold λ. This process generated 120 000 poselets. After removing low occurrence ones and similar ones, there were 2000 left.

Poselet Selection For each of these 2000 poselets, patches were converted into Histogram of Oriented Gradient (HOG) features proposed by Dalal and Triggs [DT05] and linear SVMs were trained with non-people negative images. The training process has multiple iterations with extra hard false positive added continuously. After the training process, the performances of all SVMs were evaluated, and only the top 300 were kept while others were abandoned.

Detection Firstly, Generalized Hough Transform framework was used to fit transformations from the poselet to the object. Then, for a query image, each poselet detector was used to sweep over the query image. All hits were collected and grouped into clusters. A vote based on the learned transformation with corresponding weight was casted from each cluster, in which weights were learned via Max Margin Hough Transform [MM09]. The optimization function is as follows,

$$\min_{w,b,\xi} \frac{1}{2} w^t w + C \sum_{i=1}^{T} \xi_i \tag{3.9}$$

$$s.t. y_i(w^t A_i + b) \geq 1 - \xi_i \tag{3.10}$$

$$w \geq 0 \tag{3.11}$$

$$\xi_i \geq 0 \tag{3.12}$$

$$\forall i = 1, 2, \ldots, N \tag{3.13}$$

With the weights, the probability of detecting the object O, at position x is,

$$P(O|x) \propto \sum_i w_i a_i(x) \tag{3.14}$$

Poselet is generally regarded as a state-of-the-art approach, and have been widely adapted [ZFD12, GHG14, ZDG14, ZPT15, GGM14]. However, there remains a need to design a set of keypoints, and annotate all images, which makes it an impractical option for Internet-scale dataset. Moreover, though it was claimed that the model was designed to capture a *part* of a pose, however, the fact might be different from the poselet list of the most widely used poselet model (demonstrated on `https://www.eecs.berkeley.edu/Research/Projects/`

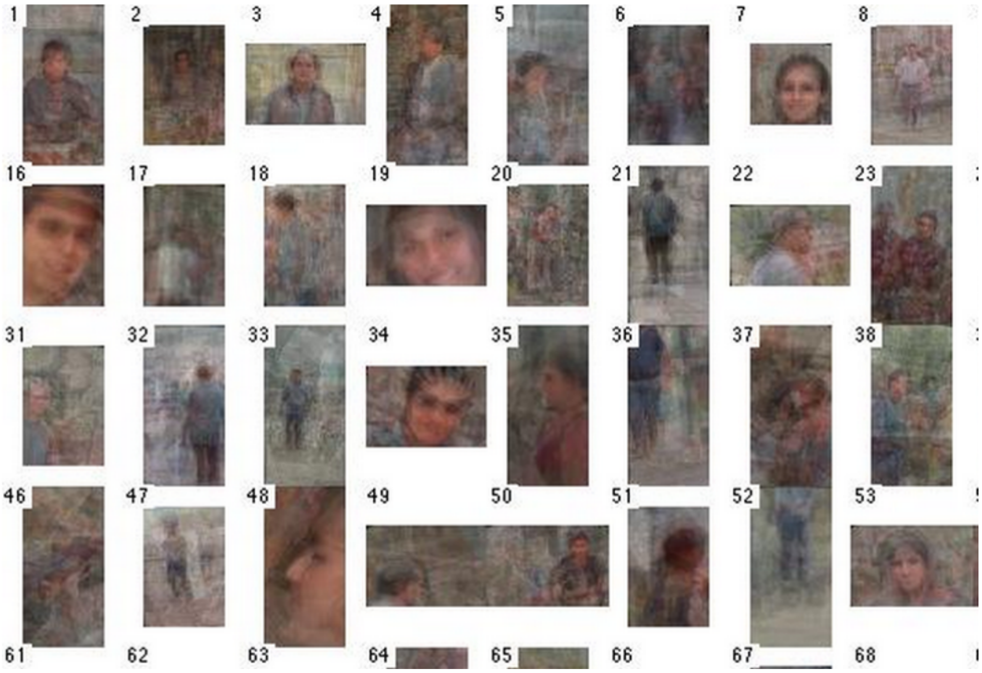


Figure 3.1: Some poselet used by the human body model in [BMB09].

`CS/vision/shape/poselets/poselets_person.html`), as shown in Figure 3.1 We can see that very few templates are really about a part, while most of them are just outlines of pedestrian.

3.4 Network Models and Community Finding

3.4.1 Emergence of Complex Network

Since Euler's work on the *Seven Bridges of K´onigsberg* in 1736, for around 300 years researchers have extensively studied graphical models, or "networks", and the important role they play as the representation of complicated systems in virtually all fields (including social science, engineering, biology, etc.) over the world. The social relationship between people can be regarded as a network, cities with the roads and freeways between them can be regarded as a network, electrical power grids can also be regarded as a network, etc. While more real systems have been studied using graphical models and more mathematical properties of networks have been discovered, traditionally graph theory became an extremely useful tools to study network until recent times.

The development of computer and the Internet provides gigantic amount of computing resources and data processing power, enables researchers to address some even more complicated systems, like proteins and the interaction between them, computer clients and the Internet among them, the World Wide Web, which consists of billions of websites, network of acquaintances of a considerable population of people and the communication between them(i.e. Facebook), etc. This kind of system, while sharing the same basic structure with their smaller brothers studied before, is significantly larger, and can have much more dramatical dynamics. These changes imposed huge challenges and were beyond the capability of existing classic graph theory approaches. To address this issue and characterize the topology and dynamics of real networks, a new field, Complex Network, emerges [AB02, BA99, DM00, New03, BLM06].

3.4.2 Properties of Complex Network

It's widely accepted that while at a significantly larger scale (number of nodes and edges), some properties which generated considerable interest on these old, classic networks are not meaningful anymore. In contrast, statistical properties become the new research focus. Numerous studies have suggested that there are at least three striking statistical features of complex networks. They are scale-free, small-world, and communities.

3.4.2.1 Scale-Free: Preferential Attachment and Power-Law Degree Distribution

A famous early attempt to model real systems is random graph by Erdós and Rényi [ER59]. In a random graph, edges are distributed in a highly homogeneous way, and the vertex degree distribution $P(k)$, should be a Poisson distribution. However, careful measures on numerous real systems suggested that the real distribution is significantly biased from a

Poisson one, in contrast, it's more like a power-law one with an exponent β between 2 and 3 [BA99,BAJ99,AHL00,SR04]. Therefore, a long tail exists in the distribution diagram, and large degree nodes have an important presence in the graph.

The mechanism behind this phenomenon has been intensively studied and many models have been proposed. Among them, a dynamic model, *Preferential Attachment*, proposed by Barabasi (BA) is the most significant one [WS98, BA99, ASB00, JNB01, SR04, SR05, KR08, KSR08]. In this model, there is a mechanism called preferential attachment: When the network grows and a new node is added, it is more preferentially attached to high-degree nodes. The "preference" can be modeled with different functions. Specifically, when the connecting probability is proportional to the node degree, it is called linear PA, and results in a power-law degree distribution. BA model provides a general way to represent a wide variety of real systems and their growing characteristics, and is widely adapted.

3.4.2.2 Small-World Effect

The first famous demonstration of small-world effect is probably the so-called "six degrees of separation" by Stanley Milgram. To probe the average path lengths in network of acquaintance, letters were passed from person to person with a designated destination. A large amount of trials showed that mails, if reached the target, usually took only a few of steps. To be more specific, there are usually around 6 people on the path. This experiment showed the fact that in a large scale network, pairs of vertices are connected by short paths through the network.

Recent literatures have defined the small-world effect in a more precise way [WS98]. Denote l as the average degree of a network in Equation 3.15.

$$l = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i>j} d_{ij} \tag{3.15}$$

Given mean degree, networks has the small-world effect if $l \sim O(log(n))$. The small-world effect has been widely observed in a variety of real systems [WS98], and is believed to have a crucial role in many features and processes that take place in networks, including synchronization, rumor and epidemic spreading, etc. Studies also suggested that the mechanism is likely to be associated with the high values of the clustering coefficient and the communities structures in the network [New00, New01].

3.4.2.3 Community Structure

The massive analysis of networks suggests that large systems usually share some dramatic structural features and a high level of organization [GN02, NG04, BLM06]. Most of them can be partitioned into clusters, or communities. Members of each community are densely connected, but the links between different communities are much sparser.

Community structure is an important property of many complex networks [PDF05, TWH03, CKL09, TLM10, For10, YSN11]. From some more concrete systems, like families in an acquaintance network, geographically close cities in a highway system, to more abstract and virtual ones, like customers sharing similar interest of products, articles describing related topics, proteins sharing identical functions. In these networks, some nodes solely belong to one densely connected communities, whereas some others may have a higher level of participation and act as bridges between communities. The differences in the roles in the community structure are of such a great importance that sometimes they become the most important identities of nodes (like political bias in a congressmen network, movie genre in a movie network, and the native language in a friendship network).

Studies of many network systems in the real world showed that community structure usually also have organized in a hierarchical manner [RB03]. Communities in a network include smaller communities, which in turn are composed by even smaller communities, etc.

3.4.3 Community Finding

Since the community structure is such an important feature of network systems, discovering the communities of a network is an effective way for understanding the architecture and dynamics of the network. Community detection algorithms are proposed to identify the modules and to find community organization of a graph/network only using the topology of the graph without any extra information, and to measure how good such community assignments are. Furthermore, in our case, *when the network was created using a set of data entries and a given similarity/distance measure*, discovering communities also allows us to do clustering of these nodes, according to their membership assignment in the discovered modules.

3.4.3.1 Traditional Approaches

Traditionally, there are two board classes of techniques to find communities: agglomerative methods and divisive methods. These two kinds of approaches try to address the same community finding problem from opposite directions. In agglomerative methods, which is also called hierarchical clustering, nodes are grouped according to their similarity hierarchically until there is only one cluster left. In contrast, the divisive approaches, which are also referred as partitional clustering, try to solve the community finding problem by addressing the classical graph theory problem, *min-cut*: To partition a graph into 2 sub-graphs of comparable sizes, with the aim of minimizing "bridges" between them. Though the studies of community finding algorithms has a long history dating back to around 80 years ago in social science [Ric27], and related problems were discussed in graph theory even earlier, most of them were originated from small graphs and are incapable to handle large-scale networks. Specifically, the graph partition problem is NP-complete, so efficient heuristic is required to keep this task manageable.

In the last few years, a series of novel approaches have been proposed [LLM10, For10]. A breakthrough was the betweenness-based approach by Girvan and Newman [GN02]. In Girvan and Newman's new divisive algorithm, centrality measures like edge betweenness (How many times an edge takes place in the shortest paths of other vertices pairs.) are used to identify these bridge edges, which lie between communities. After a series of bridge removal, the network will be divided apart and communities will reveal. After this milestone paper, community finding has attracted huge amount of focus and numerous new methods have been proposed. Moreover, because of the increasing need of finding communities in large scale networks, besides the detection quality, the time complexity becomes the most important factor.

3.4.3.2 Modularity Optimizing Approaches

Modularity, Q, as defined in Equation 3.16, was introduced by Newman [New04, New06]. It changes the community finding problem into a cost function optimization one, and has 2 major contributions: One is that it provides a by then best measure of community assignment quality, the other is the inherent flexibility of modularity. The modularity measure triggered the emergence of a variety of new algorithms, some of fast variants can handle Internet-scale datasets and make them computationally feasible with state-of-the-art quality.

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \tag{3.16}$$

In the initiated study, Newman proposed a greedy agglomerative approach [New04] to optimize the modularity Q. To begin with, each node is a community. Each step a new edge is added and communities are joined to form larger communities. Correspondingly, connecting node i and j will also increase modularity by

$$\Delta Q = A_{ij} + A_{ji} - 2k_i k_j = 2(A_{ij} - k_i k_j)$$

In each step, the edge (i, j) which will result in maximal modularity increase ΔQ is chosen. This algorithm allows clustering to be done at a much higher speed than the fastest algorithm, and first brings the analysis of much larger networks into reality.

This work was later improved by Clauset [CNM04] using more sophisticated data structures: Max-heaps were used to maintain the modularity variations for fast maximum value retrieval; and arrays are maintained to keep the sum of rows of the edge matrix e_{ij}. These data structures vastly boosted the performance while keeping exactly the same numerical results. This variant is able to handle networks with more than one million nodes, and is widely adapted in the analysis of Internet-scale systems.

However, Newman's approach tends to form badly unbalanced dendrograms. Consequently, the running time is usually close to the worse case and the community assignment is not the most favorable. To address this problem, multiple algorithms were proposed [DDA06, WT07, SC08], Blondel et al. proposed another approach which yields much better partitions [BGL08]. The beginning configuration is the same as that of Newman's: every single node belongs to its own community. However, the agglomerating process is quite different. Instead of looking for the global maximum ΔQ, all vertices are swept in one pass while each gets merged with the neighbor on the other end of the local maxima edge. This technique provides almost linear time complexity. Therefore, it is considerably faster than other approaches and able to handle networks with billions of nodes.

CHAPTER 4

Human Prototype Representation

4.1 Introduction

4.1.1 Motivation and Problem Statement

Creating statistical models and the task of unsupervised learning of visual concepts from Internet-scale data can be aptly described as a problem that an alien might face on a reconnaissance trip to the earth. An advanced alien probe that can tap into the radio-signal backbone of the Internet, will have access to a huge time-stamped database of human-created images and videos at the pixel level. They can only rely on fundamental and universal knowledge about statistical and mathematical patterns, and about space and geometry to try to retrieve objects and structures that have correlations, and have statistical measures that separate them from randomness. The alien can use various contextual groupings of the data it collects, for example grouping images based on time intervals, to make sure that the same objects will repeat in the image corpus and thereby increase the chances of making statistics count in its favor.

To make matters more concrete, let us consider the following problem: An alien visitor, who hacks the Internet to obtain a large set of images featuring humans (still unknown to the alien) in various positions and scenes, wants to automatically construct a statistical model of the object (i.e., human) and the various parts that it has. We find it useful to phrase the alien's task as a *jigsaw puzzle design problem*: Determining a set of representative "pieces" (i.e., distributions of visually similar image patches) for each body part (i.e., what we humans will call head/face, shoulders, arms, legs etc.), such that (i) each piece represents a specific visual configuration of the underlying part (e.g., a specific pose for the "arm" part), and (ii) spatial relationships among the parts are specified, dictating how they all fit to make a human body.

In general, to summarize all the objects that we use, the alien will have to solve the following largely-open problem: Given a large and completely unlabeled image corpus featuring multiple instances of an *unknown* but a specific set of *structured objects* (i.e., each object has several parts that have consistent and stable spatial relationships), how can one automatically discover and build composite abstract representations of the underlying unknown objects, their different parts (i.e., the groups of jigsaw "pieces"), and the relative spatial positions of the parts (i.e., how the "pieces" fit) within an object.

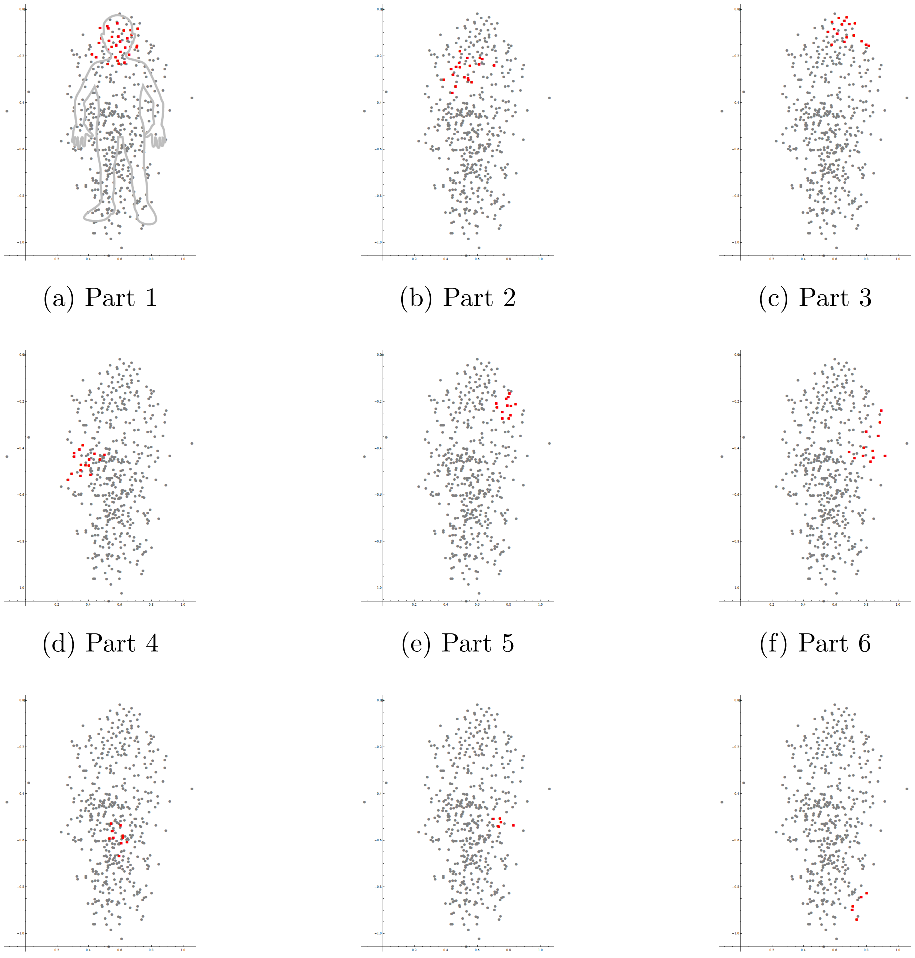
4.1.2 Our Approach and an Outline of the Chapter

We address the human prototype learning problem using the technique described in Chapter 2. Firstly, we create a high quality celebrity image set. We develop an approach to learn the graphical model in a fully unsupervised way from the celebrity image set, and the corresponding algorithm to do detection using the learned model as described in **Chapter 2**. The approach is summarized in Figure 4.2. The first step (described in Section 4.2) is to randomly crop the images into a large set of patches at multiple scales, and represent each cropped image patch using a general feature descriptor (e.g., what an alien would be expected to have), such as local Histogram of Gradients (HOG) [DT05] (instead of sophisticated object-dependent feature descriptors that are used in the Computer Vision community, where labeled objects need to be learned). Then we use K-Means clustering to separate these patches into visually similar groups. Each such group is really a candidate "piece" in our SUV model. Since the associated image patches, in terms of the underlying objects (See Figure 4.2), represents parts in specific views, we refer to each cluster also as a "viewlet". With this network, we can find structurally similar nodes [KFH08, BGH04], and because of our augmented edge with relative spatial information, the spatial similarity was also verified. This enables us to find a k-partite like sub-graph in our network, which establishes the connection between visual templates of the same part (like arm, torso, head, etc.) with dramatically appearance difference, which, in some early studies, was added manually or using text information from the Internet [KHE12]. We also show that the global spatial information can be restored completely from the recorded pairwise data.

