67

Object Recognition with Latent Conditional Random Fields

by

Ariadna Quattoni

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[September 2005]

August 2005

Signature Redacted

Author ....
Department of Electrical Engineering and Computer Science
August 22, 2005

Signature Redacted

Certified by ....
Michael Collins
Associate Professor
Thesis Supervisor

Signature Redacted

Certified by ....
Trevor Darrell
Associate Professor
Thesis Supervisor

Signature Redacted

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Object Recognition with Latent Conditional Random Fields

by

Ariadna Quattoni

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

In this thesis we present a discriminative part-based approach for the recognition of object classes from unsegmented cluttered scenes. Objects are modelled as flexible constellations of parts conditioned on local observations. For each object class the probability of a given assignment of parts to local features is modelled by a Conditional Random Field (CRF). We propose an extension of the CRF framework that incorporates hidden variables and combines class conditional CRFs into a unified framework for part-based object recognition. The random field captures spatial coherence between region labels. The parameters of the CRF are estimated in a maximum likelihood framework and recognition proceeds by finding the most likely class under our model. The main advantage of the proposed CRF framework is that it allows us to relax the assumption of conditional independence of the observed data (i.e. local features) often used in generative approaches, an assumption that might be too restrictive for a considerable number of object classes.

In the second part of this work we extend the detection model and develop a discriminative recognition system which both detects the presence of objects and finds their regions of support in an image. Our part based model allows joint object detection and region labelling; in contrast to previous methods ours can be trained with a combination of examples for which we have labelled support regions and examples for which we only know whether the object is present in the image. We extend the detection model by incorporating a segmentation variable; the segmentation variable is assumed to be observed in the fully labelled data and hidden on the partially labelled one. Our latent variable model learns sets of part labels for each image site, which allows us to merge part-based detection with part-based region labelling (or segmentation).

Thesis Supervisor: Michael Collins
Title: Associate Professor

Thesis Supervisor: Trevor Darrell
Title: Associate Professor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Chapter Overview

In this chapter we describe the main motivation behind the development of our discriminative latent model for object detection and segmentation.

1.2 Motivation: A discriminative latent variable model

Object class recognition is one of the most important problems in computer vision and has received significant attention in the last few years. Due to the large inner class variation in most visual object categories, developing a general model is a challenging task and has become one of the main focuses of high-level vision research.

Most current object recognition approaches use local feature representations (where images are represented as sets of features) and can be roughly divided into two separate groups, according to the main paradigm followed : generative (Weber 2000, Fergus 2003, Lowe 1999) and discriminative (Torralba 2004, Kumar 2003, 2004, Opelt submission, Viola 2001).

One of the main advantages of the best performing generative models is that they can handle missing data (i.e., the correspondence between local features and parts in the model) in a principled manner. For this thesis we develop a discriminative approach for object recognition that can handle missing correspondences by incorporating latent variables into the model.

In the first part of this work we present a latent discriminative model for object detection, which decides weather an object of a given class is present in the image or not. The limitation of this model is that while it performs well for detection it does not necessarily provide an accurate segmentation of the image in terms of foreground/background. We address this problem in the second part (Chapter 5) by extending the detection model to perform both detection and foreground/background segmentation.

1.2.1 Detection Model

We define the detection problem in the following manner: given a training set of n pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the ith image and y_i is the category of the object present in $\mathbf{x}_i$, we would like to learn a model that maps images to object categories. We are interested in learning to recognize rigid objects such as cars, motorbikes, and faces from one or more fixed view-points.

The part-based models that we consider represent images as sets of *patches*, or local features. These features can be high entropy patches which are detected by an interest operator such as that described in (Lowe 1999) or segments (i.e.,"blobs") obtained with a bottom up segmentation such as that described in (Sharon 2000).

Thus an image $\mathbf{x}_i$ can be considered to be a vector $\{x_{i,1}, \ldots, x_{i,m}\}$ of m patches or image regions. Each patch $x_{i,j}$ has a feature-vector representation $\phi(x_{i,j}) \in \Re^d$; the feature vector might capture various features of the appearance of a patch, as well as features of its relative location and scale.

This scenario presents an interesting challenge to conventional classification approaches in machine learning, as the input space $\mathbf{x}_i$ is naturally represented as a set of feature-vectors $\{\phi(x_{i,1}), \ldots, \phi(x_{i,m})\}$ rather than as a single feature vector. Moreover, the patches underlying the local feature vectors may have complex interdependencies: for example, they may correspond to different parts of an object, whose spatial arrangement is important to the classification task.

The most popular approach for part-based object recognition is the generative model proposed in (Fergus 2003). This classification system models the appearance, spatial relations and co-occurrence of local parts. However, one of the limitations of this framework is that to make the model computationally tractable one has to assume the independence of the observed data (i.e., local features) given their assignment to parts in the model.

This assumption might be too restrictive for a considerable number of object classes made of structured patterns. For example when modelling the building object category one would expect the data from each local feature or image site to be dependent on its neighbors because edges at adjoining sites follow some underlying structured pattern. The same is true for other object classes made of structured patterns. Some work has been done in the past to model the dependencies in the observations (Cheng 2001, Wilson 2004). However, modelling such dependencies in a generative framework is a complex problem and most of the techniques proposed make simplifying assumptions to get some sort of factored approximation of the likelihood.

A second limitation of generative approaches is that they require a model $P(x_{i,j}|h_{i,j})$ of patches $x_{i,j}$ given underlying variables $h_{i,j}$ (e.g., $h_{i,j}$ may be a hidden variable in the model, or may simply be y_i). Accurately specifying such a generative model may be challenging – in particular in cases where patches overlap one another, or where we wish to allow a hidden variable $h_{i,j}$ to depend on several surrounding patches.

A preferable approach may be to use a feature-vector representation of patches, and to use a discriminative learning framework, while at the same time having a latent variable that allows us to model the hidden correspondences between local image features and parts in the model. In this thesis we follow an approach of this type.

Similar observations concerning the limitations of generative models have been made in previous work which has led to research on discriminative models for sequence labelling such as MEMM's (McCallum 2000) and conditional random fields (CRFs)(Lafferty 2001). For example, in vision, CRF's have been applied to the task of detecting man made structures in natural images and have been shown to outperform Markov Random Fields ("MRFs") (Kumar 2003).

A strong argument for these models as opposed to MRFs concerns their flexibility in terms of representation, in that they can incorporate essentially arbitrary feature-vector representations $\phi(x_{i,j})$ of the observed data points.

In this thesis we follow such an approach and propose a new model for object recognition based on Conditional Random Fields. We model the conditional distribution $p(y|\mathbf{x})$ directly. A key difference of our approach from previous work on CRFs is that we make use of hidden variables to model missing correspondences.

In previous work on CRFs (Lafferty 2001) each "label" y_i is a sequence $\mathbf{h}_i = \{h_{i,1}, h_{i,2}, \ldots, h_{i,m}\}$ of labels $h_{i,j}$ for each observation $x_{i,j}$. The label sequences are typically taken to be fully observed on training examples. In our case the labels y_i are unstructured labels from some fixed set of object categories, and the relationship between y_i and each observation $x_{i,j}$ is not clearly defined. Instead, we model intermediate part-labels $h_{i,j}$ as hidden variables in the model.

The model defines conditional probabilities $P(y, \mathbf{h} \mid \mathbf{x})$, and hence indirectly $P(y \mid \mathbf{x}) = \sum_{\mathbf{h}} P(y, \mathbf{h} \mid \mathbf{x})$, using a CRF. Dependencies between the hidden variables $\mathbf{h}$ are modelled by an undirected graph over these variables. The result is a model where inference and parameter estimation can be carried out using standard graphical model algorithms such as belief propagation (Pearl 1988).

1.2.2 Extending the model to joint detection and segmentation

Ideally, a general purpose object recognition system would be able to perform not only detection (i.e., detecting the presence of the object in the image) but also segmentation (i.e., determining which regions in the image correspond to the object and which to the background). The reason for this is that while for some applications like image retrieval detection might be sufficient (since the user is only interested in images that contain an object of a certain class) for others, like video surveillance, segmentation might be required. We present a model that can perform joint detection and foreground/background segmentation in Chapter 5.

Detection is often an easier task than segmentation because a few discriminative features can be enough to determine the presence of the object. Also, in many natural settings the background around the object can be correlated with the class label of the image (i.e., cars tend to appear in roads and desks tend to appear in offices). In this case an object recognition system should be able to exploit discriminative contextual information as long as it is constrained to generate an accurate segmentation.

Object recognition systems that perform both segmentation and detection are normally trained using fully labelled data (i.e., data for which we know which regions in the image correspond to the object and which to the background), but obtaining fully labelled data can be expensive. Thus it would be preferable to have a model that is able to learn from a combination of a small number of fully labelled and a large number of partially labelled examples (examples where we only know that the object is present in the image). In the second part of this thesis we present a discriminative part based model for joint object detection and segmentation that can be trained with such a combination of fully-labelled and partially-labelled data.

To integrate fully labelled data into our model and learn both segmentation and detection we incorporate a segmentation variable, observed in the fully labelled data and hidden when trained on partially labelled (i.e., unsegmented) data. During training for the partially labelled examples we maximize the likelihood of the correct classification and for the fully-labelled examples the joint likelihood of the correct segmentation and classification. The proposed model captures the intuitive notion that there is a dependency between the correct segmentation of the image and the category of the object present in it.

Chapter 2

Background and Related Work

2.1 Chapter Overview

Since the model proposed in this thesis can be regarded a generalization of the Conditional Random Field (CRF) approach that incorporates hidden variables we start this chapter with a brief overview on CRF's.