From an object perspective, many of the "pieces"/viewlets are noisy, irrelevant, or cover multiple parts (Section 4.2.2). The next step (Section 4.3) is to construct the Spatial Relation Network, where each node is a piece/viewlet, and edges are based on co-occurrence and stable geometric relations. The SRN structure has multiple uses in the SUV framework, including (i) It automatically filters out irrelevant and noisy pieces. (ii) It allows computations of a global position and scale for each piece/viewlet. The SRN still does not group the pieces into parts. The wedges and triangles in it are further processed to construct the Spatial Exclusive Network (SEN) (Section 4.4).

We next show how (Section 4.6) SUV can successfully extract all the human body parts by applying it to a corpus of more than 9000 mostly celebrity images. As a further evaluation of our SUV model, we construct a *"head/face" part detector*. Instead of the network-based grouping strategy described in Section 2.3, we use a simplified anchor-based grouping strategy, and use network deduced transform rules to predicate the position of head/torso from detected parts. We show that when evaluated on a test set of almost 3000 images of humans (with mostly full body), our automated detector's performance (98% precision at 87.8% recall) compares very favorably with the performance of a popular OpenCV Viola-Jones face detector (76% precision at 92.9% recall), which has been carefully manually trained using large-scale labeled data. The intuitive reason that our automated detector performs so well is because it is using other body parts to help in the detection of the location of the face. We also train a torso classifier, and show that the resulting classifier outperforms the Deformable Part Model (DPM) in experiments, and have comparable results with the poselet approach (which was trained in a heavily supervised way).

Section 4.7 presents a minimally trained face detector created using our framework that performs better than well-trained state-of-the-art techniques.

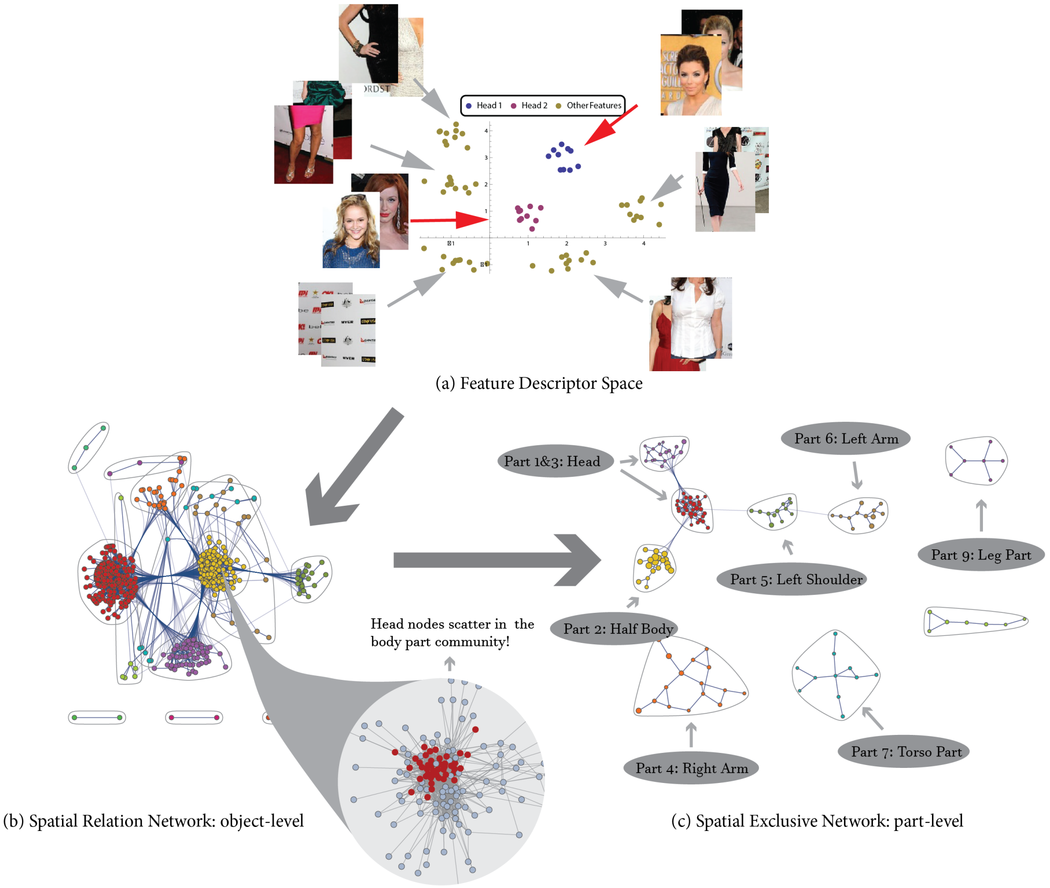


(a) Part 1 (b) Part 2 (c) Part 3

(d) Part 4 (e) Part 5 (f) Part 6

(g) Part 7 (h) Part 8 (i) Part 9

Figure 4.1: For each viewlet related to the yet-to-be-determined object, we automatically compute a global position as explained in Sec. 4.5. Note that each “piece” is a compact region of the feature descriptor vector space, as illustrated in Figure 4.2(a). Then in each sub-figure we highlight (in red) the groups of pieces that we compute as belonging to a distinct part. Note that, (i) The pieces from the same part occupy a distinct region, verifying the efficacy of SUV, and (ii) The different parts (head, shoulder, legs etc. see also Figure 4.2) almost outline a human body figure; a human contour figure is overlaid in the first figure as a reference and no knowledge of it was used during the computations.



(a) Feature Descriptor Space

(b) Spatial Relation Network: object-level

(c) Spatial Exclusive Network: part-level

Figure 4.2: The different steps in the SUV framework. It starts with breaking up the given corpus into a large number of patches or crops, and quantizing the feature space into candidate viewlets, also referred to as viewlets. We have shown example viewlets in terms of the human body parts, just to illustrate visually what they stand for; of course, at this stage the algorithm does not "know" what they are, until the parts are computed at the end of the procedure. From an object perspective, many of the "pieces" are noisy, irrelevant, or cover multiple parts. The next step is to construct the SRN based on co-occurrence and stable geometric relations. The SRN has multiple uses in the SUV framework, including (i) automatically filters out irrelevant and noisy pieces, and (ii) allows computations of a global position and scale of each piece/viewlet. Finally, by processing wedges and triangles in the SRN, we compute the SEN, where pieces representing the same parts form distinct connected components. Figure 4.10 provides a visual verification that the parts were computed accurately.

4.2 Step I: Clustering in the Feature Descriptor Space and Viewlets

4.2.1 Learning the vocabulary

As in Section 2.2.1, we start our SUV model by first randomly cropping images into patches of fixed size but at multiple scales. Then we use a general feature descriptor, such as the local Histogram of Gradients (HOG) [DT05] to represent each patch as a vector. These vectors are then clustered using the K-Means algorithm to obtain K clusters[1]

Each cluster comprises patches that are visually similar to each other, within the quantization and expressiveness properties of the feature descriptor chosen to represent the patches. In terms of the underlying objects (See Figure 4.2), each such cluster represents parts in specific configurations. For the human example, a cluster could represent heads, arms in different poses, half heads, half bodies etc. Thus, the clustering process breaks up the feature descriptor space into Voronoi-type separable regions, and we define each such region as a distinct *Viewlet*. *Each viewlet is now a potential "piece" in our SUV model.*

Moreover, we notice that the frequency of most viewlets is inversely proportional to its rank, which follows the general trends of the Zipf law, as seen in Figure 4.3. It might imply the intrinsic similarity between our viewlets and words in natural language corpora.

4.2.2 Challenges in Going from Viewlets to Actual Objects

Identifying objects from images is challenging because the same real world element can look dramatically different in different images, and even visually similar images, once converted into feature descriptors, can be distant from each other in feature space. These differences are caused by:

1. **Different 2D projections of the same object** — When a 3D object is projected to a 2D space, there are many different projections due to the reduction of degrees of

[1] See Section 4.6 for a detailed discussion of cropping procedure, the specific HOG image feature descriptor choice, and the choice of K for our dataset.

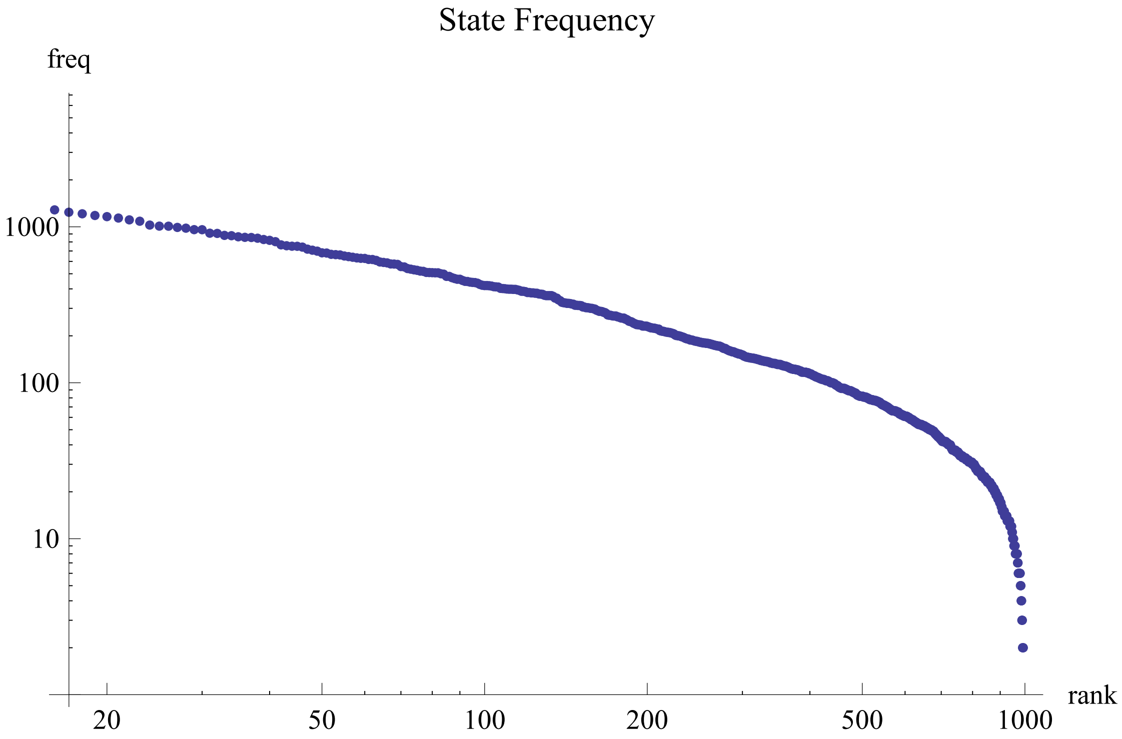


Figure 4.3: Viewlet frequency vs. rank table.

freedom. In addition, objects may have many internal degrees of freedom (e.g., flexible joints in humans).

2. **Differences in scale and translation** — While it may be easy for humans to identify the object in different image patches with slightly shifted positions and scaled sizes, sophisticated features are required to do this in computer vision.

As a result, viewlets computed in Step I using a typical feature descriptor have the following limitations from the perspective of object and part modeling:

1. *One-to-Many:* The same part (e.g., "head") could be represented by several viewlets that are far apart in the feature descriptor space. Thus, standard measures such as Euclidean norms cannot be used to find their semantic similarity.

2. *Transitional:* Due to our blind and random cropping, some viewlets are, what may be called as *transitional*, and may overlap different parts (for example, a half-body

viewlet).

3. *Non-Object or Background:* Viewlets will also capture perfectly valid visual patterns that appear as background in the image corpus, but are not related to any of the objects of interest.

4. *Many-to-One:* Because of the feature descriptor's resolution limitations, several viewlets will comprise multiple objects/parts in them.

We next outline steps how to use co-occurrence and geometric relationships amongst viewlets to eliminate *Non-Object* and *Many-to-one* viewlets and identify and organize the *One-to-Many* and *Transitional* viewlets into parts and objects.

4.3 Step II: Constructing and Analyzing the Spatial Relation Network (SRN)

In Chapter 2, we defined a Spatial Relation Network (SRN) as $G(V, E)$, where V is the set of nodes, and each node corresponds to one unique viewlet; thus $|V| = K$. Next two viewlet nodes S_i and S_j are connected by an edge e_{ij} based on the following construction, which has two parts to it. In the first step we go back to the original image corpus (e.g., the 9000-image learning set used in Section 4.6) and we detect in each image the viewlets that appear in it. That is, given an image, we first perform a dense scan at each pyramid level, with a fixed-size sliding window, and assign a viewlet to each resulting patch using a k-Nearest-Neighbor algorithm (See Section 4.6 for a particular implementation). Then, for each pair of patches (P_A, P_B), in the image I, and the corresponding pair of assigned viewlets (S_A, S_B), we count this as a co-occurrence of the viewlets S_A and S_B on image I.

This step is repeated for every image in the learning set, resulting in a co-occurrence count O_{ij} for every pair of viewlets (S_i, S_j). We account for potential detection errors and rare statistically insignificant co-occurrences by setting a threshold t, and only pairs with $O_{ij} > t$ are considered for the next step, where we utilize stable geometric properties.

4.3.1 Determining Geometrically Stable Edges

If the pair of viewlets are describing the same real world object, the geometric relation of them should be stable and consistent through all co-occurrences (e.g., a "head" and an "arm" viewlet pair). The same is not true for those pairs that are related at a scenario level. We next describe a method to compute geometric relationships between viewlets. (Discussed in greater detail in Section 2.2)

Co-occurrences can be represented by pairs of image crops, each of which has its own determined position and size in the original image. Let B denote a bitmap and P denote an image patch of B. Since all image patches in our dataset are cropped from an original image, P is a sub-matrix of B, and can be determined by 4 parameters: the origin (x, y) of the crop, the width w, and the height h.

$$P = B_{x\cdots(x+w-1),y\cdots(y+h-1)} \tag{4.1}$$

If we use a fixed aspect ratio, $\alpha = \arctan\frac{h}{w}.$, then only three parameters are required to define the relation of a pair of patches from the same image. Those 3 parameters correspond to the x, y, s we discussed deeply in Chapter 2, as shown in the following equations.

$$Z_{ij}^{(s)} = \frac{s_j}{s_i} = \frac{w_j}{w_i} = \frac{h_j}{h_i} \tag{4.2}$$

$$Z_{ij}^{(x)} = \frac{(x_j - x_i)}{(s_i + s_j)} \tag{4.3}$$

$$Z_{ij}^{(y)} = \frac{(y_j - y_i)}{(s_i + s_j)} \tag{4.4}$$

To construct edges for the SRN, we are going to a *volatility* score to check the aggregated variances of the preceding three variables for all pairs of visual words, and only those with a sum less than our threshold are kept as edges.

We normalized the weights of x, y in v as in Equation 4.5 and Equation 4.6, according to the ratio α such that the same intrinsic variances alongside x-axis and y-axis will result in the same value change in v. have,

$$\hat{Z}_{ij}^{(x)} = \frac{Z_{ij}^{(x)}}{\cos\alpha} = \frac{(x_j - x_i)}{(w_i + w_j)} \tag{4.5}$$

$$\hat{Z}_{ij}^{(y)} = \frac{Z_{ij}^{(y)}}{\sin\alpha} = \frac{(y_j - y_i)}{(h_i + h_j)} \tag{4.6}$$

Now, we are ready to estimate the statistical stability of any pair of visual word by computing variances of $\hat{Z}_{ij}^{x)}$, $\hat{Z}_{ij}^{(y)}$, $Z_{ij}^{(s)}$ through all co-occurrences. In accord with Equation 2.5, $\log s$ is used to be compute variance. Here is the final definition of the sum of variances V(olatility),

$$V = \mathrm{Var}(\hat{Z}_{ij}^{x}) + \mathrm{Var}(\hat{Z}_{ij}^{(y)}) + \mathrm{Var}(\log Z_{ij}^{(s)}) \tag{4.7}$$

Now, we are ready to estimate the statistical stability of any pair of viewlet by computing volatility score v from all co-occurrences.

4.3.2 Properties and Limitations of SRN

The construction of the SRN enabled us to:

- Filter all Many-to-One noisy viewlets (since they will not have stable edges with other viewlets and thus will appear as isolated nodes in the SRN) and only keep those which are all describing the same object.
- Create a network in which only viewlets of the same kind of objects will be connected. And each densely connected community will be about a certain kind object.

Figure 4.4 illustrates one community in the SRN for our dataset. For example, by applying community finding algorithm on this network, we got communities which consist of all human body part nodes. When we check viewlets in one such community, we find that while the constituent viewlet nodes are far apart in the HOG feature space, all of viewlets in this community relate to different representation of the multiple human body parts, as shown in Figure 4.5. The construction of SRN thus enabled us to break the links between celebrity body part state from all scene related states (like carpet, advertisement, etc.).

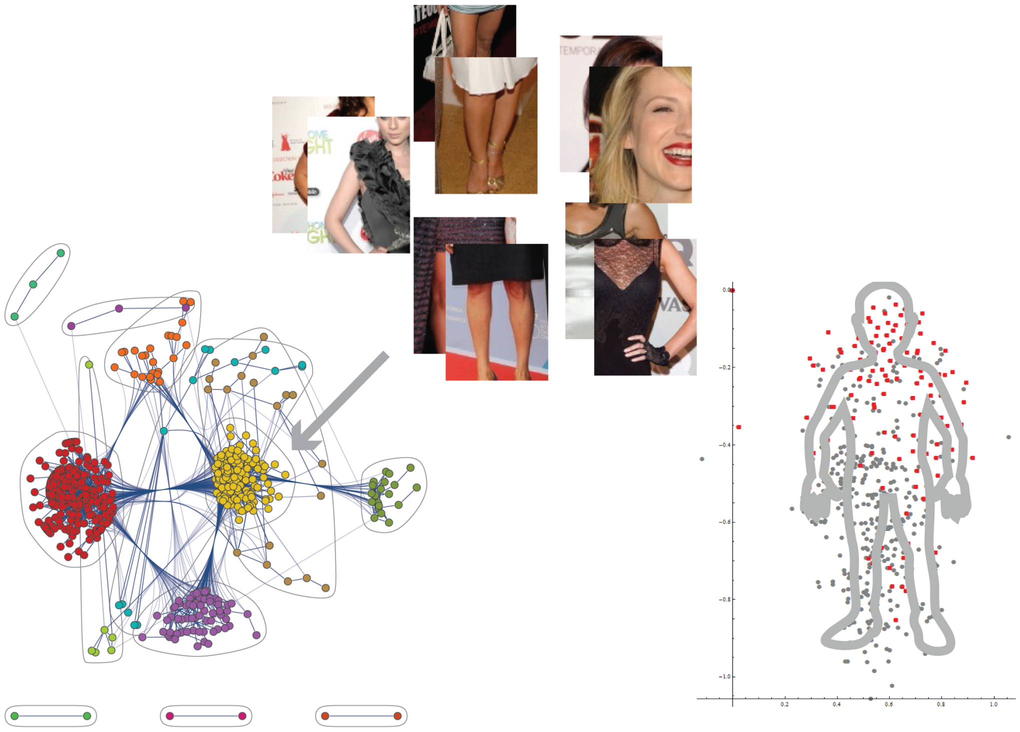


Figure 4.4: One community in the Spatial Relation Network (SRN) and the corresponding reconstructed global position plot.

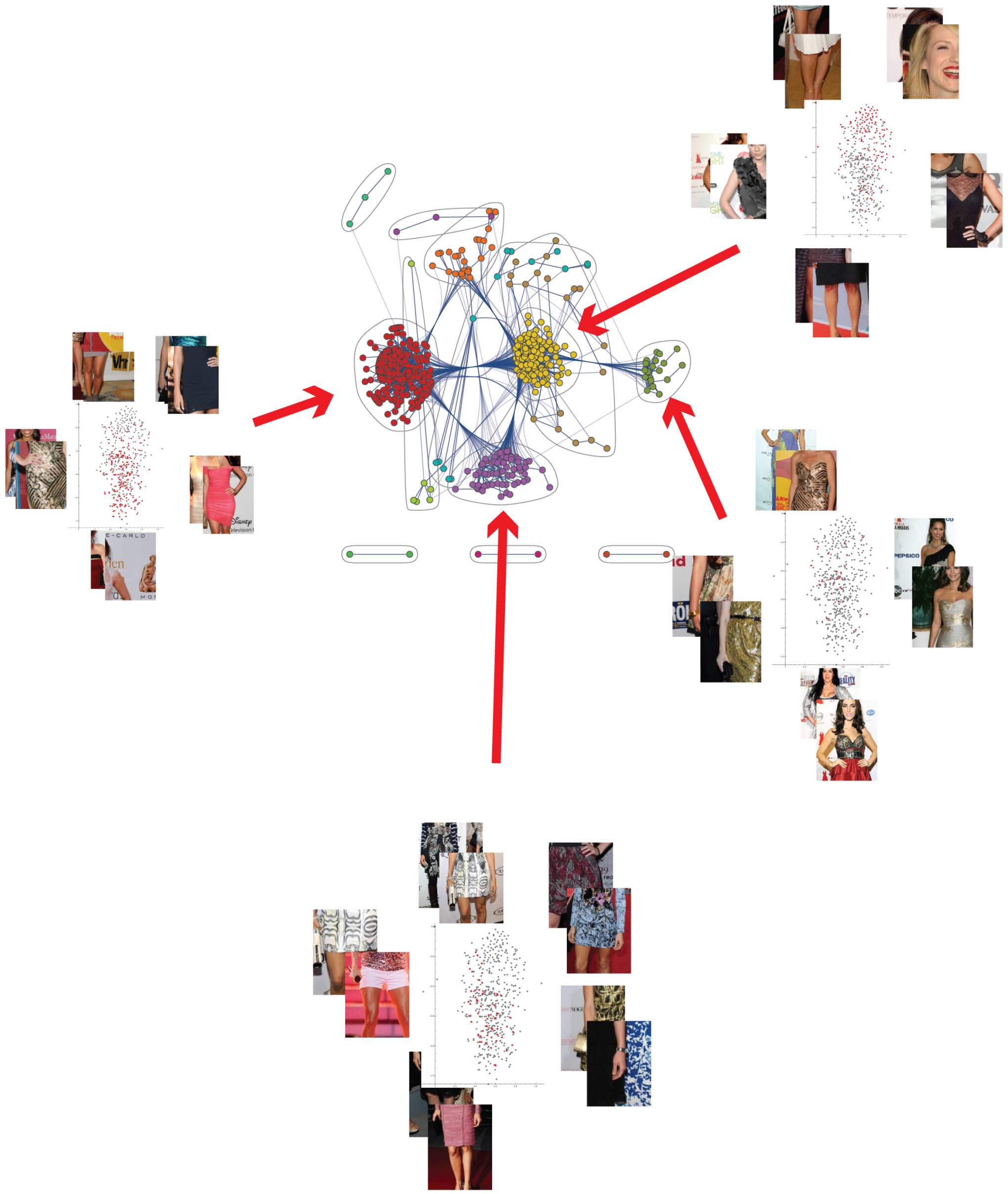


Figure 4.5: Communities in the Spatial Relation Network (SRN) and their semantic meanings.

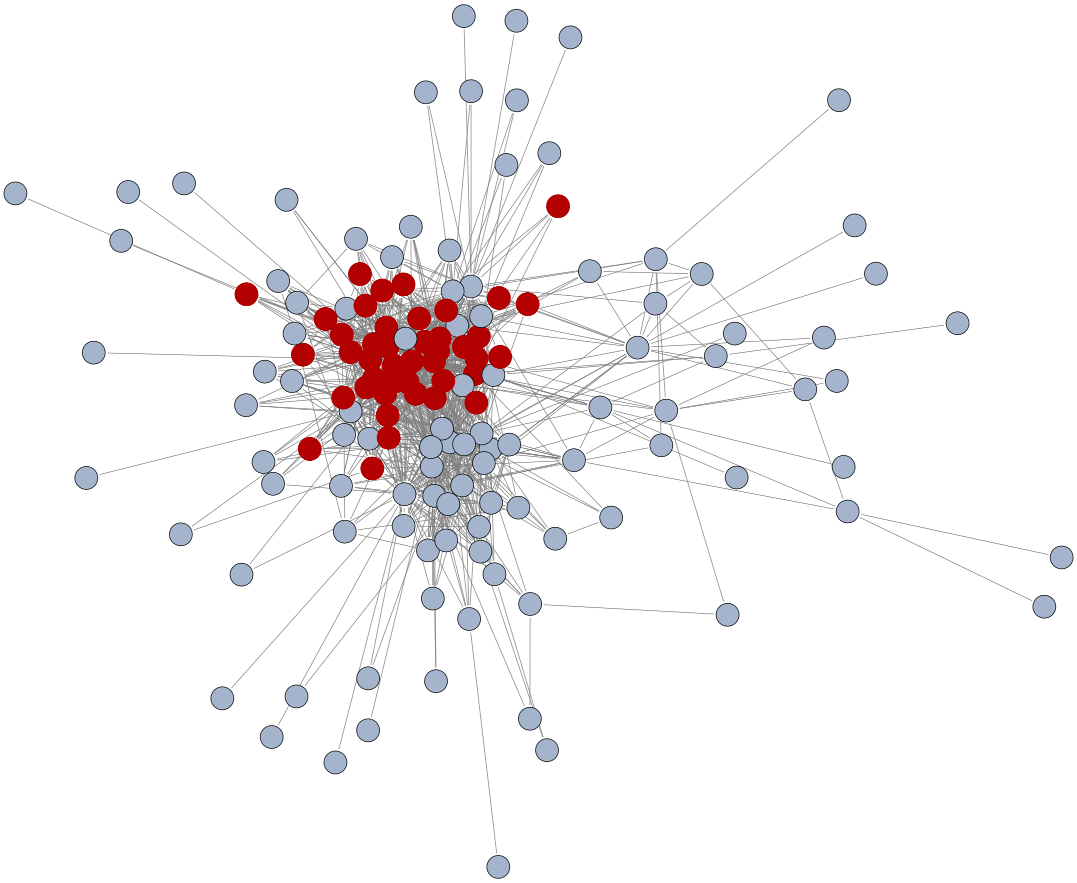


Figure 4.6: Head nodes (red) scattered in one SRN community.

However, although we separated body part nodes from background state nodes, the body nodes are still mixed together as a large community, with head nodes scattered everywhere. For example, 2 head nodes do not necessarily have a better chance to be connected. In fact, sometimes it might be even impossible, if these 2 head nodes are about different kinds of heads, such as long hair versus short hair, because let alone the existence of stable geometric relationships, they even won't have sufficient co-occurrences.

As a result, in order to concentrate head nodes further and recognize them as the same "part", the SRN is still less than ideal, and we need a network which can represent semantic meaning of its nodes better.

4.3.3 Applications of SRN

In SRN we have not only kept information about which pairs have stable geometric relations, but also the concrete values, and this enables us to perform the following operation that is critical to the design of "part" detectors discussed in Section 4.7: *Given an image patch of a specific viewlet, estimate the most likely location and scale of a representative image patch corresponding to any other viewlet.* Colloquially, if one sees the left shoulder of a person, where would the head of the person be (see for example, Figure 4.11)?

Thus, for each edge in the SRN, we define a pair of functions, f_{S_A,S_B} and f_{S_B,S_A} from the statistically learned Δ_x^*, Δ_y^*,r, to transform (i.e., translate and scale) patches of one state to that of the other. Given any patch P_{S_A} (i.e., a patch detected to be in viewlet S_A along with its coordinates and scale) we obtain the corresponding patch for viewlet S_B:

$$P_{S_B} = f_{S_A,S_B}(P_{S_A}) \tag{4.8}$$

For patches of states which are not neighbors, but connected via a path, $S_{A_1}, S_{A_2}, \cdots, S_{A_n}$, we have, by the transitivity of the translation-and-scaling transform,

$$P_{S_{A_n}} = f_{S_{A_{n-1}},S_{A_n}}(f_{S_{A_{n-2}},S_{A_{n-1}}}(\cdots f_{S_{A_1},S_{A_2}}(P_{S_{A_1}}))) \tag{4.9}$$

As a result, we can transform any states to a desired state, as long as they are connected via a path. In our study we verified the consistency and stability of the transitivity relationships empirically: Multiple paths connecting two viewlets yield very close transforms.

Other important applications of the SRN are discussed in Section 4.5, *where using iterative message passing algorithms on the SRN, we compute self-consistent values for the global positions of the "pieces" or viewlets and their scale values.* These computations allow us to establish the stability and consistency of the SUV framework.

4.4 Step III: Determining Parts via the Spatial Exclusive Network (SEN)

4.4.1 Functional Characterization of Parts

Going back to our SUV framework, a functional definition of “parts” follows intuitively: *Two distinct pieces correspond to the same part if they fit in the same relative location with respect to other pieces.* For example, two different viewlets representing say the head, can fit in very similar relative positions with respect to other pieces that represent arms or legs. In terms of the SRN, given a pair of nodes, A and B (say two different head nodes), and a third node, C (say a torso node), if edges $A \leftrightarrow C$, $B \leftrightarrow C$ both exist, and the geometric relations on these two edges are similar to each other, we claim that A and B are functionally similar to each other, and thus they are alternatives of the head part.

This definition has several advantages,

- First of all, it is consistent with our understanding of real world objects, which can be regarded as the combination of 2 portions: the atomic modules, and the way how these modules are organized.

- It is easy to represent this definition in our network, and it is much more feasible to use our geometric relation network to find nodes, which are the projections of different alternatives of the same part. We cannot use our SRN effectively if we used other definitions of parts such as using for example sophisticated CV segmentation algorithms,

Using this definition, we can extract patches, which represent the same real world parts.

4.4.2 Wedges and Triangles in the SRN

As we discussed in Section 2.2.3, a dual way of looking at the part definition would be as follows: Two pieces that are replaceable, or have a *mutually exclusive, but identical*

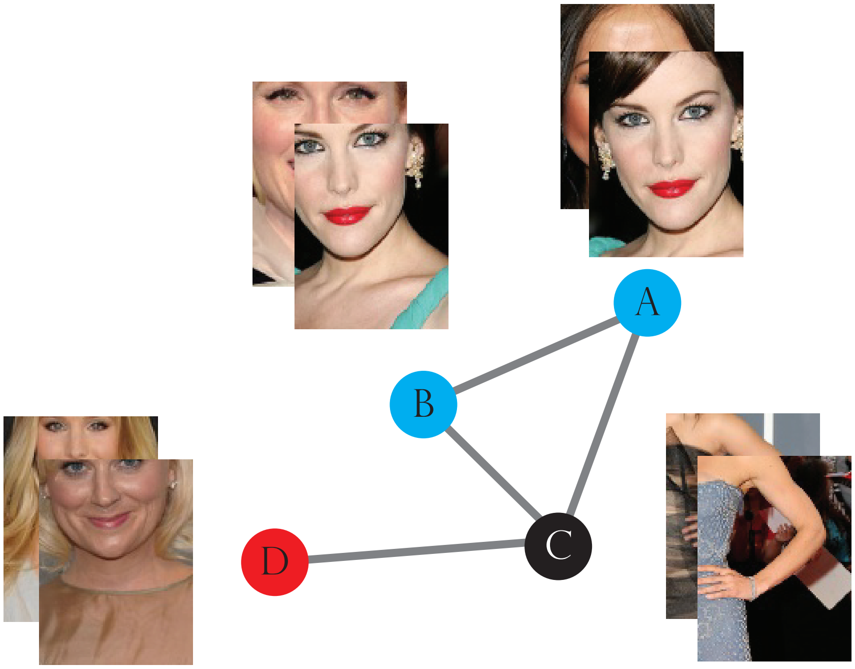


Figure 4.7: Wedge(B,D,C) and triangle(A,B,C) structures in the SRN.

geometrical relationship with other pieces (representing other parts). Thus, if two head nodes are visually different (See Figure 4.7) then the chances that they co-occur sufficiently many times and have a stable geometric relationship are very small, and there is no edge between these two nodes in the SRN. Thus, as shown in Figure 4.7, *we would observe a wedge* with respect to a third node (C), when two head nodes are of different types (like A and D in figure). However, if the nodes are only slightly shifted versions of each other (hence, a stable edge in SRN between them), like A and B in the figure, they also could *form a triangle* with a third node (C).

Using principles in Chapter 2, we now compute the Spatial Exclusive Network (SEN) from the SRN as follows: For every pair of nodes A, B, we first determine if they share at least two other nodes C and E, such that the geometric relationships $A \leftrightarrow C$, $B \leftrightarrow C$ are almost identical (i.e., the difference is within a small threshold) and the geometric relationships $A \leftrightarrow E$, $B \leftrightarrow E$ are also almost identical. We add a third node to suppress noise. If a pair of nodes A, B satisfies the above condition, we add an edge between the two to construct the SEN.

4.4.3 Properties

After applying community finding algorithms on the SEN, as shown in Figure 4.2(c), we observed that the network is further dissembled into small components, and each of them has a well-defined semantic meaning. Each community corresponds to a distinct human body part as also labeled in Figure 4.2(c).

In terms of HOG features, many viewlets in the leg community are closer to viewlets corresponding to the advertising wall than to each other, since the image patch is primarily that of an advertising wall. Our ability to group these leg viewlets together, demonstrates that we are successfully identifying the semantic meaning of images by leveraging network communities instead of purely computer vision features.

Figure 4.1 further validates our part-finding results. As explained in Section 4.5 we can use the SRN to compute global positions of every viewlet and as one can see, the viewlets corresponding to each part occupy distinct regions in 2-D almost defining a human body contour.

4.5 Community Properties

4.5.1 Global Scale and Positions

In Section 2.2.4, we presented that by solving optimization problems (Equation 2.43 and Equation 2.44), the global structure of the target object can be reconstructed.

In the SRN, edges are representing stable spatial relations of endpoint viewlets. For a densely connected component in this network, to reconstruct the structures of the entire system, we decide to take advantage of that relative information to calculate a set of absolute locations for all meaningful viewlets while minimizing the distortions. This task is in some

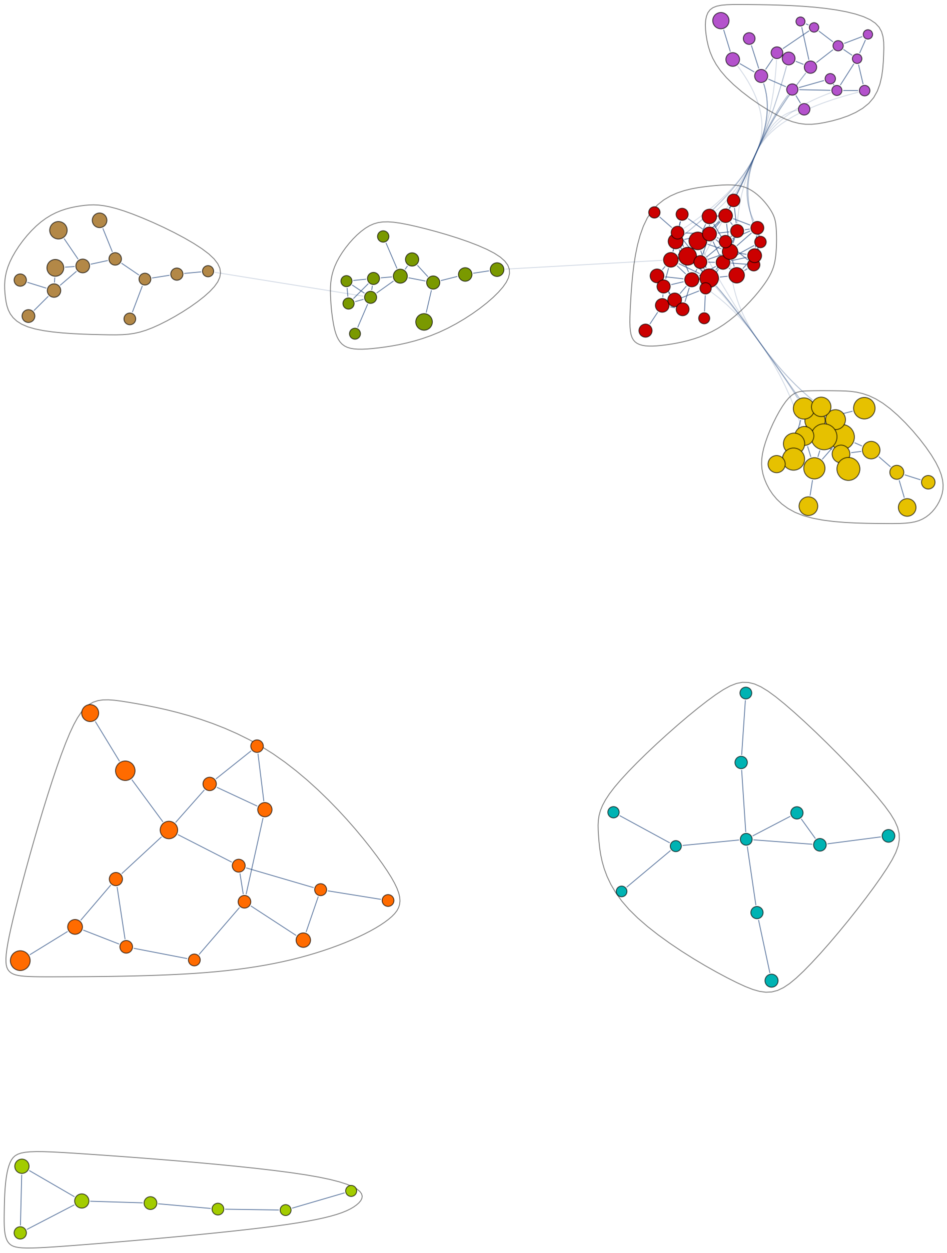


Figure 4.8: Communities in the Spatial Exclusive Network.

ways similar to that of some previous works, like Multi-Dimensional Scaling (MDS [Kru64]). We derive an iterative approach to calculate the position and size of each node using the relative positions and size of its neighbors. The approach are summarized in Algorithm 1.