Section 2.3 of this chapter includes a description of one of the most popular generative part based approaches as well as related discriminative approaches.

2.2 Background: Conditional Random Fields

Some problems in vision can be stated as labelling problems in which the input to the system is a set of image features and the output or solution is a set of corresponding labels.

In vision the most common method for solving the labelling problem is the Markov Random Field approach. An MRF is a generative model that defines a joint probability distribution $P(\mathbf{y}, \mathbf{x})$ over random observations variables $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ and corresponding label variables $\mathbf{y} = [y_1, y_2, \ldots, y_m]$.

An alternative to the MRF framework is to model the conditional probability $P(y|x)$ over labels given local observations directly, since such a model does not need to enumerate all possible observations it has the advantage that it doesn't need to make unwarranted independence assumptions for tractable inference.

A conditional random field is simply a Markov Random Field globally conditioned on observation $\mathbf{x}$. Formally we define an undirected graph $G = (V, E)$ where each node $v \in V$ corresponds to an element y_i of $\mathbf{Y}$. $\mathbf{y}, \mathbf{x}$ is a conditional random field if when conditioned on $\mathbf{x}$ each random variable y_i obeys the Markov property with respect to graph G. The absence of an edge between two vertices in G implies that the random variables represented by these vertices are conditionally independent given all other random variables in the model.

2.3 Related Work: Part-Based Models for Object Recognition

2.3.1 Generative Approach

In (Fergus 2003) the authors present a framework where each object model consists of a mixture of parts. For each part there is an associated distribution over appearance scale and location. This is a generative model where for an image $\mathbf{x}$ the appearance of each local feature given its assignment to a part in the model is assumed to be independent of all the other local features. In contrast, no independence assumptions are made about the location of the features, which are modelled as a multivariate gaussian distribution.

The model is trained with partially-labelled data (i.e., data for which only the presence or absence of the object in the image is known). The correspondences between local features and parts in the model are regarded as missing data. The model handles these missing correspondences by integrating over all possible assignments of parts to local features.

One of the limitations of this approach is that since it requires to enumerate all possible hidden assignments its complexity is exponential on the number of model parts. The other limitation is that as we already mentioned the model assumes the independence of each local feature given its assignment to a model part, an assumption that might be too restrictive for certain object classes.

2.3.2 Discriminative Random Fields Approach

The part based discriminative approach that we use to model objects is more closely related to the work of Kumar and Herbert (Kumar 2003, Kumar 2004). They train their model using fully-labelled data where each image region is assigned a part label from a discrete set of object parts.

A CRF is trained and detection and segmentation are performed by finding the most likely labelling of the image under the learned model. The main difference between our approach and Kumar's is that we do not assume that the part assignment variables are fully observed and are instead regarded as latent variables in our model. Incorporating hidden variables allows use of the partially labelled data during training.

2.3.3 "Dictionary" Approaches

Another related discriminative model for object recognition is the one presented in (Yang 2000). This framework like ours builds a discriminative classifier based on a part-based feature representation. Such a representation is obtained by measuring the similarity between image patches (detected with an interest point detector) to a pre-defined dictionary of parts. The dictionary is built by extracting and clustering patches from a set of representative images of the target class. The feature representation also includes geometric relations between every pair of parts. These features are used to train a Sparse Network of Winnows (SNow Learning Architecture).

Leibe (Leibe 2003) proposes a similar approach in which the model learns a dictionary of parts based on the appearances of local patches obtained with an interest point detector. This dictionary also incorporates the relative spatial positions of the parts as features. Their approach uses a voting scheme to combine the output of different part detectors and classify unseen images.

Dork (Dork submission) proposes a re-ranking approach in which learning is performed in two stages. In the first stage a set of candidate parts are found by clustering image patches (obtained with an interest point detector) using a mixture of Gaussians. Each component in the mixture is regarded as a candidate part classifier. In the second stage the candidate part classifiers are ranked based on the mutual information between the object category label and the part classifier. Finally, the top n classifiers are selected to be used for testing. An image is classified positively if more than t parts where detected by the top n part classifiers, where t is a threshold chosen through cross validation.

In (Ullman 2001) the authors propose a fragment based approach. Similar to Dorko's work, image patches are first extracted from the training images and representative object fragments are chosen based on the mutual information between an object class and a fragment. Recognition proceeds in two stages, in the first stage fragments are detected in the image by measuring the distance between image patches and the stored object fragments. In the second stage the local information from the fragments is combined in a probabilistic manner to determine weather the object is present in the image or not.

They propose two frameworks in which local fragment information can be combined. In the first approach they combine the local information with a naive-bayes classifier that assumes that each fragment in an object is independent of the others. The second approach tries to relax this independence assumption between fragments by imposing a hierarchy between them such that the probability of a particular fragment depends on the probability of its parent fragment.

One of the main differences between the above "dictionary" approaches and ours is that we do not perform a pre-selection of discriminative parts but rather we incorporate such a step during training of the classifier. In this way we ensure that the parts learnt are optimal in a discriminative sense. We believe that the problem with the pre-selection of parts is that it might cause "error-chaining" that will result in suboptimal classifiers. In other words, if the preprocessing step does not capture discriminative parts it might be hard for the classifier to achieve good accuracy even if it is trained in a discriminative manner.

2.3.4 Boosting Approach

In (Opelt submission) Opelt proposes an approach based on boosting weak classifiers. In this approach an image is represented by a list of local features obtained with a variety of detectors and represented with different descriptors such as SIFT. The weak hypotheses of Adaboost are calculated from these features. Intuitively, a weak hypothesis classifies an example based on the presence of a feature.

Chapter 3

Latent Conditional Random Fields for Object Detection

3.1 Chapter Overview

Section 3.2 describes the general form of the model while Section 3.3 gives details about the specific features used for the experiments in Chapters 4 and 5. Finally, Section 3.4 shows how the model can be efficiently trained using belief propagation.

3.2 General Form of the Model

Our task is to learn a mapping from images $\mathbf{x}$ to labels y. Each y is a member of a set $\mathcal{Y}$ of possible image labels, for example, $\mathcal{Y} = \{\texttt{background}, \texttt{car}\}$. We take each image $\mathbf{x}$ to be a vector of m "patches" $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$.[1] Each patch x_j is represented by a feature vector $\phi(x_j) \in \Re^d$. For example, in our experiments each x_j corresponds to a patch that is detected by the feature detector in (Lowe 1999). Chapter 4 gives details of the feature-vector representation $\phi(x_j)$ for each patch. Our training set consists of labelled images $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots n$, where each $y_i \in \mathcal{Y}$, and each $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$. For any image $\mathbf{x}$ we also assume a vector of

[1] Note that the number of patches m can vary across images, and did vary in our experiments. For convenience we use notation where m is fixed across different images; in reality it will vary across images but this leads to minor changes to the model.

"parts" variables $\mathbf{h} = \{h_1, h_2, \ldots, h_m\}$. These variables are not observed on training examples, and will therefore form a set of hidden variables in the model. Each h_j is a member of $\mathcal{H}$ where $\mathcal{H}$ is a finite set of possible parts in the model. Intuitively, each h_j corresponds to a labelling of x_j with some member of $\mathcal{H}$. Given these definitions of image-labels y, images $\mathbf{x}$, and part-labels $\mathbf{h}$, we will define a conditional probabilistic model:

$$P(y, \mathbf{h} \mid \mathbf{x}, \theta) = \frac{e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y',\mathbf{h}} e^{\Psi(y',\mathbf{h},\mathbf{x};\theta)}}. \tag{3.1}$$

Here θ are the parameters of the model, and $\Psi(y, \mathbf{h}, \mathbf{x}; \theta) \in \Re$ is a potential function parameterized by θ. We will discuss the choice of Ψ shortly. It follows that

$$P(y \mid \mathbf{x}, \theta) = \sum_{\mathbf{h}} P(y, \mathbf{h} \mid \mathbf{x}, \theta) = \frac{\sum_{\mathbf{h}} e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y',\mathbf{h}} e^{\Psi(y',\mathbf{h},\mathbf{x};\theta)}}. \tag{3.2}$$

Given a new test image $\mathbf{x}$, and parameter values θ^* induced from a training example, we will take the label for the image to be $\arg\max_{y \in \mathcal{Y}} P(y \mid \mathbf{x}, \theta^*)$. Following previous work on CRFs (Kumar 2003, Lafferty 2001), we use the following objective function in training the parameters:

$$L(\theta) = \sum_i \log P(y_i \mid \mathbf{x}_i, \theta) - \frac{1}{2\sigma^2}||\theta||^2 \tag{3.3}$$

The first term in Eq. 3.3 is the log-likelihood of the data. The second term is the log of a Gaussian prior with variance σ^2, i.e., $P(\theta) \sim \exp\left(\frac{1}{2\sigma^2}||\theta||^2\right)$. We will use gradient ascent to search for the optimal parameter values, $\theta^* = \arg\max_\theta L(\theta)$, under this criterion.

We now turn to the definition of the potential function $\Psi(y, \mathbf{h}, \mathbf{x}; \theta)$. We encode structural constraints with an undirected graph structure, where the hidden variables $\{h_1, \ldots, h_m\}$ correspond to vertices in the graph. E denotes the set of edges in the graph and $(j, k) \in E$ denotes that there is an edge in the graph between variables h_j and h_k. E can be an arbitrary graph; intuitively it should capture any domain specific knowledge that we have about the structure of $\mathbf{h}$. For example in our case it could encode spatial consistency between part labels.