Algorithm 1 Global structure inference

```
Require: $G(V, E)$ Edge set $E$ nonempty; Vertex set $V$ nonempty.
  $i_{hub} \Leftarrow get\_maxdegree(G)$
  $x_{i_{hub}} \Leftarrow 0; y_{i_{hub}} \Leftarrow 0; s_{i_{hub}} \Leftarrow 1$
  $visited \Leftarrow visited \cup \{i_{hub}\}$
  repeat
    $shift \Leftarrow 0$
    for all $i \in visited$ do
      $N \Leftarrow get\_neighbors(i)$
      for all $j \in N$ do
        $x_\delta \Leftarrow get\_inferred\_x(x_j, j, i) - x_i;$
        $y_\delta \Leftarrow get\_inferred\_y(y_j, j, i) - y_i;$
        $s_\delta \Leftarrow \log(get\_inferred\_s(s_j, j, i)) - \log s_i$
        $shift \Leftarrow shift + x_\delta + y_\delta + s_\delta;$
        $x_i \Leftarrow x_i + get\_weight(j, i) \times x_\delta;$
        $y_i \Leftarrow y_i + get\_weight(j, i) \times y_\delta;$
        $s_i \Leftarrow s_i \times \mathrm{e}^{get\_weight(j,i) \times s_\delta};$
        $visited \Leftarrow visited \cup \{j\}$
      end for
    end for
  until $shift \leq \epsilon$
```

From Algorithm 1, we get global position assignments for all nodes in the largest component of the Spatial Relation Network. As we discussed in Section 4.3, there is a community of human body parts. We plot all nodes using the global coordinates we got, and have nodes from body part community highlighted, as shown in Figure 4.9 and Figure 4.1.

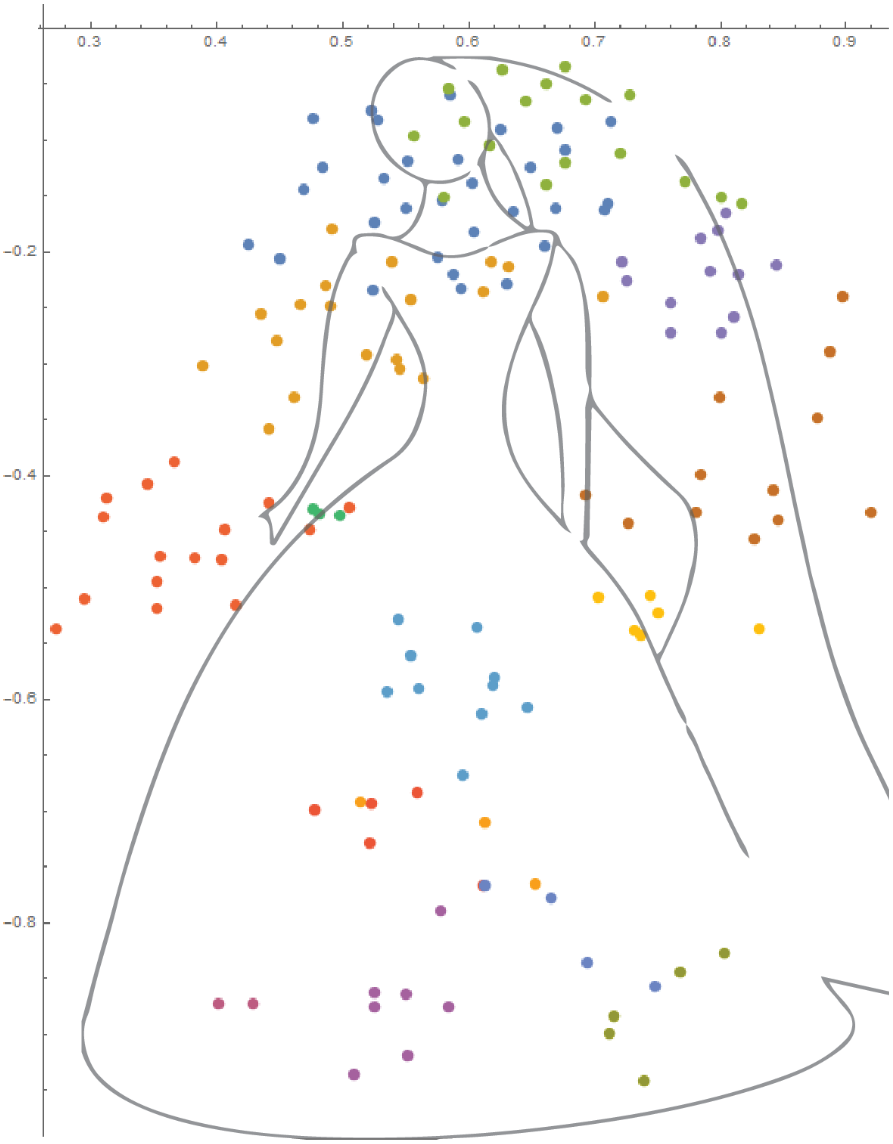


Figure 4.9: Global structure reconstruction.

From Figure 4.9, we notice that the community partition of the Spatial Exclusive Network (SEN) is highly correlated with the global geometric value we derive from the SRN. Furthermore, the global positions of the nodes in these communities of the SEN mirrored the human body in real world. Our ability to reverse engineer human body structure demonstrates that we can are successfully identifying the semantic meaning of images by leveraging network communities instead of purely hard knowledge encoding (manual tagging, specific features, etc.)

Using the same algorithm with some slight modification, as discussed in Section 2.2, we can extract a global scale value for each meaningful viewlet. Some examples of these values are listed in Figure 4.10 with corresponding images. We can see that while the nodes from some communities of the SEN are all sharing similar scale values, and the values themselves, are in good accord with our understanding of the corresponding parts.

4.6 Dataset and Feature Descriptor

4.6.1 Data Description

To test images with flexible objects with more diverse backgrounds and resolutions, we also collect a dataset, which consists of 12 047 high quality celebrity images crawled from web, of various resolutions and aspects, with an average size of 472 × 665. We believe that using celebrity image set can maximize the variety of dresses and body gestures, which occupy most of the area of an image, and can easily fool many densely sampling object detectors. For the celebrity dataset collected, we used 9638 images as the learning set, and the rest as the test set. To check the localization result bias with high precision, we use the head part as the anchor, and manually annotate the test set.

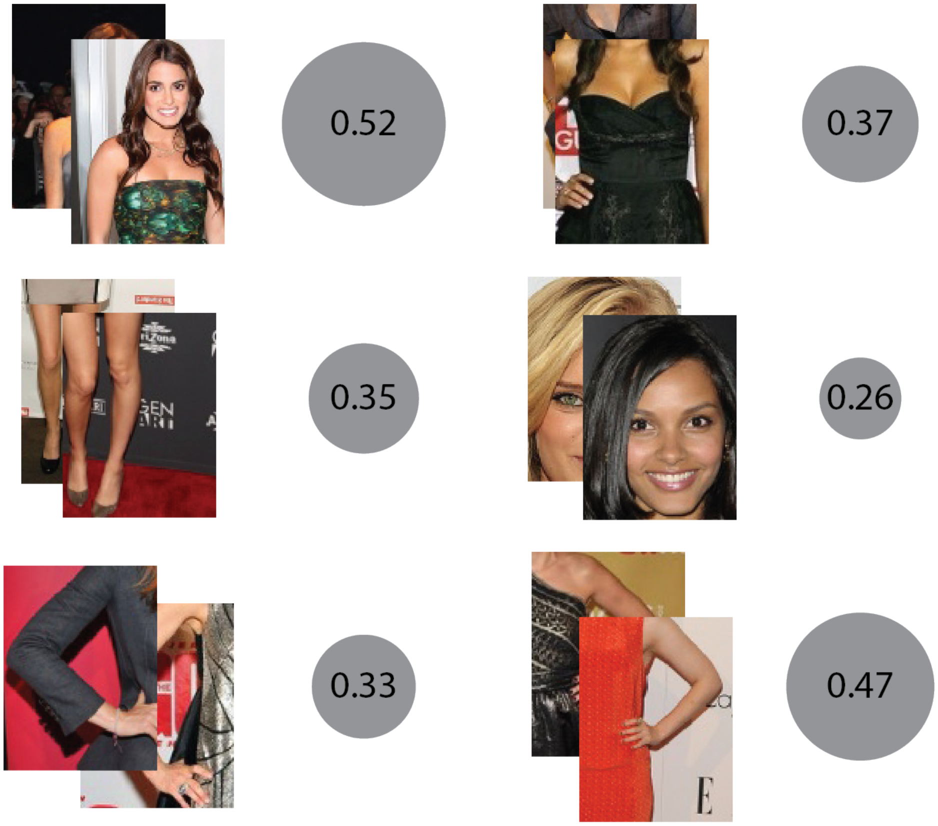


Figure 4.10: Global scale values for some body viewlets.

4.6.2 Feature Descriptor

For each image in the learning set, a multi-scale pyramid representation is created with a step ratio 1.2. At every level, fixed-size samples are collected randomly. In this process, increasing sampling densities is used to obtain datasets of larger size, until there is no significant change in variance of the resulting vector set (i.e., estimated via the number of required dimensions to retain 95% variance in PCA stage). When images are being converted into descriptor vectors, Histogram of Oriented Gradient (HOG) [DT05] is used with the following parameters.

- Cell and Block size — We use 128×96 image patch, each of which can be regarded as 16×12 cells, with each cell 8×8 pixels. For each pixel, we calculate gradient for all the 3 channels and collect the maximum value. Histogram of Gradients are calculated for each cell. Each histogram has 27 values. 18 of them are gradient bins from 20-degree-segments of the 360-degree direction space, and the rest 9 are for energy bins, which are the sums of pairs of absolute values of opposite-orientation bins out of the preceding 18 bins.

- L2-hys normalization — L2-hys normalization is used on each block to make the feature descriptors robust to the illumination conditions.

After feature extraction, PCA is applied for more compacted representation and dataset variance evaluation to find the best sampling density.

4.7 Experiments

In Section 4.5, we re-constructed the real world structure from our network model, and derived many related properties for the part we discovered, the close consistency between the derived value and ground-truth further endorses the correctness of our model.

Because of our speedy algorithms, we can search the entire image densely for the best position of each part, instead of just checking much sparser points of interest (which are generally the results of corner seeking, and entropy maximizing). As a result, we can also perform object localization with high precision, and localize multiple objects in the same image.

4.7.1 Learning

After densely sampling our learning set, we obtain a patch set consisting of 239 856 image patches. These patches are converted into image features as described in Section 4.6. We do multiple passes of K-Means clustering with $k \in \{250, 500, 1000, 2000\}$ to search for the best k, and obtain a vocabulary with 1006 visual words ($k = 1000$ is picked as the best, after that we test the K-Means convergence with many runs indicated by the last digit of k, the version with $k = 1006$ is picked and used in the later stage.).

The SEN/SRN construction process is similar to what we described in Section 2.2. After this, we pick the head related community in SEN. Viewlets in this community are all head templates. From these viewlets we further pick 3 most densely connected ones, which usually represents head part best, as core nodes (i.e. the canonical representation of the head). Because of the spatial information augmented on all these edges, we can calculate a geometric transform from any other object node to these core head nodes. In addition to this, we also pick the best mapping (with the lowest variance) from each of the SRN nodes to one of the core ones. As show in Figure 4.11, these mapping will help us to tweak detected ones to the best position.

4.7.2 Detection

4.7.2.1 Reference-Point-Based Grouping

After the preceding stage, we get 46 active viewlets/nodes out of 1006 we have. Now, using the dense sampling method as described in the HOG paper, we create a pyramid for each input image as we did for the training ones. However, instead of doing a random sampling, we shifted the detection window in an exhaustive manner for the sake of completeness. As

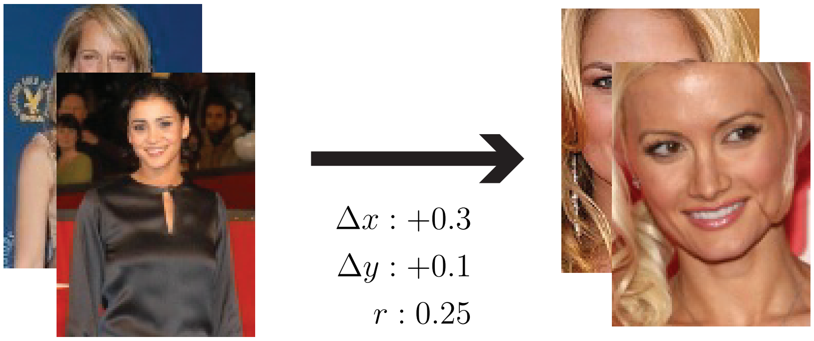


Figure 4.11: Mapping detected head to the best position using spatial relations: In this example, according to the information on edge, to transfer a detected patch (left ones) to the head representation, we must shift towards left 0.3 × width, shift downwards 0.1 × height, and shrink to 0.25 of the original size.

described in preceding section, a HOG calculation and PCA matrix multiplication are applied to each patches, which is then classified into one of 1006 viewlets using k-Nearest-Neighbor approach. If there exist one or more patches, which are classified into an active viewlets, transform rules are applied to get the precise head position. One face can be detected from multiple detections of the 46 states/nodes, which are grouped by their final position after applying geometric transform. A confidence score would also be given according to the number of corresponding detected nodes, and how good they are. In our experiment, we are rejecting all groups with only one detection, which considerably boosted the precision of our detector.

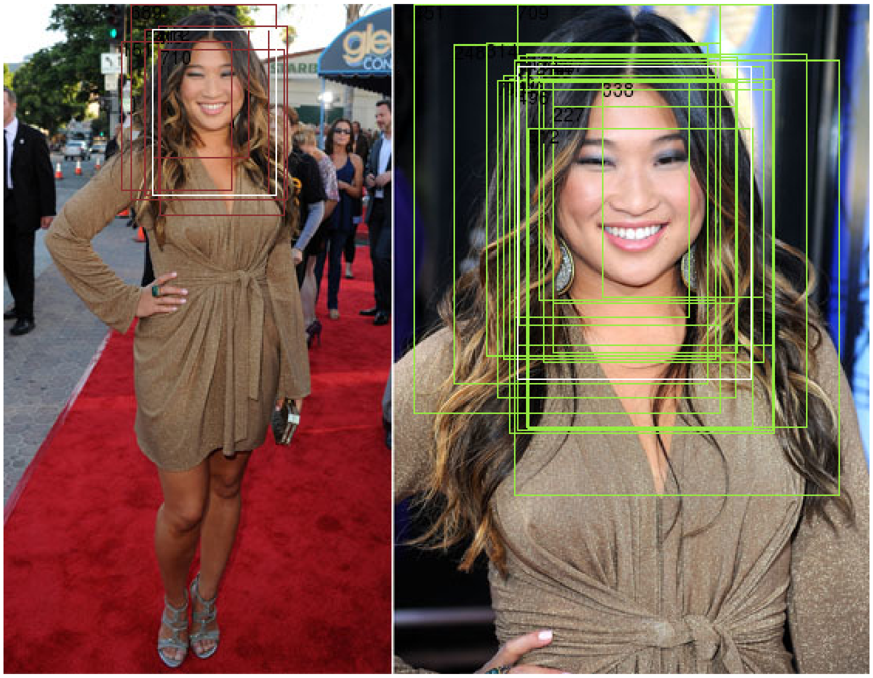
4.7.2.2 Network-based Grouping

Instead of grouping patches by the inferred position of head, a pure network-based approach as described in Section 2.3 can also be applied. For each image in the test set, we apply the same technique we used in Section 2.2: densely sampling, patch-to-visual word mapping, candidate patches discovering and grouping as we described in Section 2.3. After all of these, we can get one giant connected component for images with only one celebrity in it, and we are also getting *2 or more large groups* in images with more than one celebrity, as seen in Figure 4.12. The faces' positions are then determined by all the patches in the same group and the corresponding spatial transforms. In addition, a confidence score would also be given according to the number of corresponding detected parts, and how good they are as described in the preceding section.

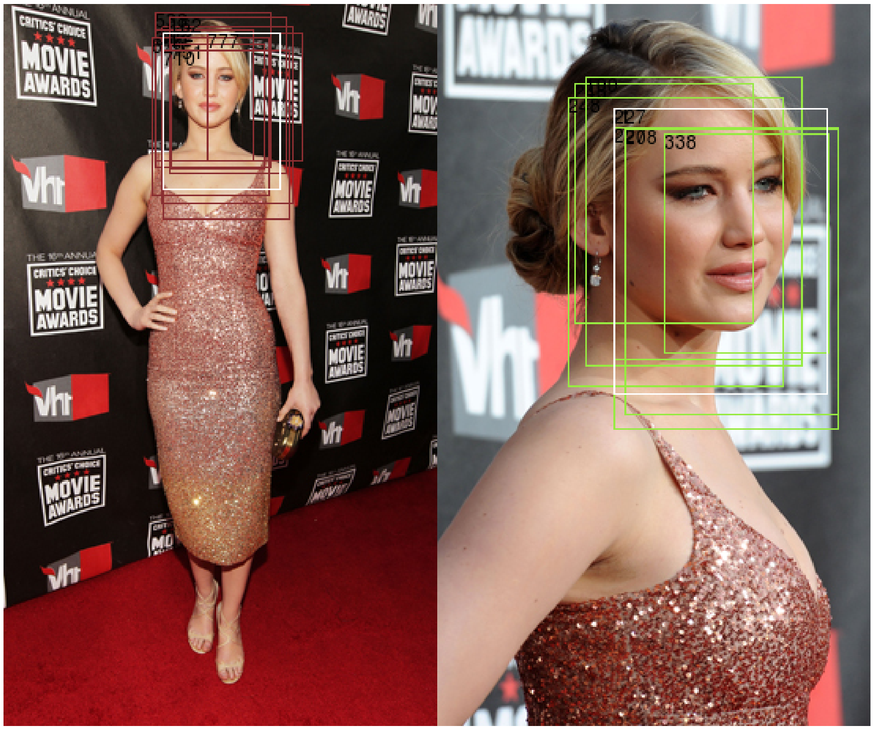
4.7.3 Results and Comparison

For comparison, we choose OpenCV's Haar-cascade implementation [Man01] of Viola-Jones algorithm [VJ01, LM02], with its exhaustively trained model, which is widely accepted in industry. The OpenCV face detector is based on Haar-like features. It has a decision-tree structure, where several classifiers are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The classifiers at every stage of the cascade were created by boosting of basic classifiers using one of four different techniques: Discrete Adaboost, Real Adaboost, Gentle Adaboost, and Logitboost.

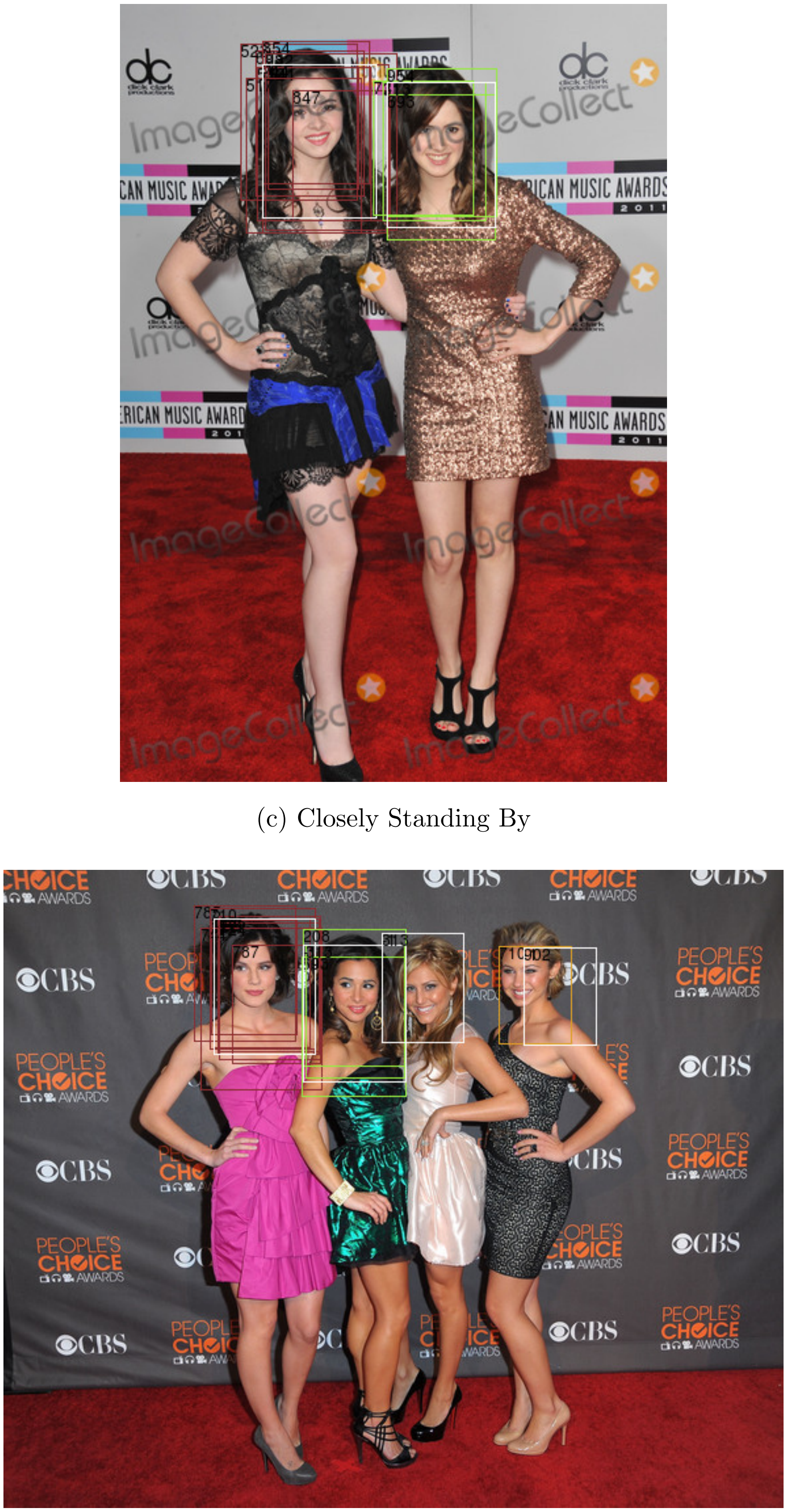
We must point out that, the Viola-Jones detector in OpenCV needs a large dataset of positive samples for training. This requires manual labeling of thousands of images to cover all the race and age groups, emotions, and facial hair styles. Furthermore, each of these samples is deformed in order to create additional training data (e.g., rotated, placed on an arbitrary background, and have levels changed), to increase the chance of hit during the detection stage. In contrast, our algorithm requires minimal supervision. We only need to specify a core community representing the head from SRN, and all related body part communities of different semantic meanings will be activated automatically. Those body part communities already have the geometric relations to the head nodes learned during the Spatial Relation Network construction stage. Moreover, each of the communities consists of all different nodes of various appearances sharing the same meaning, since the representation completeness has already been handled by our network.



(a) Different Scales



(b) Different Viewpoints



(c) Closely Standing By

(d) Multiple Celebrities

Figure 4.12: Images with more than one celebrity. Each celebrity has her own corresponding group.

Table 4.1: Experiments on celebrity dataset.

	Haar-like	Ours
True Positive	3072	2899
False Positive	972	58
Precision	76.0%	98.0%
Recall	92.9%	87.8%

Our test set, as described in Section 4.6, consists of 2409 images, most of which are about celebrities and with various scenes and resolutions. The faces in this test set were carefully labeled by multiple subjects, to establish the ground truth, so that performance results can be computed for the different detectors. The results are presented in Table 4.1. These results demonstrate that our approach achieves a much higher precision while maintaining a reasonable coverage. By dropping nodes from our detector set, we also obtained the corresponding ROC curve, as in Figure 4.13.

We can see that our approach achieves high precision very quickly. It should be ascribed to that for each face, decision was made from various sources, including shoulder, upper body outline, hair, and even arms, etc., which rigorously provide high robustness throughout hard false patches.

This also demonstrates the representative power of our network. The network organizes many *viewlets* together to represent one object, even if these visual templates are far apart in feature space. In contrast, if we want to identify all of these representations in a SVM or a boosted approach, all of them should be manually encoded, or discovered using other sources, like the Internet [KHE12], before they can be included in some positive areas in the feature space separated one or a few hyper-planes (half space in the case of SVM).

Figure 4.14 demonstrates how the network-based matching and spatial transform rule works. Through using head as an anchor to show with our model the localization bias can

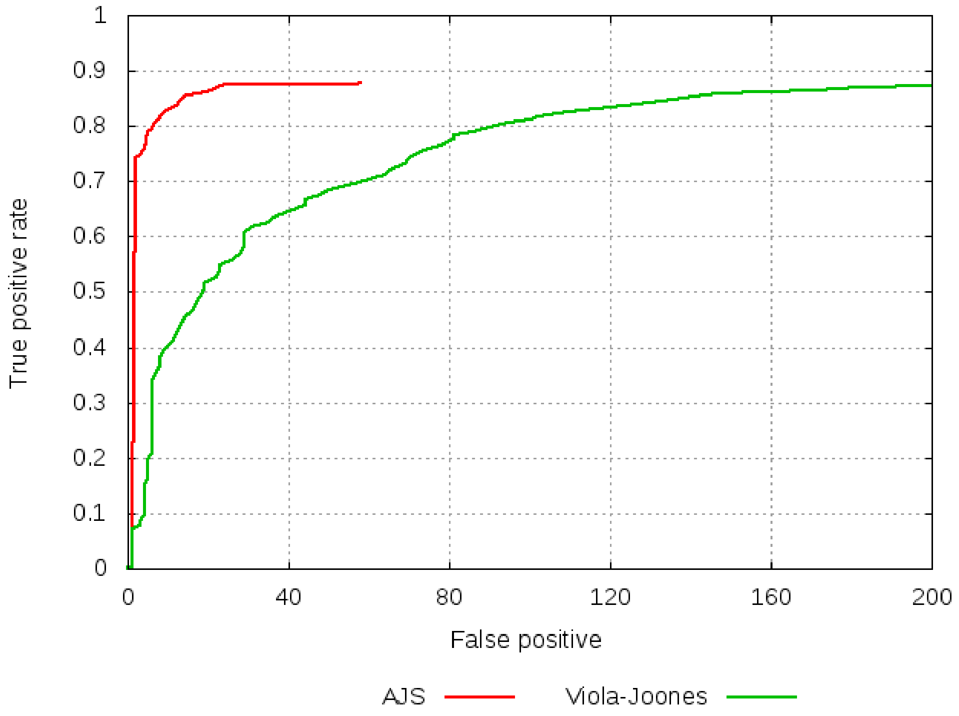
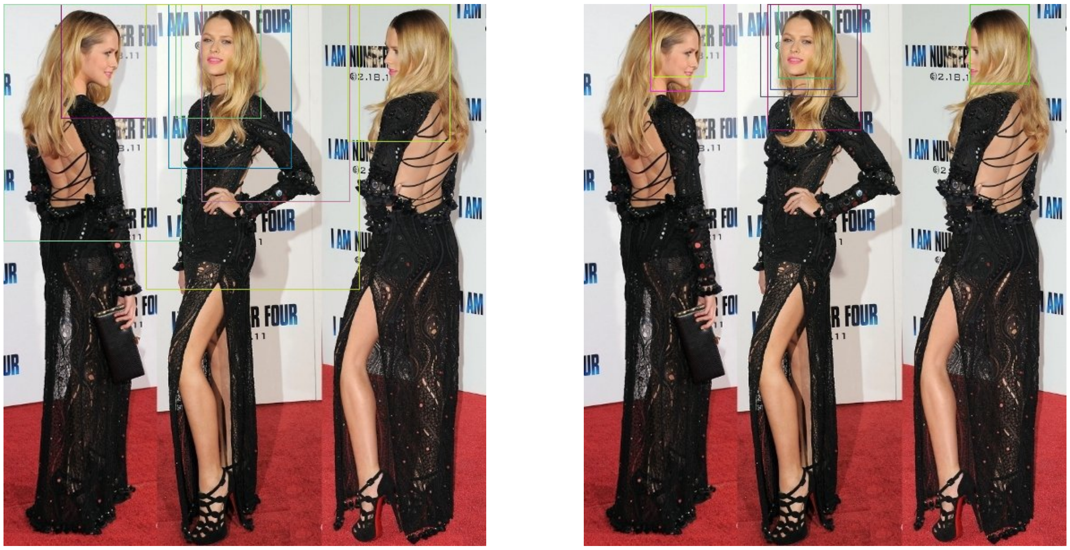


Figure 4.13: Face/head detection performance on celebrity dataset.



(a) Detected Part

(b) Inferred Head

Figure 4.14: Examples in face/head detection.

be measured more precisely. There are questions remaining unanswered: Since head part is relatively feature rich, the main localization power may solely come from itself, while other parts aren't as precise.

To study the contribution made by the other part, firstly, we enriched the test dataset with many difficult ones. In the current version, there are images like Figure 4.14. While it is possible that the middle one can be handled by a good face detector, by no means the left one or the right one can be detected solely relying on face. As a result, the other parts are playing a big role here, in Figure 4.14(a) we can see the activated patches, and in Figure 4.14(b) we can get a rough idea that how the spatial transform rule works.

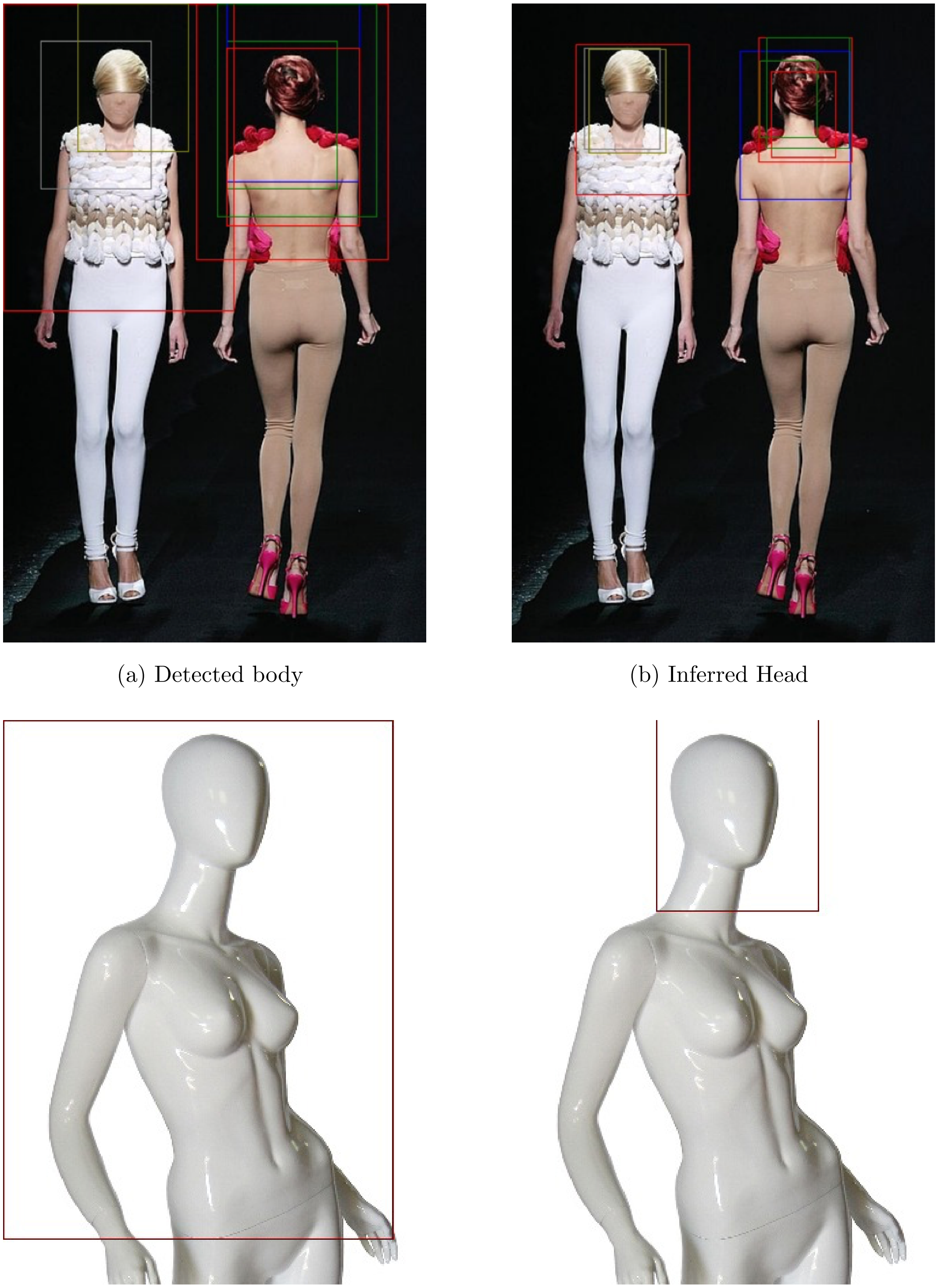
Moreover, We try some more abstract images, like Figure 4.15. They are both interesting examples not only because of an outright absence of face feature, but they are dramatically different from any of images in our training set. However, we observe that our detector is able to take advantage of the very high level structure information encoded in the network (half body) to capture the outline in certain viewlets, and using the rules on the path to the ground-truth head viewlets, the detector was able to deduce the head's position.

4.7.4 Torso Localization

Comparing to head, torsos are much harder to localize, for the following reasons,

Lack of Category Characteristics or "Cues" Unlike face, which can be easily identified with features like eyes, nose, month and their relative geometric relations, torsos do not have that kind of characteristic "cues" which can have us to differentiate a torso patch from a non-torso patch.

Huge Within Category Variety The visual of torsos is vastly determined by the dress/clothes patterns. With a huge variety of dress patterns we have today, the corresponding descriptors will not be clustered in the feature space.



(a) Detected body

(b) Inferred Head

(c) Detected body

(d) Inferred Head

Figure 4.15: Detection of the face/head part using our model even when the facial features are absent.

Table 4.2: Torso detection performance.

	Approaches		
	DPM ([FMR08])	Poselet ([BMB09])	Ours
True Positive	1239	3115	2810
False Positive	5263	1678	108
Coverage	38.3%	96.3%	86.9%
Precision	19.1%	65.0%	96.3%

These points make a rigid-template-based torso detector impractical. However, part-based models, with their factored object representation and geometric relation modeling, are usually capable of "detecting" torsos by inferring from the real detected parts. Thus, torso detection becomes a great benchmark for deformable part-based approaches. We also compare our model with 2 other important part-based approaches we introduced in Chapter 3.

The learning process is similar to what we described in preceding Section 4.7.3. The difference is that, this time, we are picking torso nodes as the core nodes, and learning transform rules accordingly. Some results are demonstrated as in Table 4.2.

For the Deformable Part Model(DPM) [FMR08] approach, we used the person model [TP13] trained from PASCAL VOC 2007 dataset with the default threshold used in the code. For Poselet [BMB09] approach, we used the April 2013 release posted by the authors (which can be found on `https://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/poselets/`) with threshold 3.6, the value used by the author in PASCAL VOC 2007 competition.