We will see later that when E is a tree there exist exact methods for inference and parameter estimation in the model, for example using belief propagation. If E contains cycles then approximate methods, such as loopy belief-propagation, may be necessary for inference and parameter estimation. We define Ψ to take the following form:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_{j=1}^{m} \sum_{l} f_l^1(j, y, h_j, \mathbf{x})\theta_l^1 + \sum_{(j,k)\in E} \sum_{l} f_l^2(j, k, y, h_j, h_k, \mathbf{x})\theta_l^2 \qquad (3.4)$$

where f_l^1, f_l^2 are functions defining the features in the model, and θ_l^1, θ_l^2 are the components of θ. The f^1 features depend on single hidden variable values in the model, the f^2 features can depend on pairs of values. Note that Ψ is linear in the parameters θ, and the model in Eq 3.1 and Eq 3.4 is a log-linear model. Moreover the features respect the structure of the graph, in that no feature depends on more than two hidden variables h_j, h_k, and if a feature does depend on variables h_j and h_k there must be an edge (j, k) in the graph E.

Assuming that the edges in E form a tree, and that Ψ takes the form in Eq 3.4, then exact methods exist for inference and parameter estimation in the model. This follows because belief propagation (Pearl 1988) can be used to calculate the following quantities in $O(|E||\mathcal{Y}|)$ time:

$$\forall y \in \mathcal{Y}, \quad Z(y \mid \mathbf{x}, \theta) = \sum_{\mathbf{h}} \exp\{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)\}$$

$$\forall y \in \mathcal{Y}, j \in 1 \ldots m, a \in \mathcal{H}, \quad P(h_j = a \mid y, \mathbf{x}, \theta) = \sum_{\mathbf{h}:h_j=a} P(\mathbf{h} \mid y, \mathbf{x}, \theta)$$

$$\forall y \in \mathcal{Y}, (j, k) \in E, a, b \in \mathcal{H}, \quad P(h_j = a, h_k = b \mid y, \mathbf{x}, \theta) = \sum_{\mathbf{h}:h_j=a,h_k=b} P(\mathbf{h} \mid y, \mathbf{x}, \theta)$$

The first term $Z(y \mid \mathbf{x}, \theta)$ is a partition function defined by a summation over the $\mathbf{h}$ variables. Terms of this form can be used to calculate $P(y \mid \mathbf{x}, \theta) = Z(y \mid \mathbf{x}, \theta) / \sum_{y'} Z(y' \mid \mathbf{x}, \theta)$. Hence inference—calculation of $\arg\max P(y \mid \mathbf{x}, \theta)$— can be performed efficiently in the model. The second and third terms are marginal distributions over individual variables h_j or pairs of variables h_j, h_k corresponding to edges in the graph. Later in this chapter we show that the gradient of $L(\theta)$ can be defined in terms of these marginals, and hence can be calculated efficiently.

3.3 The Specific Form of the Model

We now turn to the specific form for the model used in the experiments in Chapters 4 and 5. We define

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_j \phi(x_j) \cdot \theta(h_j) + \sum_j \theta(y, h_j) + \sum_{(j,k) \in E} \theta(y, h_j, h_k) \tag{3.5}$$

Here $\theta(k) \in \Re^d$ for $k \in \mathcal{H}$ is a parameter vector corresponding to the k'th part label. The inner-product $\phi(x_j) \cdot \theta(h_j)$ can be interpreted as a measure of the compatibility between patch x_j and part-label h_j. Each parameter $\theta(y, k) \in \Re$ for $k \in \mathcal{H}$, $y \in \mathcal{Y}$ can be interpreted as a measure of the compatibility between part k and label y. Finally, each parameter $\theta(y, k, l) \in \Re$ for $y \in \mathcal{Y}$, and $k, l \in \mathcal{H}$ measures the compatibility between an edge with labels k and l and the label y. It is straightforward to verify that the definition in Eq 3.5 can be written in the same form as Eq 3.4. Hence belief propagation can be used for inference and parameter estimation in the model.

3.4 Learning and Inference

This section considers estimation of the parameters $\theta^* = \arg\max L(\theta)$ from a training sample, where $L(\theta)$ is defined in Eq. 3.3. We use a quasi-Newton method to optimize $L(\theta)$ (note that due to the use of hidden variables, $L(\theta)$ has multiple local minima, and our method is therefore not guaranteed to reach the globally optimal point). In this section we describe how the gradient of $L(\theta)$ can be calculated efficiently. Consider the likelihood term that is contributed by the i'th training example, defined as:

$$L_i(\theta) = \log P(y_i \mid \mathbf{x}_i, \theta) = \log \left(\frac{\sum_{\mathbf{h}} e^{\Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}}{\sum_{y', \mathbf{h}} e^{\Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}} . \right) \tag{3.6}$$

We first consider derivatives with respect to the parameters θ_l^1 corresponding to features $f_l^1(j, y, h_j, \mathbf{x})$ that depend on single hidden variables. Taking derivatives gives

$$\frac{\partial L_i(\theta)}{\partial \theta_l^1} = \sum_{\mathbf{h}} P(\mathbf{h} \mid y_i, \mathbf{x}_i, \theta) \frac{\partial \Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta_l^1} - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \frac{\partial \Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta_l^1}$$

$$= \sum_{\mathbf{h}} P(\mathbf{h} \mid y_i, \mathbf{x}_i, \theta) \sum_{j=1}^{m} f_l^1(j, y_i, h_j, \mathbf{x}_i) - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \sum_{j=1}^{m} f_l^1(j, y', h_j, \mathbf{x}_i)$$

$$= \sum_{j,a} P(h_j = a \mid y_i, \mathbf{x}_i, \theta) f_l^1(j, y_i, a, \mathbf{x}_i) - \sum_{y', j, a} P(h_j = a, y' \mid \mathbf{x}_i, \theta) f_l^1(j, y', a, \mathbf{x}_i)$$

It follows that $\frac{\partial L_i(\theta)}{\partial \theta_l^1}$ can be expressed in terms of components $P(h_j = a \mid \mathbf{x}_i, \theta)$ and $P(y \mid \mathbf{x}_i, \theta)$, which can be calculated using belief propagation, provided that the graph E forms a tree structure. A similar calculation gives

$$\begin{aligned} \frac{\partial L_i(\theta)}{\partial \theta_l^2} &= \sum_{(j,k) \in E, a, b} P(h_j = a, h_k = b \mid y_i, \mathbf{x}_i, \theta) f_l^2(j, k, y_i, a, b, \mathbf{x}_i) \\ &\quad - \sum_{y', (j,k) \in E, a, b} P(h_j = a, h_k = b, y' \mid \mathbf{x}_i, \theta) f_l^2(j, k, y', a, b, \mathbf{x}_i) \end{aligned}$$

hence $\partial L_i(\theta) / \partial \theta_l^2$ can also be expressed in terms of expressions that can be calculated using belief propagation.

Chapter 4

Detection Experiments

4.1 Chapter Overview

This chapter presents our detection experiments. Section 4.2 describes the data sets used for the experiments in this chapter and chapter 5. Section 4.3 discusses results for a model that encodes dependencies between hidden variables with a minimum spanning-tree. Finally, in Section 4.4 we explore other graph structures for encoding dependencies and show their effect on detection performance.

4.2 Data sets

For the experiments in this and next chapter we used 3 different data sets. The first data set is the Caltech-4 data set which can be obtained from the Vision group at Caltech. It contains images for 4 object classes each from a single point of view. The object classes are: Car (rear view), Face (front view), Plane (side view) and Motorbikes (side view). Each image contains a single instance of the object in diverse natural backgrounds. In addition, this data set also provides a generic background class which consists of indoor and outdoor images taken around Caltech campus. We use this data set for the 2-class (background vs. object) detection experiments described in section 4.3. We used a subset of this data set (Cars and Motorbikes) for the joint detection and segmentation experiments described in Chapter 5.

The second data set consists of images of side views of cars. Each image contains a single instance of the object in natural backgrounds and was obtained from http://l2r.cs.uiuc.edu/ cogcomp/Data/Car/. This data set was used for the 2-class detection experiments in section 4.3 as well as for the 3-class detection experiments. This data set and the car rear data set from Caltech-4 where used for the experiments in section 4.4.

The third data set (the animal data set) is a subset of the Caltech-101 data set. Each image contains a single or multiple instances of an animal from different point of views in a natural background. This data set was used for the 4-class experiments in section 4.3.

For each image we compute a set of patches which are obtained using the SIFT detector (Lowe 1999). Each patch $x_{i,j}$ is then represented by a feature vector $\phi(x_{i,j})$ that incorporates a combination of SIFT and relative location and scale features.

For the experiments in Section 4.3 the tree E is formed by running a minimum spanning tree algorithm over the parts $h_{i,j}$, where the cost of an edge in the graph between $h_{i,j}$ and $h_{i,k}$ is taken to be the distance between the centers of mass of patches $x_{i,j}$ and $x_{i,k}$ in the image. Note that the structure of E will vary across different images. Our choice of E encodes our assumption that parts conditioned on features that are spatially close are more likely to be dependent. For the experiments in section 4.4 we investigate more complex graph structures that involve cycles and demand approximate inference methods for parameter estimation and inference.

4.3 Minimum Spanning-Tree Detection Experiments

We carried out three sets of experiments on a number of different data sets. The first two experiments consisted of training a two class model (object vs. background) to distinguish between a category from a single viewpoint and background. The third experiment consisted of training a multi-class model to distinguish between n classes.

The only parameter that was adjusted in the experiments was the scale of the images upon which the interest point detector was run. In particular, we adjusted the scale on the car side data set: in this data set the images were too small and without this adjustment the detector would fail to find a significant amount of features.

For the experiments we randomly split each data set into three separate data sets: training, validation and testing. We use the validation data set to set the variance parameters σ^2 of the gaussian prior. For the first and second experiments the recognition task was a simple object present—absent one. For the third experiment the task was to determine the presence of an object and its view-point, either rear or side. The performance figures quoted for the first and second experiments are receiver-operating characteristic equal error rates, tested against the background data set. For the third experiment we report recall and precision figures for each object view-point.

4.3.1 Results

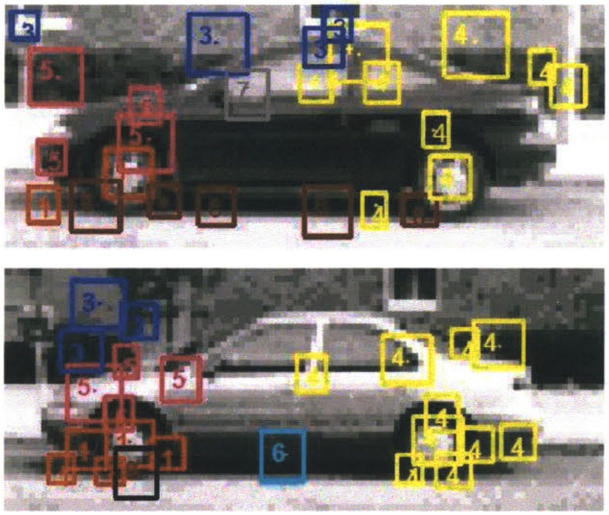


Figure 4-1: Examples of the most likely assignment of parts to features for the two class experiments (car data set).

In Table 4.1(a) we show how the number of parts in the model affects performance. In the case of the car side data set, the ten-part model shows a significant improvement

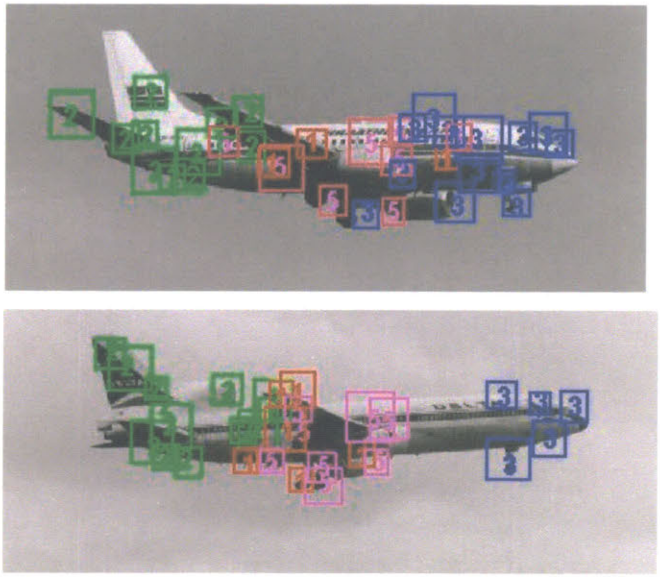


Figure 4-2: Examples of the most likely assignment of parts to features for the two class experiments (plane data set).

compared to the five parts model while for the car rear data set the performance improvement obtained by increasing the number of parts is not as significant. Table 4.1(b) shows a performance comparison with previous approaches (Fergus 2003) tested on the same data set (though on a different partition). We observe a significant improvement for most data sets.

Tables 4.2 and 4.3 show results for the multi-class experiments. Notice that random performance for the animal data set would be 25 % across the diagonal. The model exhibits best performance for the Leopard data set, for which the presence of part 1 alone is a clear predictor of the class. Table 4.2 shows results for a multi-view experiment where the task is two distinguish between two different views of a car and background.

(a)

Data set	5 parts	10 parts
Car Side	94 %	99 %
Car Rear	91 %	91.7 %

(b)

Data set	Our Model	Others [1]
Car Side	99 %	-
Car Rear	94.6 %	90.3 %
Face	99 %	96.4 %
Plane	96 %	90.2 %
Motorbike	95 %	92.5 %

Table 4.1: (a) Equal Error Rates for the car side and car rear experiments with different number of parts. (b) Comparative Equal Error Rates.

Data set	Precision	Recall
Car Side	87.5 %	98 %
Car Rear	87.4 %	86.5 %

Table 4.2: Precision and recall results for 3 class experiment.

Data set	Leopards	Llamas	Rhinos	Pigeons
Leopards	91 %	2 %	0 %	7 %
Llamas	0 %	50 %	27 %	23 %
Rhinos	0 %	40 %	46 %	14 %
Pigeons	0 %	30 %	20 %	50 %

Table 4.3: Confusion table for 4 class experiment.

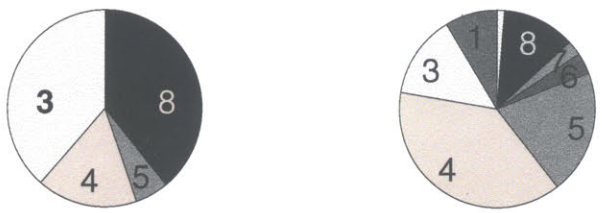
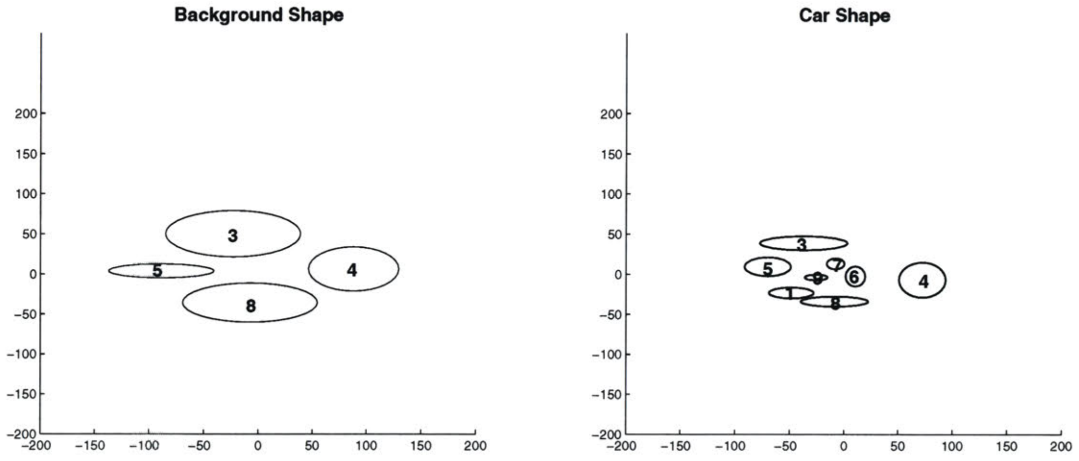


Figure 4-3: Graph showing part counts for the background (left) and car side images (right)



(a)

(b)

Figure 4-4: (a) Graph showing mean and variance of locations for the different parts for the car side images; (b) Mean and variance of part locations for the background images.

Figures 4.1 and 4.2 display the Viterbi labelling:

$$\mathbf{h}^* = \arg\max_{\mathbf{h}} P(\mathbf{h} \mid y, \mathbf{x}, \theta) \tag{4.1}$$

where $\mathbf{x}$ is an image and y is the label for the image given by our model) for a set of example images showing the most likely assignment of local features to parts in the model. Figure 4.4(a) and 4.4(b) show the mean and variance of each part's location for car side images and background images respectively. The mean and variance of each part's location for the car side images were calculated in the following manner:

First we find for every image classified as class a the most likely part assignment under our model. Second, we calculate the mean and variance of positions of all local features that were assigned to the same part. Similarly Figure 4.3 shows part counts among the Viterbi labellings assigned to examples of a given class.

As can be seen in Figures 4.4(a) and 4.4(b), while the mean location of a given part in the background images and the mean location of the same part in the car images are very similar, the parts in the car have a much tighter distribution which seems to suggest that the model is learning the shape of the object.

As shown in Figure 4.3 the model has also learnt discriminative part distributions for each class, for example the presence of part 1 seems to be a clear predictor for the car class. Some part assignments seem to rely on a combination of appearance and relative location. Part 1, for example, is assigned to wheel-like patterns located on the left of the object. The parts however, might not carry semantic meaning. It appears that the model has learnt a vocabulary of very general parts with significant variability in appearance and learns to discriminate between classes by capturing the most likely arrangement of these parts for each class.

4.4 Exploring different neighborhood structures

Our approach assumes that parts conditioned on proximate observations are likely to be dependent and the neighborhood structure of the class conditional CRFs (i.e., the associated graphs E) models the dependencies between part assignments of proximate patches.

In the previous experiments we set E to be a minimum spanning tree where the cost of an edge between two hidden variables was the 2-D distance between the center of mass of the corresponding image patches. Such dependencies between hidden part assignments can be encoded using n-neighbor lattices over local observations. However, increasing connectivity leads to an increase in the computational complexity of performing inference in such models. When E contains cycles we need to resort to approximate inference methods for inference and learning in the model.