4.8 Summary

In this chapter, we implemented the novel data driven framework described in Chapter 2 for unsupervised learning of unknown objects represented in the image corpus. The approach introduces the construction of a novel statistical SUV model, where first the corpus is broken down randomly into image patches. These patches are then clustered into groups to create *viewlets* to be designed. Once the viewlets are defined, then the entire task of finding parts is reduced to a large-scale data mining problem, via a network representation (SRN) that captures the geometric relationships among the viewlets. This representation exploits the fact that object-related pieces will have stable geometric relationships among them. Next, an intuitive and computational definition of parts, allows us to exploit the structure of the SRN to define the part-based network SEN as in Section 2.2.3. We applied the SUV framework to an image corpus crawled from the Internet and successfully extracted the different human body parts. As a further evaluation, we built in a completely automated fashion, a k-Nearest-Neighbor detector for the head/face part and showed that the resulting precision is very high and compares very favorably with existing approaches that are based on supervised learning from large-scale tagged data.

We also note that the SUV methodology is particularly suited for processing Internet-scale image corpora that have an unknown but specific bias with regards to the underlying objects. Blind creation of such corpus is not difficult and various approaches can be used; e.g., (i) grouping streamed images by breaking them up into temporal intervals will ensure that the context does not change abruptly, or (ii) all images labeled with the same keywords (e.g. Academy Awards: Oscars 2015) will have a bias towards the same object set (e.g., celebrity images). All the computation steps are highly scalable and can be easily parallelized and adapted to the Map-Reduce framework for cloud computing purposes so that very large-scale corpus can be processed.

CHAPTER 5

Multiple Prototype Learning

5.1 Introduction

In this chapter, we further study the model by learning multiple objects simultaneously from the Caltech-101 dataset [FFP06].

Same as preceding chapter, we first build the network-based framework as described in Chapter 2 and Chapter 4. As in preceding chapters, using the scale attribute on vertices and relative spatial ratios on edges, the network is scale-invariant and translation-invariant to the resolution variety of images. With the algorithm described in Section 2.2, we can find viewlets of the same semantic part (like head, arm, etc.) using the structural similarity of nodes in the network and spatial information on the augmented edges. As in Chapter 4, we also show that global spatial information can be reconstructed completely from pairwise information. Studies about reconstructing global structure from pairwise information can be tracked from Multi-Dimensional Scaling (MDS) in 70's [Kru64]. Recent researches ([YAK13, CJA14]) also show that with a large amount of data it is possible to extract the structure without all $n(n-1)/2$ pair values.

Furthermore, we also implement the network-based grouping described in Section 2.3 for detection. The network-based grouping eliminates the need of picking anchor nodes and extracting transform rules, which makes the entire pipeline fully automated. In view of the patch-level grouping and suboptimal structure described in Section 2.3, the multi-object-in-one-image problem is handled natively. What's more, the resulting groups also come with a reliable confidence measure of the predication, which is the number of semantic parts covered by the group.

In the experiment, to learn models from images of 4 different categories (airplane, car, face, motorbike), 2 different cases are designed by whether a shared dictionary or separated ones are used:

In the shared dictionary experiment, we use patches from all categories to construct an 800-viewlet visual dictionary, based on which the network for each category is learned as described in **Chapter 2**. Then, SVM classifiers are trained using the confidence score from learned model as inputs. The resulting classifier outperforms the approach by Fergus et al. [FPZ07], though theirs were taking more advantage from using patches of other categories as negative samples for discriminative training.

In the separate dictionary experiment, we demonstrate that when the images presented to our learning algorithm are beyond experience (regions in the feature space not covered by the dictionary (knowledge)), structure information can still be extracted and abstracted to form prototype representation. The resulting detectors outperform Fergus' counterpart work [FPZ03] vastly.

5.2 Experiments and Results

In Section 2.2.3 we re-constructed the real world structure from our network model, and derive related properties for the discovered parts. The close accordance between the derived values and ground-truth further supports the representative power of our model. Based on the learned model, we design a series of experiments including both object detection and localization.

We evaluate our approach with the widely-used Caltech-101 dataset [FFP06]. The input of our experiment is a set of unlabeled images describing the same object without any extra information, the output is a visual vocabulary and a learned model, which includes a set of foreground related visual words from the vocabulary, an SRN (Spatial Relation Network)

Table 5.1: the Caltech-101 image count per category.

Category	Total	Training	Test
Face	435	218	217
Motorbike	800	400	400
Airplane	800	400	400
Car	800	400	400

with its parameters, and an SEN (Spatial Exclusive Network) with its part-based *viewlet* communities.

5.2.1 Data and Feature Descriptor

Similar to Section 4.6.2 , we are using Histogram of Oriented Gradient (HOG) [DT05] features in our experiments with the same setting. For the Caltech-101 data, in order to compare with previous work, we pick the same 4 categories: rear view of car, airplane, motorbike and human face. In addition, the same half-half partition of training and test data is used as in [FPZ07].

5.2.2 Shared Vocabulary

After densely sampled training images from all 4 categories, we obtain 105 164 patches, each of which is converted into a 427-dimension feature vector. For feature space discretization, we firstly search a good K through $\{100, 200, 400, 800, 1600\}$ by checking the average entropy between the assignments of current iteration and previous one. $K = 800$ is chosen as the vocabulary size, upon which we did K-Means clustering on the feature vectors of all patches.

To create a Spatial Relation Network for each category, we sweep over all training images again. During the scanning, all co-occurrences are recorded and the statistics of spatial

Table 5.2: Shared vocabulary model statistics for the Caltech-101 dataset.

	SRN			
Category	Nodes	Edges	Avg deg.	SEN Parts
Face	226	1097	9.71	16
Motorbike	199	1411	14.2	20
Airplane	245	730	5.96	9
Car	251	185	14	24

relations are aggregated for all pairs of nodes as described in Section 2.2. Directed by the desired sparsity of networks (i.e. the average degree in the largest connected component), we search proper threshold for each category. Consequently, 4 networks, one per category, are obtained, as seen in Table 5.2.

After that, to group nodes by semantic part, we create our Spatial Exclusive Network (SEN) using a threshold for the tolerance. The SEN computes whether the two relevant spatial relations in a wedge/triangle are identical or not, which results in many small clique-like components. Using the SEN, we grouped most of SRN viewlets into different communities by their structural and spatial similarity. After that, we get several parts (sets of semantic equivalent viewlets) out of each SRN, as shown in Table 5.2.

For detection, given a query image from the test set or any other sources, we are going to construct a patch network as described in Section 2.3. Firstly, we create a pyramid as we did for the training ones. However, instead of doing a random sampling, we sweep the detection window in an exhaustive manner for the sake of completeness. As described in preceding section, a HOG calculation and PCA matrix multiplication are applied to each patches, which is then classified into one of K states using k-nearest-neighbor approach. As in Section 2.3, edges are constructed between pairs of patches which have a similar spatial relation as the corresponding node in SRN. For all m patches in an image pyramid, after checking all the $\frac{m(m+1)}{2}$ pairs, we would get one or more connected components (the graph definition here is slightly different from that of the model, since nodes of the graph represent patches, instead of viewlets as defined in Section 2.2, see Section 2.3 for more details).

After obtaining patch groups defined by the connected components in preceding paragraph, we looked into each group and evaluate the quality as an object embedding. Every patch in the group is checked so that if there exist two or more patches classified into the same nearest part, which is defined by communities in SEN. The corresponding part is considered present, and the part count increments by one. Once the part count is large enough (threshold can be adjusted according to coverage/precision requirement as described in Section 2.3), a object predication would be made. Before the part counting stage, the number of patches in group can also be regarded as a preliminary confidence score measure. In our experiment, we are rejecting all groups with only less than 5 patches, which considerably reduces the number of candidate groups.

The process described in preceding paragraph will be carried out 4 passes with different models, and from each one we can get one giant connected patch graph and the associated score (semantic part count). SVMs are trained using these 4 scores as input, which is used to predict the most probable category for query images. Some of our results are presented in Table 5.3.

5.2.3 Separated Vocabulary

In the preceding experiment, a shared vocabulary was used, which implies that when an image from other than current category is presented to a model, at least each patch, is within the agent's knowledge and can be correctly assigned to a viewlet. So questions might arise that if the discriminative power is truly from the spatial and structural information from the model, or is just from the visual words (like bag of visual features, or other structure-less

Table 5.3: Confusion matrix for shared vocabulary model on the Caltech-101 dataset.

	Recognized category				Fergus et al. [FPZ07]			
Query Image	F	M	A	C	F	M	A	C
Face	0.982	0.000	0.018	0.000	0.862	0.073	0.028	0.014
Motorbike	0.000	0.990	0.010	0.000	0.000	0.977	0.013	0.000
Airplane	0.005	0.013	0.967	0.015	0.003	0.042	0.888	0.060
Car	0.000	0.000	0.020	0.980	0.008	0.092	0.197	0.670

models.).

To answer this question, we carry out another experiment on the same dataset. This time, instead of a shared vocabulary constructed from patches of all categories, we create one separate dictionary for each category. Because the diversity is considerably less in the patch set from a single category than that of mixed ones, smaller candidate K's are proposed during searching process. However, to control variables to compare with previous experiments, we enforce $K = 800$ in three categories: airplane, motorbike and face, while using $K = 400$ in the car dataset, since most of the images looks highly similar.

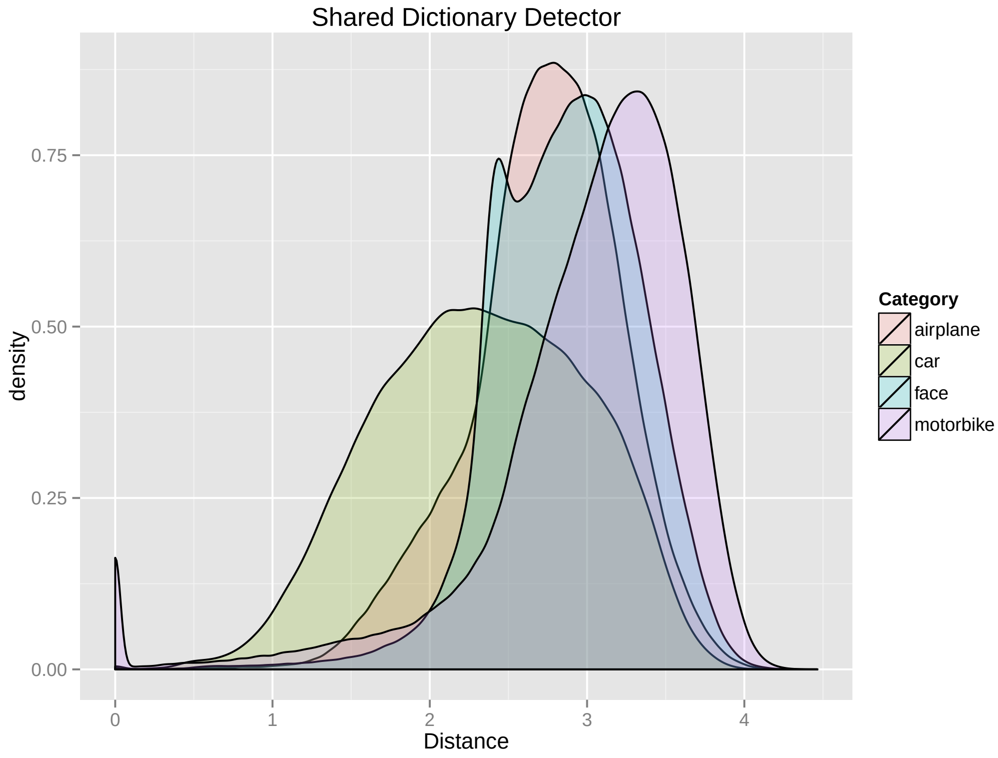
Because the dictionary of each experiment is solely extracted from the training images of one category, we expect that it is not complete enough to cover feature vectors from images of other categories. To measure feature space completeness of these dictionaries to images of different category, a good approach will be to check the distribution of distances from each queried vector to the nearest neighbor vector of the assigned viewlets. Ideally, if the coverage is good, we expect the neighbor to be contained in a sphere of similar radius, and the distance distribution will be close to a Gaussian one. From Figure 5.1 we can see that, comparing with the shared dictionary model, being tested with images from other than native category, models trained from single category dictionary show distortion to different

Table 5.4: Separated vocabularies model statistics for the Caltech-101 dataset

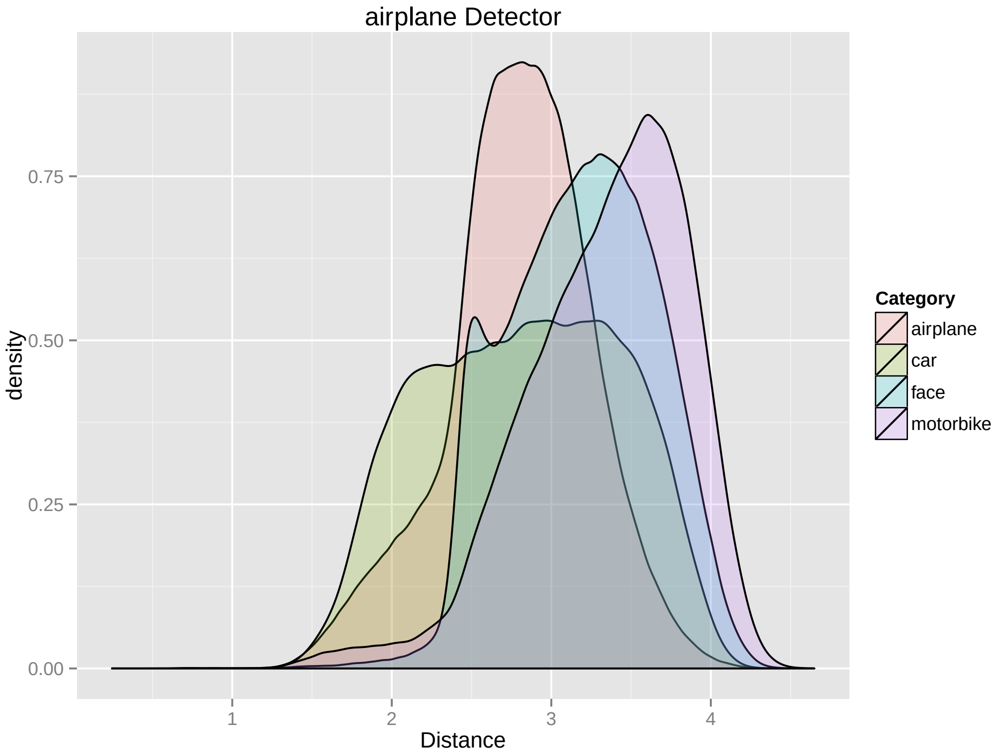
	SRN			
Category	Nodes	Edges	Avg deg.	SEN Parts
Face	147	1842	25.1	22
Motorbike	297	1311	8.83	27
Airplane	226	983	8.70	10
Car	197	850	8.63	17

extents. The face and motorbike dictionaries, which have more diversity in images, are doing relatively well, while the car model has the worst distortion. This result is further confirmed by the ROC plot in Figure 5.2.

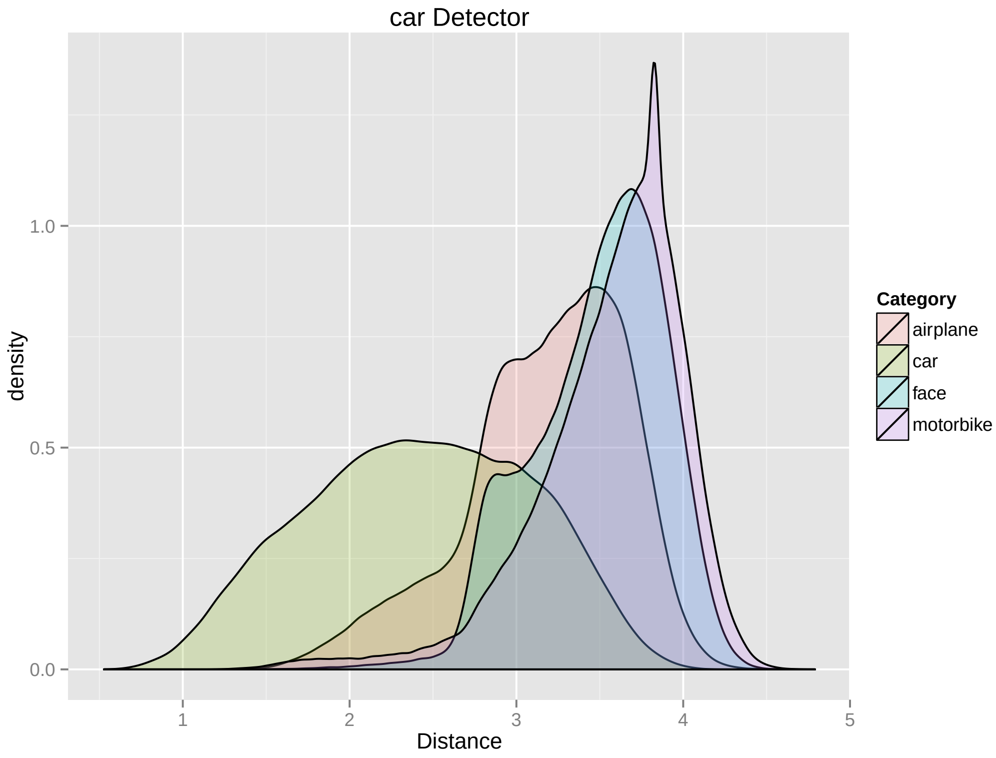
After that, we apply the same process described in preceding session, and get models with statistics shown in Table 5.4. Now, given an image from face category, the patches are beyond the coverage of the motorbike model's vocabulary. Without these face-related nodes acting as "sinks" to have patches absorbed, the viewlet assignments are vastly distorted, and foreground nodes and background ones have similar chances to be assigned to. As a result, if the model are still able to discriminate faces from motorbike, the power can only be from the structural and spatial information. The results are shown in Table 5.5, alongside those from the similar experiment in [FPZ03]. We can see that our SUV model outperforms that of Fergus's [FPZ03] with a large margin.