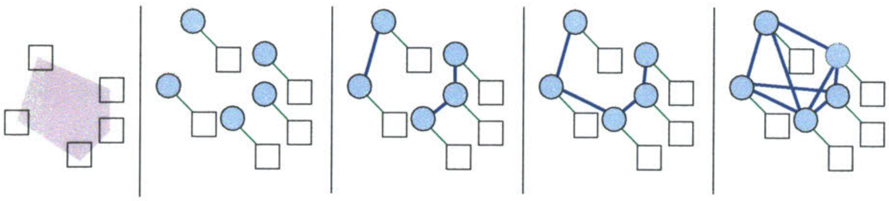


Figure 4-5: Connectivity Graphs

In this section we evaluate a range of different dependency structures and measure the effect of different amounts of connectivity on recognition performance under our model. We demonstrate the importance of encoding local dependencies by comparing the class conditional CRF model to an equivalent model that assumes the independence of hidden part assignments given the object class labels. We also compare the performance of the minimum spanning-tree and lattice models and show that there is no significant performance loss due to the minimum spanning-tree approximation.

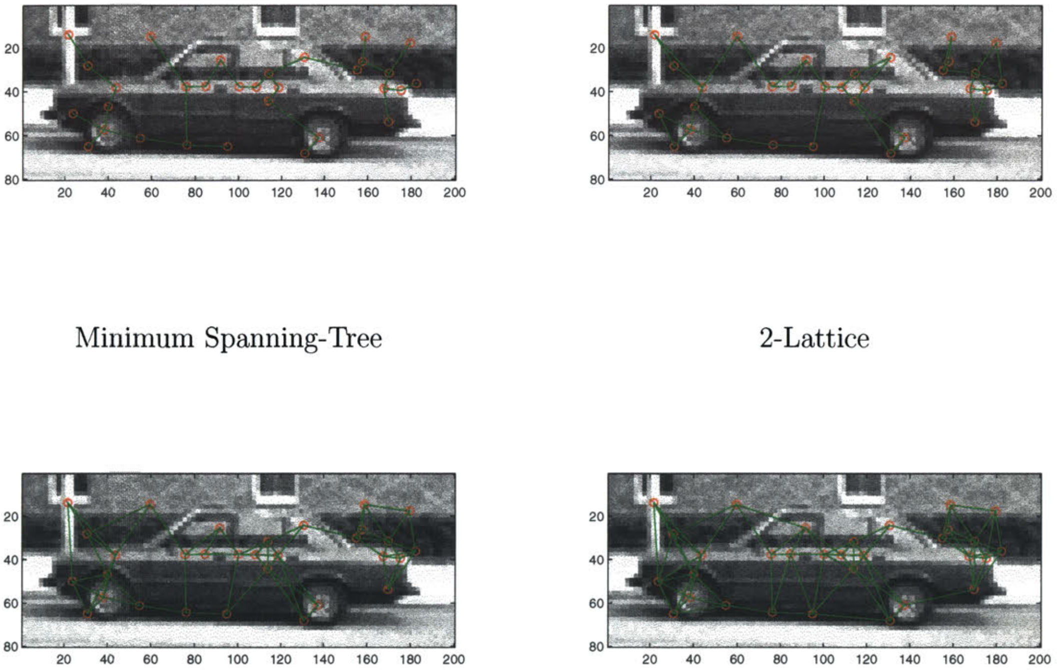
4.4.1 Graph Structure and Inference

The graph E encodes the amount of connectivity between the hidden variables h_j. Intuitively, E determines the ability of our model to capture conditional dependencies between part assignments, so that is natural to ask what are the effects of the graph on classification performance. Or in other words, do we loose something by having less densely connected graphs?

There are several possible ways of defining E. If E has no connectivity (i.e., E contains no edges) the potential function for our model reduces to:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_j \phi(x_j) \cdot \theta(h_j) + \sum_j \theta(y, h_j) \tag{4.2}$$

This graph might be too poor to capture important dependencies between part assignments, specially given that our observations contain overlapping image patches.



Minimum Spanning-Tree

2-Lattice

3-Lattice

4-Lattice

Figure 4-6: Encoding Part Dependencies in the model

Another option for defining E is to use a minimum spanning-tree (Figure 4.6, top left), where the weights on the edges are the distances between the corresponding image patches. The advantage of using such a graph is that, as we mentioned earlier, when E contains no cycles, and Ψ takes the form in Eq 3.4, we can perform exact inference on E, using belief propagation on time: $O(|E||\mathcal{Y}|)$.

Another way of encoding useful dependencies is by defining E to be an n-Lattice over the local observations (Figure 4.6, top right, bottom right and bottom left). We build an n-neighbor lattice by linking every node to its n closest nodes, (i.e., the nodes that correspond to the n closest local observations). The downside of such approach is that when E contain cycles computing exact inference becomes untractable, and we need to resort to approximate methods. For these experiments we choose to use loopy-belief propagation since it has been shown to have good convergence properties, but any other variational method could be applied.

4.4.2 Local Connectivity Experiments

The goal of our experiments is to evaluate the effect of different neighborhood structures on recognition performance. We conducted 4 experiments on 2 data sets. The first data set contained 400 images of side views of car and the second data set contained 400 image of rear views of cars. We will refer to these data sets as side and rear respectively.

For each of the 4 experiments we define a neighborhood structure for our model and train a two class classifier (object vs. background) to distinguish between a category from a single viewpoint and background. The experiments were carried out in the following manner: first each data set was split into 3 data sets, a training data set of 200 images, a validation set of 100 images and a testing set of 100 images. As in the experiments in Section 4.3 the validation set was used to select the regularization term (i.e., the variance of the Gaussian prior).

For the first experiment we define E to be an unconnected graph, for the second a minimum spanning-tree, for the third a 2-lattice and for the fourth a 3-lattice. For the first and second experiments gradient ascent is initialized randomly while for the

Comparative Performance - Rear Data Set	
0-Connectivity	88%
Min S-Tree	91%
2-Lattice	92%
3-Lattice	91%
Comparative Performance - Side Data Set	
0-Connectivity	90%
Min S-Tree	99%
2-Lattice	99%
3-Lattice	99%

Table 4.4: Comparative Results

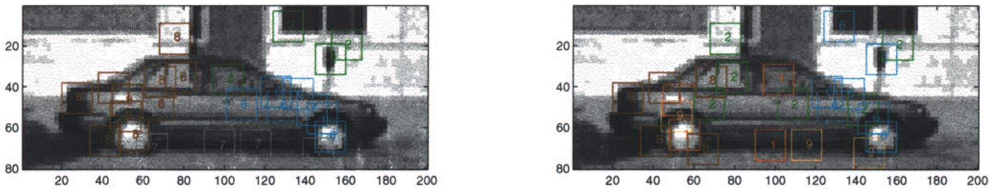


Figure 4-7: Viterbi labellings for min spanning tree and unconnected model, left and right respectively

third and fourth experiments we use the minimum spanning-tree solution as initial parameters.

Table 4.4 shows the equal error rates for the 4 variants of the model for the Side and Rear data set respectively. From this table we observe a significant improvement in performance when some form of part connectivity is incorporated into the model. On the Side data set the equal error rate increases from 90% to 99% and on the Rear data set from 88% to 91%. This is not surprising because our local observations contain overlapping patches and thus it is reasonable to expect the model to gain in performance by learning smooth part assignments.

Figure 4.7 shows the most likely assignment of parts to features for the min spanning tree model and the unconnected model for an example in which the min spanning tree model gives a correct classification but the unconnected model fails to do so. The first thing to notice is that both models give smooth part assignments, this is because the normalized location is a feature of the patch representation. Given the low resolution of the images the model relies mainly on the location of the detected features. However, as we would expect the minimum spanning-tree gives smoother part assignment than the unconnected one which allows it to classify the example correctly.

The second thing to notice is that for these data sets the minimum spanning-tree model shows no worst recognition performance than the models that use more densely connected graphs. This confirms our hypothesis that the minimum spanning-tree encodes sufficient dependency constraints.

In (Crandall 2005) Crandall presented a class of statistical generative methods for part-based object recognition that similarly to ours can be parameterized according to the degree of spatial structure that they can represent (i.e., dependencies between parts). Using the same data sets that we used for our experiments they compared the effects of incorporating different amounts of spatial structure in their model and showed that for the object classes tested a relatively small amount of spatial structure in the model can provide recognition performance as good as the performance obtained from more complex models that encode more spatial structure. Thus our results seem to be in accord with theirs though more experiments would be needed to extend this conclusions to arbitrary object classes.

One possible reason for the lack of improvement is that approximate inference might make the computation of the gradient unstable. In fact we observed in preliminary experiments that when the lattice model is not initialized with the minimum spanning-tree it results in poor performance. Thus the relatively good performance of the min-spanning tree might be due to the fact that is an adequate trade-off between the benefits of connectivity and exact inference.

Chapter 5

Joint detection and segmentation

5.1 Chapter Overview

In this chapter we extend the detection model presented in Chapter 3 to make use of fully-labelled data and perform joint detection and segmentation. Section 5.2 reviews related work on joint detection and segmentation. Section 5.3 shows how a segmentation variable can be incorporated to our previous model so that we can perform joint detection and segmentation. Finally, Section 5.4 presents experiments for the joint detection and segmentation task.

5.2 Previous work on joint detection and segmentation

The idea of using class specific constraints to guide image segmentation has received some attention in the last few years. Yu (Yu 2002) proposed a spectral graph framework that combines top-down and bottom-up information into a constrained eigenvalue problem. Borenstein (Borenstein 2002) proposed a class-specific, top-down segmentation which assumes that the object present in the image is known and that a set of discriminative templates for the class are available (learned from unsegmented training images). The segmentation task is formulated as finding the optimal match between a set of image patches and the object templates. The reliability of a match is measured in terms of the similarity of individual matches and the overall consistency of the configuration.

In a more recent work Borenstein (Borenstein 2004) presented a hierarchical approach for combining top down and bottom up segmentation. They model the space of possible segmentations as a random field where each image site corresponds to a region in a hierarchical tree obtained with a bottom-up segmentation. The single node potentials at each site incorporate class specific top-down constraints and the pairwise potentials between child and parent nodes in the hierarchical segmentation enforce bottom-up smoothness constraints. Segmentation is performed by finding the most likely foreground/background labelling given these constraints, which can be efficiently computed using belief propagation. These previous approaches integrate class specific constraints to improve segmentation assuming they know that the object is present, whereas our method learns to jointly detect and segment the object.

5.3 Extending the Model to perform joint detection and segmentation

For the joint detection and segmentation task we assume as in Chapters 3 and 4 that we are given a training set of n partially labelled pairs. Each such example is of the form $(\mathbf{x}, y)$, where $y \in Y$ is the category of the object present in image $\mathbf{x} = [x_1, \ldots, x_m]$ and x_j is the j-th image region. We will say that t is a partially-labelled example if $t = (\mathbf{x}, y)$.

In addition to the partially labelled pairs we assume that we are given l fully labelled triples of the form $(\mathbf{x}, y, \mathbf{s})$, where $\mathbf{s} = [s_1, \ldots, s_m]$ and each $s_j \in Y$ corresponds to a labelling of x_j with some member of Y. This variable is hidden for the partially-labelled data but observed for the fully-labelled data. We will say that t is a fully-labelled example if $t = (\mathbf{x}, y, \mathbf{s})$.

From this training set we would like to learn models that map images $\mathbf{x}$ to labels y

in the detection task, and that map images $\mathbf{x}$ to segmentations $\mathbf{s}$ in the segmentation task.

Our approach decouples the problem of finding the object in the image into two sub-problems: first deciding if the object is present in the image (modelled by the presence variable y), and then finding the location of the object in the image or equivalently determining which image regions correspond to the object (modelled by the segmentation variable s). As before for any image $\mathbf{x}$ we also assume a vector of hidden "parts" variables $\mathbf{h} = \{h_1, h_2, \ldots, h_m\} \in H$.

Finally, we assume we are given a mapping M that maps a variable s_j to a non-empty subset of H. Intuitively, this mapping selects for each category Y a subset of the hidden parts that is going to be used to model the category. Having this intermediate mapping is an important part of our approach because it allows us to model the category of each image region with a set of parts (i.e., a set of hidden states) rather than a single state so that we can combine part-based object detection with region labelling.

We say that $\mathbf{h}$ is consistent with $\mathbf{s}$ if for all j , $h_j \in M(s_j)$. We will denote the set of consistent hidden variable assignments by $const(\mathbf{s})$.

Given the above definitions, we define a conditional model:

$$P(y, \mathbf{s}|\, \mathbf{x}, \theta) = \frac{\sum_{\mathbf{h} \in const(\mathbf{s})} e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y',\mathbf{h}} e^{\Psi(y',\mathbf{h},\mathbf{x};\theta)}}. \tag{5.1}$$

where as before θ are the parameters of the model and $e^{\Psi(y',\mathbf{h},\mathbf{x};\theta)}$ is the potential function described in Chapter 3. We will be assuming that we have a distribution:

$$P(y, \mathbf{s}, \mathbf{h} \mid \mathbf{x}, \theta) = \frac{e^{\Psi(y,\mathbf{h},\mathbf{x};\theta)}}{\sum_{y'\mathbf{h}} e^{\Psi(y',\mathbf{h},\mathbf{x};\theta)}} \tag{5.2}$$

defined over the space of $\mathbf{h}$ values such that $\mathbf{h} \in const(\mathbf{s})$

To perform segmentation we follow a two step process where we first perform detection and then compute the best segmentation given the class label Y. For a new test image $\mathbf{x}$, and parameter values θ^* induced from a training set, we will perform detection, by taking the label for the image to be $y^* = argmax_y P(y|\ \mathbf{x}, \theta)$ where $P(y|\ \mathbf{x}, \theta) = \sum_{\mathbf{s}} P(y, \mathbf{s}|\ \mathbf{x}, \theta)$. We then label each image region by finding $\mathbf{s}^* = argmax_{\mathbf{s}} P(\mathbf{s}|y^*, \mathbf{x}; \theta)$ where $P(\mathbf{s}|y^*, \mathbf{x}; \theta) = \sum_{\mathbf{h} \in const(\mathbf{s})} P(\mathbf{h}|y^*, x; \theta)$.

We approximate this by computing $h_i^* = argmax_{h_i} \sum_{h_i \in h_i const(\mathbf{s})} P(h_i|y^*, \mathbf{x}; \theta)$ and using the deterministic mapping M to obtain $s_i^* = M^{-1}(h_i^*)$. That is once we determined the best part assignment we map each variable h_j to its corresponding y. The exact computation of $argmax_{\mathbf{s}} P(\mathbf{s}|\mathbf{x})$ would require marginalizing over the category variable y but unfortunately this computation is untractable under our model so we resort to a commonly used approximation.

We use the following objective function in training the parameters θ of the model:

$$L(\theta) = \sum_{t \in \, TrainingSet} \log P(t|\mathbf{x}, \theta) - \frac{1}{2\sigma^2} \parallel \theta \parallel^2 \tag{5.3}$$

where $P(t|\mathbf{x}, \theta) = P(y|\mathbf{x}, \theta)$ if t is a partially-labelled example and $P(t|\mathbf{x}, \theta) = P(y, \mathbf{s}|\mathbf{x}, \theta)$ if t is a fully-labelled example

As in Chapter 3 the first term in Eq 5.3 is the log-likelihood of the data and the second is a regularization term; we use gradient ascent to search for the optimal parameters values $\theta^* = argmax_\theta L(\theta)$ under this modified joint criterion.

The rest of the model is the same as the one described in Chapter 3, but notice that different from that model the new likelihood function implies that we are optimizing for joint detection and segmentation.

5.3.1 Parameter estimation in the joint detection and segmentation model

The gradient for the partially labelled examples is identical to that described in Chapter 3. Thus in this section we derive the gradient with respect to the fully labelled examples only. Consider the likelihood term that is contributed by the i'th fully labelled training example, defined as:

$$L_i(\theta) = \log P(y_i, \mathbf{s_i} \mid \mathbf{x}_i, \theta) = \log \left(\frac{\sum_{\mathbf{h} \in const(\mathbf{s})} e^{\Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}}{\sum_{y', \mathbf{h}} e^{\Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}}. \right) \tag{5.4}$$

We first consider derivatives with respect to the parameters θ_l^1 corresponding to features $f_l^1(j, y, h_j, \mathbf{x})$ that depend on single hidden variables. Taking derivatives gives

$$\frac{\partial L_i(\theta)}{\partial \theta_l^1} = \sum_{\mathbf{h} \in const(\mathbf{s})} P(\mathbf{h} \mid y_i, \mathbf{s_i}, \mathbf{x}_i, \theta) \frac{\partial \Psi(y_i, \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta_l^1} - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \frac{\partial \Psi(y', \mathbf{h}, \mathbf{x}_i; \theta)}{\partial \theta_l^1}$$

$$= \sum_{\mathbf{h} \in const(\mathbf{s})} P(\mathbf{h} \mid y_i, \mathbf{s_i}, \mathbf{x}_i, \theta) \sum_{j=1}^{m} f_l^1(j, y_i, h_j, \mathbf{x}_i) - \sum_{y', \mathbf{h}} P(y', \mathbf{h} \mid \mathbf{x}_i, \theta) \sum_{j=1}^{m} f_l^1(j, y', h_j, \mathbf{x}_i)$$

$$= \sum_{j,a} P(h_j = a \mid y_i, \mathbf{s_i}, \mathbf{x}_i, \theta) f_l^1(j, y_i, a, \mathbf{x}_i) - \sum_{y', j, a} P(h_j = a, y' \mid \mathbf{x}_i, \theta) f_l^1(j, y', a, \mathbf{x}_i)$$

where:

$$P(h_j = a \mid y, \mathbf{s}, \mathbf{x}, \theta) = \frac{\sum_{h_j = a \wedge \mathbf{h} \in const(\mathbf{s})} P(\mathbf{h} \mid y, \mathbf{x}, \theta)}{\sum_{\mathbf{h} \in const(\mathbf{s})} P(\mathbf{h} \mid y, \mathbf{x}, \theta)}$$

(5.5)

which as before can be shown to be efficiently computed using belief propagation. Similarly the gradient for the pairwise features can be written in terms of $P(h_j = a, h_k = b \mid y, \mathbf{s}, \mathbf{x}, \theta)$ which can also be efficiently computed using belief propagation. Thus as in the detection only model we can do efficient inference and parameter estimation.

5.4 Experiments with joint detection and segmentation

To test the performance of our model on both the detection and segmentation task we conducted a set of experiments on a subset of the Caltech-4 data set (Cars and Motorbikes). There are two main goals of these experiments: first we would like to show that our model performs accurate detection and segmentation using only a small set of fully labelled examples. Second, we would like to show the importance of using a hidden segmentation variable and a mapping M that allows us to model both the object and background classes with a set of part variables.

We compare our model to a baseline model with no segmentation variables where both the background and object class are modelled with a single part each. Notice that for the baseline setting we can regard the part assignment variables $\mathbf{h}$ as being fully observed for the fully labelled examples. Or in other words, since there are only two parts in the model and each category is modelled using a single part, once we know the segmentation label for a given image region we automatically know its hidden part assignment.

In addition to the two part baseline we compare our model to a detection model trained with partially labelled data only and no hidden segmentation variable. We show that without constraining the model using some fully-labelled data the parts assignments learnt do not provide an accurate segmentation of the object in terms of background/foreground.