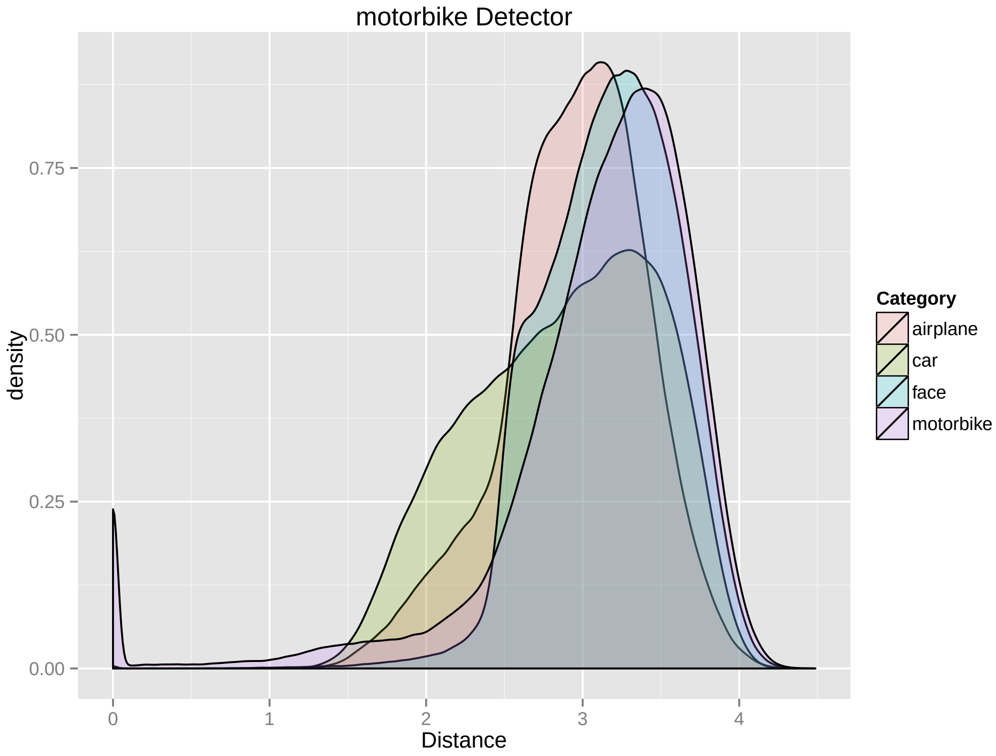
(a) Shared Model



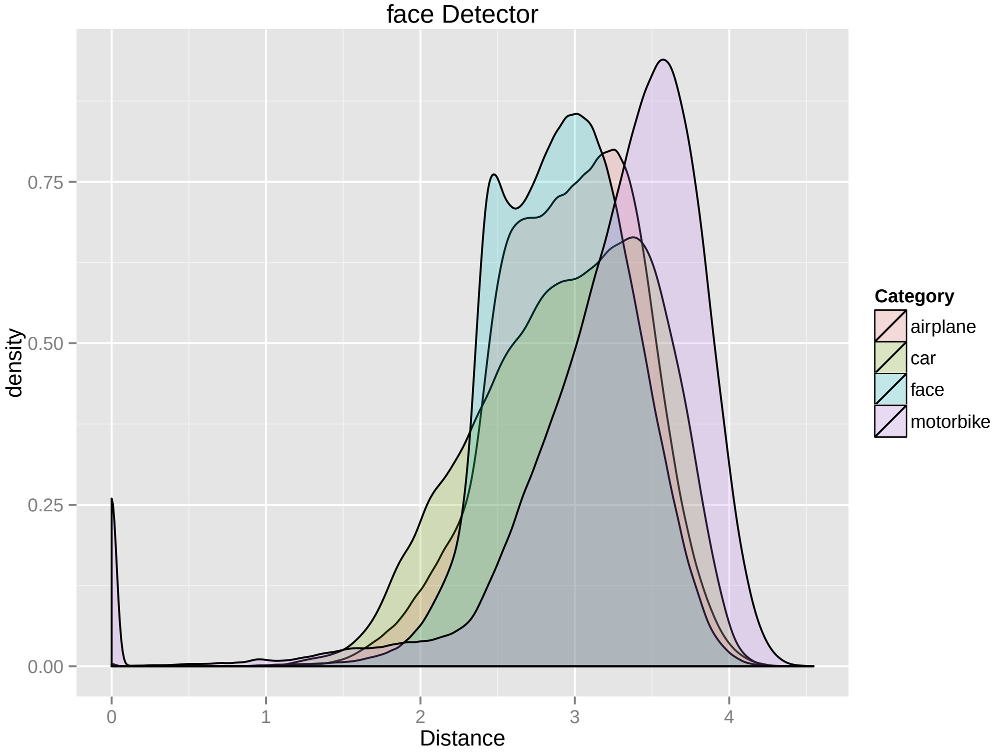
(b) Airplane Model



(c) Car Model

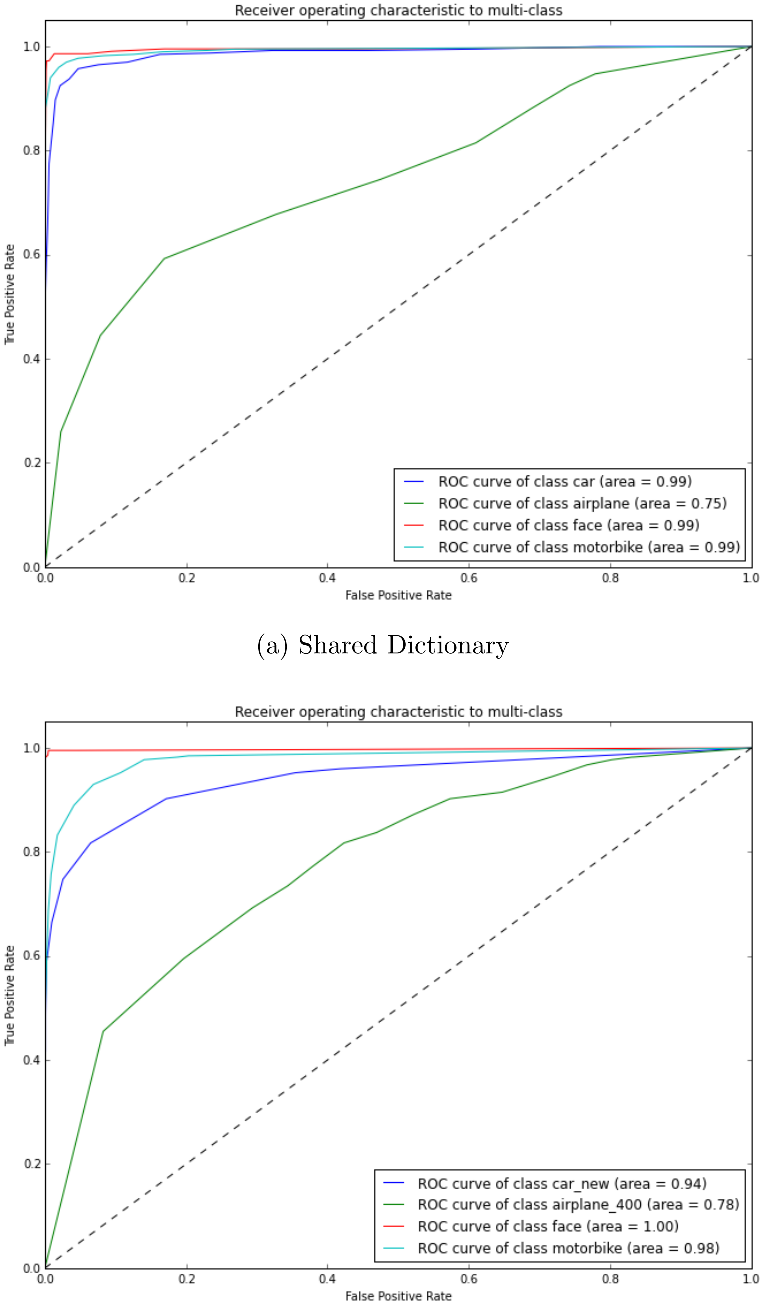


(d) Motorbike Model



(e) Face Model

Figure 5.1: Distribution of distance to the nearest neighbor (measures the completeness of dictionary of each model, from which we can see the dictionary obtained from face and motorbike training set have the least distortion, whereas the dictionary from car images is the worst, all features from other categories have a further distance than these from car category.)



(a) Shared Dictionary

(b) Separated Dictionary

Figure 5.2: The Receiver operating characteristic curve for individual model performance using both shared dictionary and separated dictionaries, we can see that from using a feature space complete dictionary to a limited one, the performance car model got the worse impact, which is in accord with what we observed in Figure 5.1 (The airplane detectors have generally the worse performance because of different airplane orientations in images, which was corrected by manual image flipping in [FPZ03]).

Table 5.5: Confusion table for separated dictionary model on the Caltech-101 dataset

	Our Model				Fergus et al. [FPZ03]			
Query Image	F	M	A	C	F	M	A	C
Face	0.98	0.069	0.215	0.252	0.964	0.33	0.32	-
Motorbike	0	0.95	0.370	0.237	0.50	0.925	0.51	-
Airplane	0	0.007	0.665	0.025	0.63	0.64	0.902	-
Car	0	0	0.002	0.600	-	-	-	-

5.3 Summary

In this chapter we have presented the experiment of our SUV model on multiple category images, with a more complete implementation. We further automated the approach by improve the detection process with network-based grouping. The experiments with shared viewlet vocabulary show that our model, though trained in an unsupervised, "perceptual" way, provides similar performance to that of state-of-the-art approaches in classification. The experiments with separated dictionary show that even when the feature vocabulary (knowledge) module in our agent is limited, the algorithm is still able to extract and abstract structural information from positive images and form a prototype of the corresponding object, which, in our opinion, is similar to how an infant learns novel object without any linguistic conceptual ability [LS96].

CHAPTER 6

Concluding Remarks and Future Work

This thesis proposed the **Structural Unsupervised Viewlets (SUV)** Model, that makes four main contributions, as follows,

A Full Bayesian Model In a Network Form: We provided an elegant representation of general deformable part-based objects, which originated from a Markov Random Field setting. With the sparsity assumption that the number of direct interactions grows linearly in the number of viewlets (i.e. visually similar parts and their various configurations), a bridge between the full multi-variate Gaussian model and a sparse network model was established, complete with concise mathematical formalism and fast learning algorithms. Moreover, no particular structure of the dependency network is enforced *a priori*, making our model more flexible and representative than the existing unsupervised and weakly-supervised approaches, where constraints such as a star or a tree model is imposed.

Robust, Scale and Translation Invariant, Unsupervised Approach: For global shape information modeling, instead of picking one part as a reference point to represent the absolute positions of the other parts, which is inherently prone to the landmark node detection error, our network is built in a distributed way where all spatial information is encoded as pairwise relative values. This encoding also includes the necessary information to make the approach scale and translation invariant. With evidence from earlier research and later demonstrated via experiment, it is shown that the global structure can be robustly re-constructed from our distributed pairwise encoding.

Mining Views of the Same Part: This study suggests an innovative way to mine structurally similar viewlets with geometric consistency, solely, from the topology of the learned network without using any extra information. This allows the discovery of viewlets of the same semantic part, but with dramatic visual differences. Such information is traditionally encoded via manual and supervised training methods.

Efficient Algorithms for Learning and Detection: Efficient algorithms and their implementations were proposed to learn viewlets and the interactions between them from unannotated images of various sizes and resolutions, and to detect the best embedding of objects in query images that also handles multi-object case natively.

In our future work, we would like to evolve the SUV model to incorporate more flexibility and we describe some future directions as follows:

Different Feature Sets to Represent Viewlets: As we discussed in Chapter 1, our approach so far has been aimed at showcasing the higher-level capabilities of a part-aware object representation platform, and we have not yet explored the improvements that can be had by trying different features that are used to represent the underlying viewlets. We chose to use HOG throughout the project mostly due to the various advantages it provides, including the availability of fast algorithms to calculate the feature vectors, and the robustness of the features to small spatial shifts and color/intensity variances. However, with a better feature descriptor, e.g. those afforded by a Deep Neural Network pipeline as opposed to the HOG←K-Means pipeline used in our work, we believe that the end-to-end performance can be vastly improved.

Bimodal Spatial Relation Edges: It is generally true that for most kinds of objects which aren't too complicated, the spatial interactions between any pair of viewlets can be adequately modeled by a unimodal distribution. Incorporating the flexibility of bimodal interactions will further extend the representation power of the SUV model. For example, in a large corpus, one might have both standing and sitting humans, and the viewlets corresponding to the head areas and those to the leg areas, will clearly

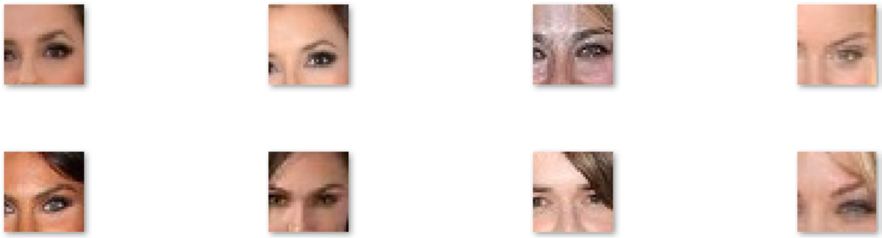


Figure 6.1: A finer viewlet for two-eyes part obtained by applying the SUV framework only to the viewlets that correspond to the head part.

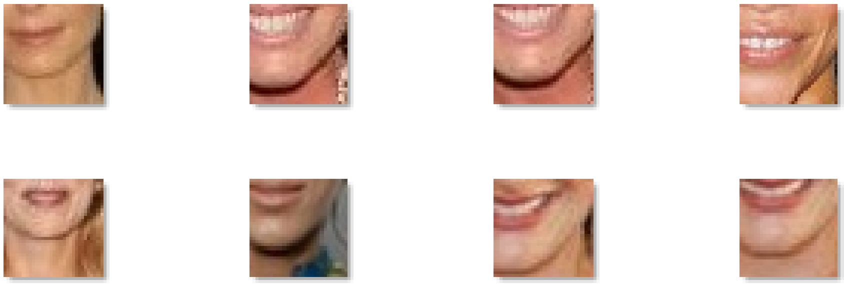


Figure 6.2: A finer viewlet for the mouth part, automatically derived by applying the SUV framework on the viewlets belonging to the head cluster.

appear in both scenarios and hence their spatial relationships will be bi-modal in distribution. While the over-fitting issue must be carefully handled when we make this extension, bimodal distributions can be handled efficiently in our framework.

Hierarchical Network Model: We would like to further improve the SUV framework, and incorporate a hierarchical structure in the network model. For example, one approach would be to recursively apply the SUV framework to the sets of semantically equivalent viewlets. We have done some preliminary tests on the head part nodes, and got finer details. As illustrated in Figure 6.1 and Figure 6.2, we have already observed that our framework is able to automatically determine the different parts of the face.

References

[AB02] Réka Albert and Albert Laszlo Barabasi. “Statistical mechanics of complex networks.” *Reviews of Modern Physics*, **74**(1):47–97, 2002.

[AGH84] D Albright, G Gross, Mental Health, New Haven, Received August, and Revised Janury. “Stimulus-selective neurons in the.” *Journal of Neuroscience*, 1984.

[AHL00] Lada a. Adamic, Bernardo a. Huberman, a. L. Barabási, R. Albert, H. Jeong, and G. Bianconi. “Power-Law Distribution of the World Wide Web.” *Science*, **287**(5461):2115, March 2000.

[AKB08] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. “CenSurE: Center surround extremas for realtime feature detection and matching.” In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5305 LNCS, pp. 102–115, 2008.

[ARK10] Itamar Arel, Derek Rose, and Thomas Karnowski. “Deep machine learning-A new frontier in artificial intelligence research.” *IEEE Computational Intelligence Magazine*, **5**(4):13–18, 2010.

[ASB00] L a Amaral, a Scala, M Barthelemy, and H E Stanley. “Classes of small-world networks.” *Proceedings of the National Academy of Sciences of the United States of America*, **97**(21):11149–11152, 2000.

[BA99] Albert-Laszlo Barabasi and Reka Albert. “Emergence of scaling in random networks.” *Science*, **286**(5439):11, October 1999.

[BAJ99] a. L. Barabási, Réka Albert, and Hawoong Jeong. “Mean-field theory for scale-free random networks.” *Physica A: Statistical Mechanics and its Applications*, **272**(1):173–187, 1999.

[BC92] Irving Biederman and Eric E. Cooper. “Size invariance in visual object priming.” *Journal of Experimental Psychology: Human Perception and Performance*, **18**(1):121–133, 1992.

[BCP04] Jason J S Barton, Mariya V. Cherkasova, Daniel Z. Press, James M. Intriligator, and Margaret O’Connor. “Perceptual functions in prosopagnosia.” *Perception*, **33**(8):939–956, 2004.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(8):1798–1828, 2013.

[Ben09] Yoshua Bengio. *Learning Deep Architectures for AI*, volume 2. 2009.

[BG93] I Biederman and P C Gerhardstein. “Recognizing depth-rotated objects: evidence and conditions for three-dimensional viewpoint invariance.” *Journal of experimental psychology. Human perception and performance*, **19**(6):1162–1182, 1993.

[BGH04] Vincent Blondel, Anahi Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. “A measure of similarity between graph vertices.” *SIAM Review*, **46**(4):647–666, January 2004.

[BGL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. “Fast unfolding of communities in large networks.” *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10):6, October 2008.

[Bie87] I Biederman. “Recognition-by-components: a theory of human image understanding.” *Psychological review*, **94**(2):115–147, 1987.

[BLM06] Stefano Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. “Complex networks: Structure and dynamics.” *Physics Reports*, **424**(4-5):175–308, February 2006.

[BM00] S. Belongie and J. Malik. “Matching with shape contexts.” *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pp. 81–105, 2000.

[BMB09] Lubomir Bourdev, Jitendra Malik, U C Berkeley, Adobe Systems, Park Ave, and San Jose. “Poselets : Body Part Detectors Trained Using 3D Human Pose Annotations.” *Computer Vision, 2009 IEEE 12th . . .*, pp. 2–9, 2009.

[BMB10] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. “Detecting people using mutually consistent poselet activations.” In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6316 LNCS, pp. 168–181, 2010.

[BMP01] S. Belongie, J. Malik, and J. Puzicha. “Shape Context: A new descriptor for shape matching and object recognition.” *NIPS*, **54**(2), 2001.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. “Shape matching and object recognition using shape contexts.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(4):509–522, April 2002.

[BNJ12] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation.” *Journal of Machine Learning Research*, **3**(4-5):993–1022, May 2012.

[BTV06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded up robust features.” In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pp. 404–417, 2006.

[BV04] Stephen SP Boyd and Lieven Vandenberghe. *Convex optimization*, volume 25. Cambridge University Press, 2004.