While for the experiments in Chapter 4 we used as local features high entropy patches for the joint detection and segmentation experiments in this chapter we use as features image regions or "blobs" obtained with a bottom-up segmentation (Sharon 2000). The reason for this choice of features is that to perform segmentation we prefer features that cover the entire image, so it is more natural to use bottom up features.

Every image was pre-segmented by an initial bottom-up segmentation, using the procedure given in (Sharon 2000) which provides a hierarchical segmentation of the image. To select the scale of the segmentation we choose the finest scale segmentation subject to the constraint that the total number of segments be less than a predetermined constant (100).

Each image region x_j was represented by a feature vector $\phi(x_j)$ that consisted of the normalized location and scale of the region, its eccentricity and orientation as well as the mean response of a set of Gabor filters at different orientations and scales. As in the first set of experiments of Chapter 4 E was set to be a minimum spanning-tree

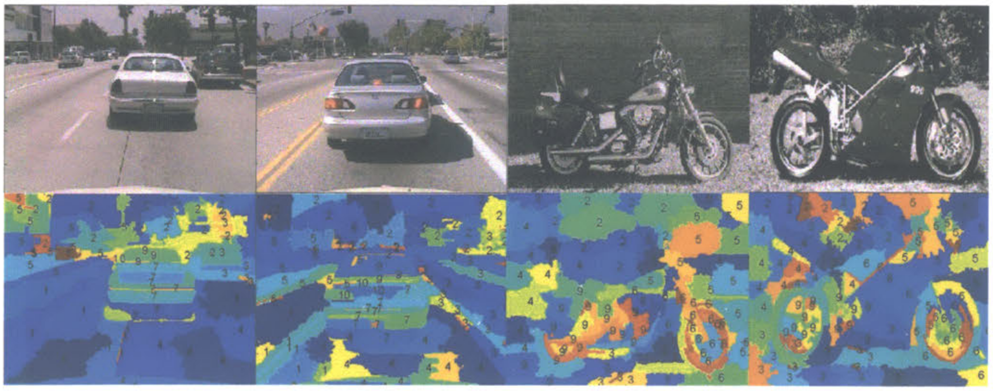


Figure 5-1: Hidden Part Assignments (FLPT). The first row shows the original image and the second row shows the optimal part assignment for each image region.

Category	P-1	P-2	P-3	P-4	P-5	P-6	P-7	P-8	P-9	P-10
Background	0 %	15%	30%	0%	5 %	10 %	% 20	0%	0%	20 %
Object	0 %	20%	20%	0%	2 %	20 %	% 23	0%	0%	15 %

Table 5.1: Distribution of region part labels for each category when learning with partially labelled data.

where the weight between two nodes h_i, h_j is the 2-D distance between the center of mass of the corresponding image regions x_i, x_j.

We trained a two class model to discriminate between the *car* class and the other three classes in the Caltec-4 data-set *motorbikes*, *planes* and *faces* as well as the generic *background* class. We also conducted experiments on the motorbike data set.

The Fully-Labelled (model FLT) training set consisted of 400 negative images (100 of each of the other classes) and 30 images of cars where we labelled each image region obtained by the bottom up segmentation with the corresponding label. If a region contained both part of the background and part of the car we labelled it according to the largest area. (Notice that since our setting of the problem is car, not-car we can regard all the negative images as being fully labelled). The Fully + Partially labelled dataset (model FLPT) consisted of the FLT dataset plus an extra 170 partially labelled car images where only the presence of the car in the image is indicated.

For a baseline detection comparison we also trained a model with 200 partially labelled positive images and 400 partially labelled negative images (model PT). For

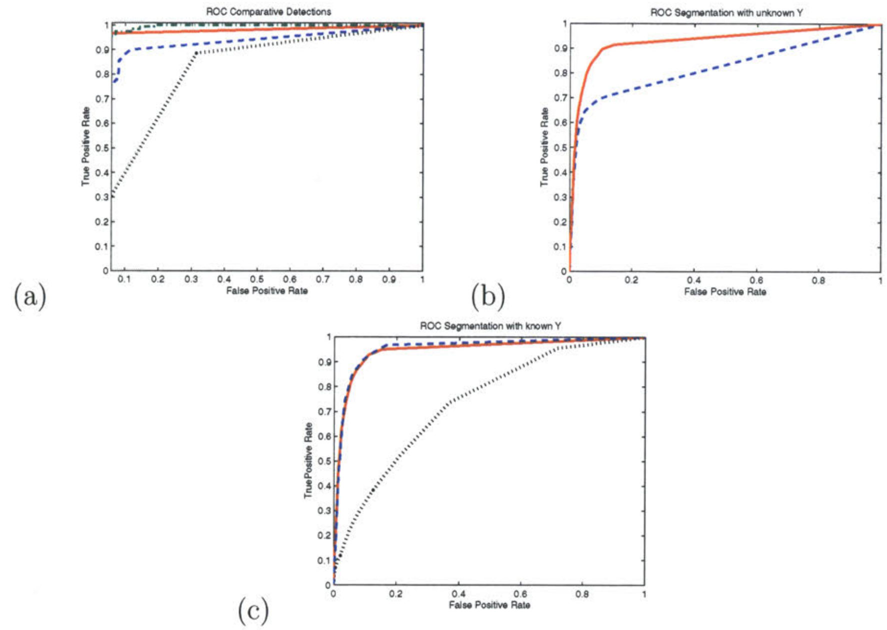


Figure 5-2: Car dataset: (a) shows detection performance, (b) segmentation performance with unknown Y, (c) segmentation performance with known Y. The 2-Part Baseline is shown with a brown dotted line, PT with green dash-dot, FLPT with solid red, and FLT with blue dashes.

these three models the total number of hidden parts was set to 10, and was defined so that the first half corresponded to background and the second half to the object class. We also trained a second baseline model with no hidden segmentation variable (model B) that used two parts, one for the background class and one for the object class and was trained with the same combination of fully-labelled and partially-labelled data as model FLPT.

We evaluated the detection and segmentation performance of our models: for detection the task was to label the whole image as containing the object or not, and for segmentation the task was to label each image region as object or background. Figure 5.2(a) shows detection ROCs for the 4 models, FLT, FLPT, PT, and B. As we would expect the performance of the FPLT model is significantly better than that of the FLT, since the later model was trained with 30 positive images only. This suggests that our model can successfully incorporate partially-labelled data to improve detection.

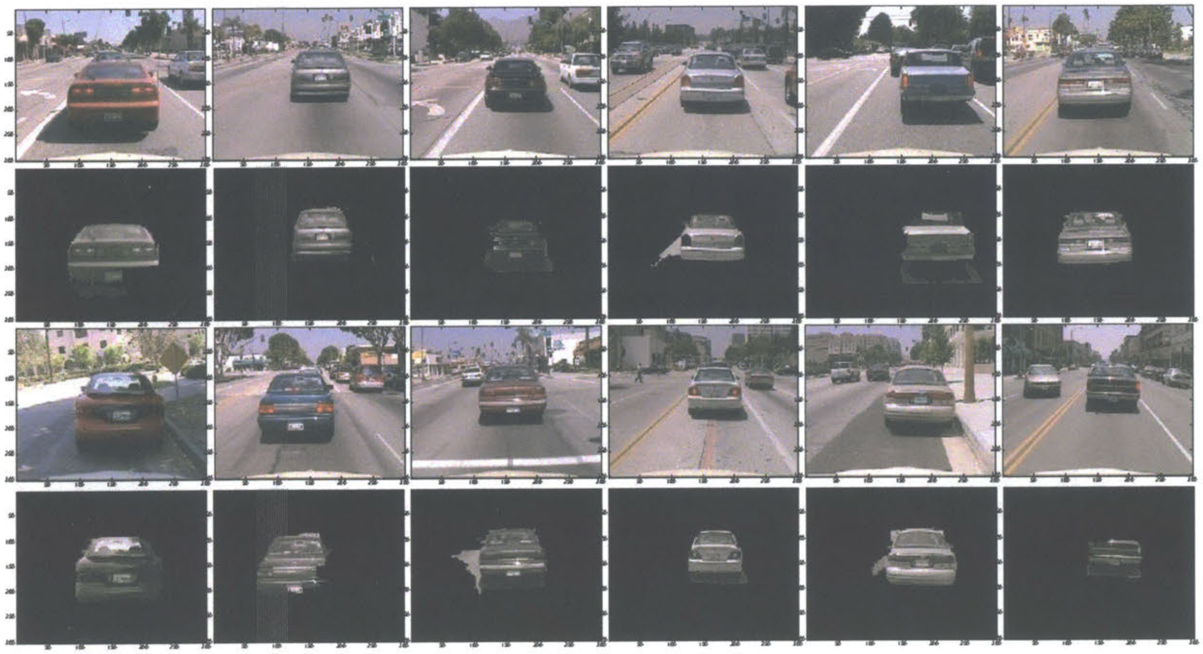


Figure 5-3: Segmentation examples from the car dataset. The first and third rows show original images, the second and fourth row the segmentation given by our model (FLPT).

We observe that FLPT performs as well as PT on the detection task. This is consistent with our assumption that it is possible to perform good detection without performing accurate localization. Our previous model (which was unable to perform segmentation) obtained equal error rates of 94.6% for the car dataset and 95% for the motorbike dataset for detection, which is comparable to the detection performance 95%, 94% for car and motorbike dataset respectively obtained with our new model FLPT.