[BWP98] Michael C. Burl, Markus Weber, and Pietro Perona. “A probabilistic approach to object recognition using local photometry and global geometry.” In *ECCV*, volume 1407, pp. 628–641, 1998.

[CFH05] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. “Spatial priors for part-based recognition using statistical models.” *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, **1**:10–17, 2005.

[CJA14] Ryan Compton, David Jurgens, and David Allen. “Geotagging One Hundred Million Twitter Accounts with Total Variation Minimization.” *arXiv preprint arXiv:1404.7152*, 2014.

[CKL09] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan. “On compressing social networks.” *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (2009)*, **9**(1):219, 2009.

[CM07] Josephine Cock and Beat Meier. “Incidental task sequence learning: Perceptual rather than conceptual?” *Psychological Research*, **71**(2):140–151, 2007.

[CML14] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. “Detect What You Can: Detecting and Representing Objects using Holistic Models and Body Parts.” *Computer Vision and Pattern Recognition*, 2014.

[CNM04] Aaron Clauset, M. E J Newman, and Cristopher Moore. “Finding community structure in very large networks.” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **70**(6 2):066111, December 2004.

[DDA06] Leon Danon, Albert Diaz-Guilera, and Alex Arenas. “Effect of size heterogeneity on community identification in complex networks.” *Journal of Statistical Mechanics: Theory and Experiment*, **11010**:6, 2006.

[Den13] Li Deng. “Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey.” *Research.Microsoft.Com*, 2013.

[DG79] Robert Desimone and Charles G. Gross. “Visual areas in the temporal cortex of the macaque.” *Brain Research*, **178**(2-3):363–380, 1979.

[DLH13] Li Deng, Jinyu Li, Jui Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero. “Recent advances in deep learning for speech research at Microsoft.” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 8604–8608, 2013.

[DM00] S. N. Dorogovtsev and J. F. F. Mendes. “Scaling Behaviour of Developing and Decaying Networks.” *EPL (Europhysics Letters)*, p. 7, 2000.

[DT05] Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection.” In *CVPR ’05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1*, volume I, pp. 886–893, 2005.

[DTS06] Navneet Dalal, Bill Triggs, and Cordelia Schmid. “Human detection using oriented histograms of flow and appearance.” In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3952 LNCS, pp. 428–441, 2006.

[Eic04] Howard Eichenbaum. “Hippocampus: Cognitive processes and neural representations that underlie declarative memory.” *Neuron*, **44**(1):109–120, 2004.

[EQ94] P D Eimas and P C Quinn. “Studies on the formation of perceptually based basic-level categories in young infants.” *Child development*, **65**(3):903–917, 1994.

[ER59] P Erdös and A Rényi. “On random graphs.” *Publicationes Mathematicae*, **6**:290–297, 1959.

[FB71] J J Franks and J D Bransford. “Abstraction of visual patterns.” *Journal of experimental psychology*, **90**(1):65–74, 1971.

[FE73] M.a. Fischler and R.a. Elschlager. “The Representation and Matching of Pictorial Structures.” *IEEE Transactions on Computers*, **C-22**(1):67–92, 1973.

[FFP03] Li Fei-Fei, Robert Fergus, and Pietro Perona. “A Bayesian approach to unsupervised one-shot learning of object categories.” *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003.

[FFP06] Li Fei-Fei, Robert Fergus, and Pietro Perona. “One-shot learning of object categories.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(4):594–611, 2006.

[FFP10] Robert Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. “Learning object categories from internet image searches.” In *Proceedings of the IEEE*, volume 98, pp. 1453–1466, 2010.

[FGM09] Pedro F Felzenszwalb, Ross B Girshick, David Mcallester, and Deva Ramanan. “Object Detection with Discriminatively Trained Part Based Models.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9):1–20, September 2009.

[FGM10] Pedro F. Felzenszwalb, Ross B. Girshick, and David McAllester. “Cascade object detection with deformable part models.” In *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, pp. 2241–2248. Ieee, June 2010.

[FH05] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Pictorial structures for object recognition." *International Journal of Computer Vision*, **61**(1):55–79, January 2005.

[FM06] Pedro Felzenszwalb and David McAllester. "A min-cover approach for finding salient curves." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2006, pp. 185–185. Ieee, 2006.

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1–8. Ieee, June 2008.

[For10] Santo Fortunato. "Community detection in graphs." *Physics Reports*, **486**(3-5):75–174, February 2010.

[FP05] Li Fei-Fei and Pietro Perona. "A Bayesian hierarchical model for learning natural scene categories." *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **2**:524–531, 2005.

[FPZ03] Robert Fergus, Pietro Perona, and Andrew Zisserman. "Object class recognition by unsupervised scale-invariant learning." *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, **2**, 2003.

[FPZ04] Robert Fergus, Pietro Perona, and Andrew Zisserman. "A Visual Category Filter for Google Images." In *ECCV '04 8th European Conference on Computer Vision*, volume 3021, pp. 242 – 256, 2004.

[FPZ05] Robert Fergus, Pietro Perona, and Andrew Zisserman. "A sparse object category model for efficient learning and exhaustive recognition." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**:380–387, 2005.

[FPZ07] Robert Fergus, Pietro Perona, and Andrew Zisserman. "Weakly supervised scale-invariant learning of models for visual recognition." *International Journal of Computer Vision*, **71**(3):273–303, July 2007.

[FZP04] Robert Fergus, Andrew Zisserman, and Pietro Perona. "Sampling methods for unsupervised learning." *Advances in Neural Information Processing Systems 17*, **17**(Mcmc):433–440, 2004.

[GCC11] Bernd Girod, Vijay Chandrasekhar, David M. Chen, Ngai Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S. Tsai, and Ramakrishna Vedantham. "Mobile visual search." *IEEE Signal Processing Magazine*, **28**(4):61–76, 2011.

[GD06] Kristen Grauman and Trevor Darrell. “Unsupervised learning of categories from sets of partially matching image features.” In *CVPR*, volume 1, pp. 19–25, 2006.

[GFM11] Ross B Girshick, Pedro F. Felzenszwalb, and David Mcallester. “Object detection with grammar models.” *Advances in Neural*, pp. 1–9, 2011.

[GGC08] Thomas Grüter, Martina Grüter, and Claus-Christian Carbon. “Neural and genetic foundations of face recognition and prosopagnosia.” *Journal of neuropsychology*, **2**(Pt 1):79–97, 2008.

[GGM14] Georgia Gkioxari, Ross B Girshick, and Jitendra Malik. “Actions and Attributes from Wholes and Parts.” *CoRR*, **abs/1412.2**, 2014.

[GHG14] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, Jitendra Malik, and Berkeley Berkeley. “Using k-poselets for detecting people and localizing their keypoints.” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3582–3589, 2014.

[GM04] Kalanit Grill-Spector and Rafael Malach. “The human visual cortex.” *Annual review of neuroscience*, **27**:649–677, 2004.

[GN02] M Girvan and M E J Newman. “Community structure in social and biological networks.” *Proceedings of the National Academy of Sciences of the United States of America*, **99**(12):7821–7826, 2002.

[Gro02] Charles G Gross. “Genealogy of the ”grandmother cell”.” *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*, **8**(5):512–518, 2002.

[Ham93] D. Hammerstrom. “Neural networks at work.” *IEEE Spectrum*, **30**(6), 1993.

[HE73] Donald Homa and Et Al. “Prototype abstraction and classification of new instances as a function of number of instances defining the prototype.” *Journal of Experimental Psychology*, **101**(1):116–122, 1973.

[Hin07] Geoffrey E. Hinton. “Learning multiple layers of representation.” *Trends in Cognitive Sciences*, **11**(10):428–434, 2007.

[Hof99] Thomas Hofmann. “Probabilistic Latent Semantic Analysis.” *Uncertainity in Artifitial Intelligence - UAI’99*, p. 8, 1999.

[HWS85] E Halgren, C L Wilson, and J M Stapleton. “Human medial temporal-lobe stimulation disrupts both formation and retrieval of recent memories.” *Brain and cognition*, **4**(3):287–295, 1985.

[JNB01] H. Jeong, Z. Neda, and a. L. Barabasi. “Measuring preferential attachment for evolving networks.” *Europhysics Letters (EPL)*, **61**(4):4, February 2001.

[KFH08] Gunhee Kim, Christos Faloutsos, and Martial Hebert. "Unsupervised modeling of object categories using link analysis techniques." In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1–8. Ieee, June 2008.

[KHE12] Hongwen Kang, Martial Hebert, Alexei a. Efros, and Takeo Kanade. "Connecting missing links: Object discovery from sparse observations using 5 million product images." In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7577 LNCS, pp. 794–807, 2012.

[KHK11] Hongwen Kang, Martial Hebert, and Takeo Kanade. "Discovering object instances from scenes of Daily Living." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 762–769. Ieee, November 2011.

[KR08] Joseph S. Kong and Vwani P. Roychowdhury. "Preferential survival in models of complex ad hoc networks." *Physica A: Statistical Mechanics and its Applications*, **387**(13):3335–3347, February 2008.

[Kre07] Gabriel Kreiman. "Single unit approaches to human vision and memory." *Current Opinion in Neurobiology*, **17**(4):471–475, 2007.

[Kru64] J. B. Kruskal. "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis." *Psychometrika*, **29**(1):1–27, 1964.

[KSR08] Joseph S Kong, Nima Sarshar, and Vwani P Roychowdhury. "Experience versus talent shapes the structure of the Web." *Proceedings of the National Academy of Sciences of the United States of America*, **105**(37):13724–13729, September 2008.

[LAM09] Hesheng Liu, Yigal Agam, Joseph R. Madsen, and Gabriel Kreiman. "Timing, Timing, Timing: Fast Decoding of Object Information from Intracranial Field Potentials in Human Visual Cortex." *Neuron*, **62**(2):281–290, 2009.

[LBD89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition.", 1989.

[LH05] Marius Leordeanu and Martial Hebert. "A spectral technique for correspondence problems using pairwise constraints." *Proceedings of the IEEE International Conference on Computer Vision*, **II**:1482–1489, 2005.

[LLM10] J Leskovec, K J Lang, and M Mahoney. "Empirical comparison of algorithms for network community detection." *Conference on World Wide Web* {*WWW*}, pp. 631–640, 2010.

[LM02] R. Lienhart and J. Maydt. "An extended set of Haar-like features for rapid object detection." *Proceedings. International Conference on Image Processing*, **1**:900–903, 2002.

[Low99] D.G. Lowe. "Object recognition from local scale-invariant features." *Proceedings of the Seventh IEEE International Conference on Computer Vision*, **2**:1150–1157 vol.2, 1999.

[LS96] N K Logothetis and D L Sheinberg. "Visual object recognition." *Annual review of neuroscience*, **19**:577–621, 1996.

[Man01] Reference Manual. "Open Source Computer Vision Library." \url{https://github.com/itseez/opencv}, 2001.

[Mar95] C J Marsolek. "Abstract visual-form representations in the left cerebral hemisphere." *Journal of experimental psychology. Human perception and performance*, **21**(2):375–386, 1995.

[MK80] A. Mikami and K. Kubota. "Inferotemporal neuron activities and color discrimination with delay." *Brain Research*, **182**(1):65–78, 1980.

[MM09] S. Maji and J. Malik. "Object detection using a max-margin Hough transform." *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1038–1045, 2009.

[New00] M. E. J. Newman. "Who is the best connected scientist? A study of scientific coauthorship networks." **64**:17, 2000.

[New01] M. Newman. "Scientific collaboration networks.I. Network construction and fundamental results." *Physical Review E*, **64**(1):1–8, 2001.

[New03] M. E. J. Newman. "The structure and function of complex networks." *Dialogues in clinical neuroscience*, **45**:167–256, 2003.

[New04] M. E J Newman. "Fast algorithm for detecting community structure in networks." *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **69**(6 2):066133, June 2004.

[New06] M E J Newman. "Modularity and community structure in networks." *Proceedings of the National Academy of Sciences of the United States of America*, **103**(23):8577–8582, June 2006.

[NG04] M. E J Newman and M. Girvan. "Finding and evaluating community structure in networks." *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **69**(2 2):26113, February 2004.

[OP94] Mike W. Oram and David I. Perrett. "Modeling visual recognition from neurobiological constraints." *Neural Networks*, **7**(6-7):945–972, 1994.

[PDF05] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. "Uncovering the overlapping community structure of complex networks in nature and society." *Nature*, **435**(7043):814–818, June 2005.

[PK68] M I Posner and S W Keele. "On the genesis of abstract ideas." *Journal of experimental psychology*, **77**(3):353–363, 1968.

[PRC81] S. Palmer, E. Rosch, and P. Chase. "Canonical perspective and the perception of objects." In J. Long and A. Baddeley, editors, *Attention and Performance IX*, pp. 145–151. Lawrence Erlbaum Associates, 1981.

[PSP85] D I Perrett, P a Smith, D D Potter, a J Mistlin, a S Head, a D Milner, and M a Jeeves. "Visual cells in the temporal cortex sensitive to face view and gaze direction." *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, **223**(1232):293–317, 1985.

[QRK05] R Quian Quiroga, L Reddy, G Kreiman, C Koch, and I Fried. "Invariant visual representation by single neurons in the human brain." *Nature*, **435**(7045):1102–1107, 2005.

[grandmother neuro.]

[RAA15] Federica B. Rosselli, Alireza Alemi, Alessio Ansuini, and Davide Zoccolan. "Object similarity affects the perceptual strategy underlying invariant visual object recognition in rats." *Frontiers in Neural Circuits*, **9**(March):1–22, 2015.

[RB03] Erzsébet Ravasz and Albert-László Barabási. "Hierarchical organization in complex networks." *Physical review. E, Statistical, nonlinear, and soft matter physics*, **67**(2 Pt 2):026112, 2003.

[RES06] Bryan C. Russell, Alexei a. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. "Using multiple segmentations to discover objects and their extent in image collections." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 1605–1612. Ieee, 2006.

[Ric27] Stuart A Rice. "The identification of blocs in small political bodies." *The American Political Science Review*, **21**(3):619–627, 1927.

[RP99] Maximilian Riesenhuber and T Poggio. "Hierarchical models of object recognition in cortex." **2**(11):1019–1025, 1999.

[SC08] Philipp Schuetz and Amedeo Caflisch. "Multistep greedy algorithm identifies community structure in real-world and computer-generated networks." *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **78**(2):1–7, 2008.

[Sch14] J Schmidhuber. "Deep Learning in Neural Networks: An Overview." *arXiv preprint arXiv:1404.7828*, pp. 1–66, 2014.

[SKK07] Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. "A quantitative theory of immediate visual recognition." *Progress in Brain Research*, **165**:33–56, 2007.

[SL90] G. L. Scott and H. C. Longuet-Higgins. “Feature grouping by ’relocalisation’ of eigenvectors of the proximity matrix.” *Procedings of the British Machine Vision Conference 1990*, pp. 20.1–20.6, 1990.

[SR04] Nima Sarshar and Vwani Roychowdhury. “Scale-free and stable structures in complex ad hoc networks.” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **69**(2 2):1–6, February 2004.

[SR05] Nima Sarshar and Vwani Roychowdhury. “Multiple power-law structures in heterogeneous complex networks.” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, **72**(2):1–11, August 2005.

[TLB10] Tinne Tuytelaars, Christoph H. Lampert, Matthew B. Blaschko, and Wray Buntine. “Unsupervised object discovery: A comparison.” *International Journal of Computer Vision*, **88**(2):284–302, July 2010.

[TLM10] Michele Tumminello, Fabrizio Lillo, and Rosario N. Mantegna. “Correlation, hierarchies, and networks in financial markets.” *Journal of Economic Behavior and Organization*, **75**(1):40–58, 2010.

[TP13] Discriminatively Trained and Deformable Part. “Discriminatively Trained Deformable Part Models (Release 5) Discriminatively trained deformable part models Discriminatively Trained Deformable Part Models (Release 5).” http://people.cs.uchicago.edu/˜rbg/latent-release5/, 2013.

[TWH03] Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. “Email as Spectroscopy: Automated Discovery of Community Structure within Organizations.” *The Information Society*, **21**(2):143–153, April 2003.

[VJ01] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features.” *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, **1**:I–511–I–518, 2001.

[WAH92] J. Weng, N. Ahuja, and T.S. Huang. “Cresceptron: a self-organizing neural network which grows adaptively.” *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, **1**:576–581, 1992.

[WAH93] J.J. Weng, N. Ahuja, and T.S. Huang. “Learning recognition and segmentation of 3-D objects from 2-D images.” *1993 (4th) International Conference on Computer Vision*, 1993.

[WAH97] John (Juyang) Weng, Narendra Ahuja, and Thomas S Huang. “Learning Recognition and Segmentation Using the Cresceptron.” *Int. J. Comput. Vision*, **25**(2):109–143, 1997.

[WG07] Xaiogang Wang and Eric Grimson. “Spatial Latent Dirichlet Allocation.” In *NIPS*, pp. 1–8, 2007.

[WKQ06] Stephen Waydo, Alexander Kraskov, Rodrigo Quian Quiroga, Itzhak Fried, and Christof Koch. "Sparse representation in the human medial temporal lobe." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, **26**(40):10232–10234, 2006.

[WOP94] E Wachsmuth, M W Oram, and D I Perrett. "Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque." *Cerebral cortex (New York, N.Y. : 1991)*, **4**(5):509–522, 1994.

[WS98] D J Watts and S H Strogatz. "Collective dynamics of 'small-world' networks." *Nature*, **393**(6684):440–442, June 1998.

[WT07] Ken Wakita and Toshiyuki Tsurumi. "Finding Community Structure in Megascale Social Networks." p. 9, 2007.

[WWP00] M. Weber, M. Welling, and P. Perona. "Towards automatic discovery of object categories." In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, pp. 101–108, 2000.

[YAK13] Yuto Yamaguchi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. "Landmark-based user location inference in social media." *Proceedings of the first ACM conference on Online social networks - COSN '13*, pp. 223–234, 2013.

[YSN11] Bin Yang, Issei Sato, and Hiroshi Nakagawa. "Secure clustering in private networks." *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 894–903, December 2011.

[ZDG14] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. "Part-based R-CNNs for Fine-grained Category Detection." *European Conference on Computer vision (ECCV)*, 2014.

[ZFD12] Ning Zhang, Ryan Farrell, and Trever Darrell. "Pose Pooling Kernels for Subcategory Recognition." (c), 2012.

[ZPT15] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. "Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues." *Computer Vision and Pattern Recognition (CVPR)*, 2015.