Figure 5.2(b) shows ROC curves for segmentation with the FLT and FPLT models. The FLPT model shows better segmentation performance. Note that this is likely to be a direct consequence of improved detection performance in the FLPT model. We also performed experiments to compare segmentation performance in a setting where the effects of improved detection were factored out. In these experiments we assume that we know the correct class label y and search for the best segmentation given the known y. See Figure 5.2(c) for ROCs for these experiments.

For the model trained with partially labelled data only we do not have a direct

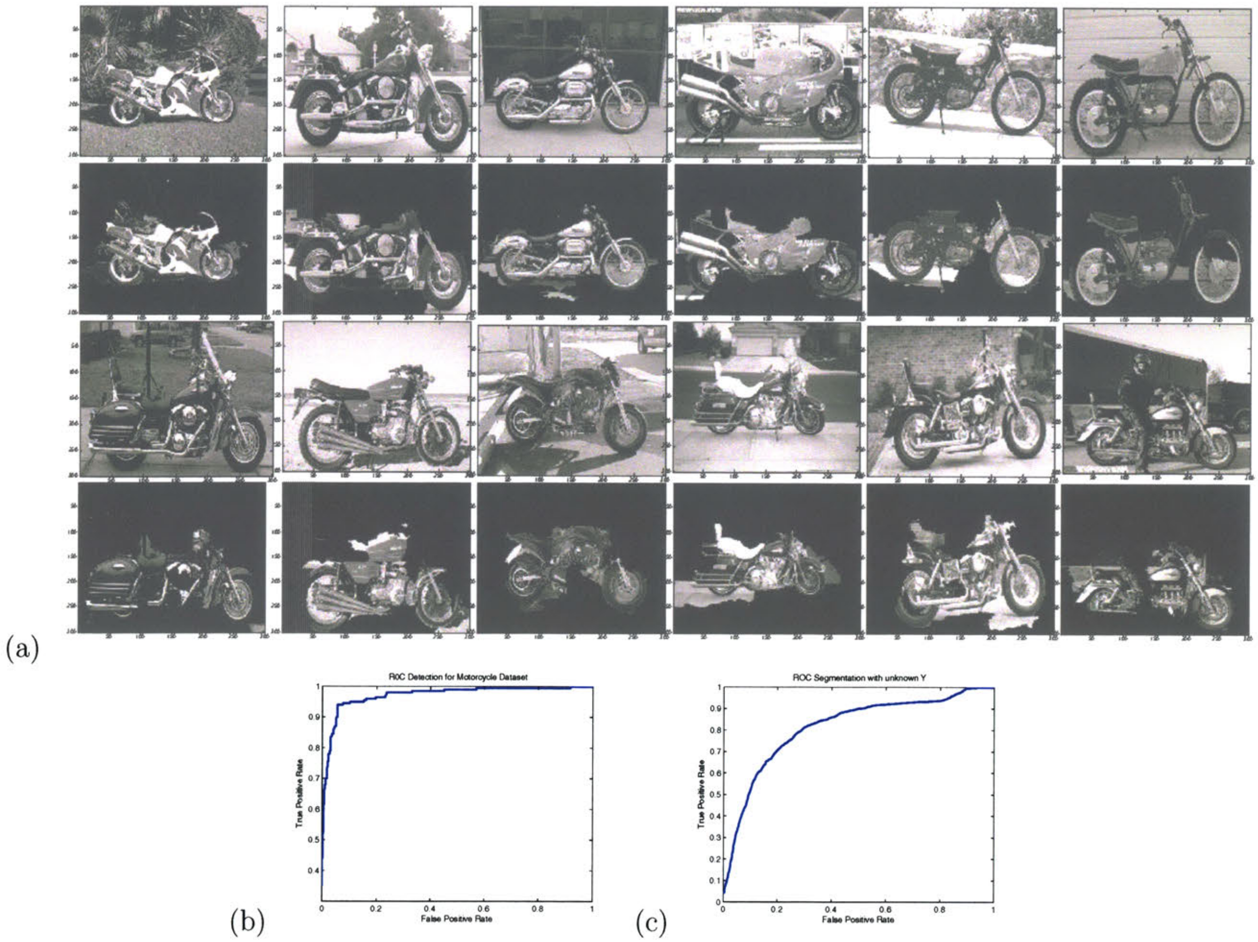


Figure 5-4: Motorbike dataset results (FLPT). (a) shows segmentation examples, (b) detection performance, and (c) segmentation performance with unknown Y.

way of estimating segmentation performance. Instead we calculate the frequency with which a part is assigned to a background region and to an object region respectively; we compute $\frac{n(Y=object,h_i=p)}{n(object)}$ and $\frac{n(Y=background,h_i=p)}{n(background)}$ where $n(Y = background, h_i = p)$ is the number of times the PT model assigns part p to a background region and $n(background)$ is the total number of background regions (Table 5.1). Since parts seem to occur with similar frequency in both categories we can see that without constraining the model with fully labelled (segmented) data it is hard to build a model that obtains a meaningful segmentation.

Figure 5.2(a) shows how using multiple parts to model both the background and the object class improves detection accuracy significantly. By using multiple parts the model is able to learn discriminative patterns that would be hard to learn otherwise. Furthermore, Figure 5.2(c) shows that using an intermediate segmentation variable combined with a hidden part assignment variable is important for performing accurate segmentation, as multiple parts may be needed to model the variability in the background and object classes.

See Figure 5.1 for evidence that FLPT has learned a part-based model for the car where the upper and lower section of the car are modelled by different parts and it has also learned a discriminative pattern of the context surrounding the car. Similarly, we see that FLPT has learned different parts for the front and back of the motorbike.

Figure 5.4 shows detection and segmentation performance for the motorbike data set. Comparing the bottom-up segmentation on the car and motorbike data sets (Figure 5.1), one can see that there seems to be more variability in the shape of the object regions as well as in the overall configuration for the motorbikes, which may make segmentation more challenging for that category.

Chapter 6

Conclusions

In this thesis we have presented a novel approach that incorporates hidden variables and combines class conditional random fields into an unified framework for object detection and segmentation. One of the main advantages of the proposed model in contrast to other object recognition approaches it that ours does not assume the independence of local observations given their assignments to parts in the model.

In addition, as with other CRFs and other maximum entropy models, our approach offers a significant amount of representational freedom as it can combine arbitrary observation features for training discriminative classifiers with hidden variables. By making some assumptions about the joint distribution of hidden variables we have shown that one can derive efficient training algorithms based on dynamic programming.

Since an object recognition system should be able to perform both detection and segmentation in this thesis we also developed a joint model for object detection and segmentation. The advantage of this model is that it can combine fully labelled and partially labelled data into a principled discriminative framework for detection and segmentation. The key difference between our approach and other part based approaches is that the incorporation of hidden parts and segmentation variables allows us to naturally combine fully and partially labelled data. The proposed latent variable model can learn sets of part labels for each image site, which allows us to merge part-based detection with part-based region labelling (or segmentation).

One important limitation of our model is that it is dependent on the feature detector. Furthermore, our model might learn to discriminate between classes based on the statistics of the feature detector and not the true underlying data, to which it has no access. This is not a desirable property since it assumes the feature detector to be consistent. As future work we would like to incorporate the feature detection process into the model.

Bibliography

[1] E. Borenstein and S. Ullman. Class-Specific, Top-Down Segmentation. In *ECCV*, 2002.

[2] E. Borenstein, E. Sharon and S. Ullman. Combining Top-down and Bottom-up Segmentation. In *CVPR*, 2004.

[3] H. Cheng, C. A. Bouman. Multiscale Bayesian Segmentation using a trainable context model. In *IEEE, Transactions on Image Processing*, 2001.

[4] D. Crandall, P. Felzenswalb and D. Huttenlocher. Spatial Priors for Part-Based Recognition using Statistical Models. In *CVPR*, 2005.

[5] G. Dork, C Shmid. Object Class Recognition using discriminative local features. In *IEEE, PAMI*, submitted.

[6] R. Fergus, P. Perona,and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[7] S. Kumar and M. Hebert. Discriminative random fields: A framework for contextual interaction in classification. In *ICCV*, 2003.

[8] S. Kumar and M. Hebert. Multiclass Discriminative Fields for Parts-Based Object Detection. In *Snowbird Learning Workshop*, 2004.

[9] J. Lafferty,A. McCallum and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[10] B. Leibe and B. Shiele. Interleaved Object Categorization and Segmentation. In *British Machine Vision Conference*, 2003.

[11] D. Lowe. Object Recognition from local scale-invariant features. In *ICCV*, 1999.

[12] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, 2000.

[13] A. Opelt, M. Fusseneger, A. Pinz, P. Auer. Generic Object Recognition with Boosting. In *IEEE PAMI*, submitted.

[14] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[15] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. In *EMNLP*, 1996.

[16] A. Quattoni, M. Collins and T Darrell. Conditional Random Fields for Object Recognition. In *NIPS*, 2004.

[17] C. Rosenberg, M. Hebert and H. Schneiderman. Semi-Supervised Self-Training of Object Detection Models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2004.

[18] E. Sharon, A. Brandt, R. Basri. Fast MultiScale Image Segmentation. In *CVPR*, 2000.

[19] H. Schneiderman, T Kanade. A statistical method for 3-D object detection aplied to faces and cars. In *CVPR*, 2000.

[20] A. Torralba, K. P. Murphy, W.T. Freeman. Sharing visual features for multiclass and multiview object detection. In *CVPR*, 2004.

[21] S.Ullman, E. Sali, M. Vidal-Naquet. A Fragment-Based Approach to Object Representation and Classification. In *Visual Form*, 2001.

[22] P. Viola, M. Jones. Rapid Object Detection using a boosted cascade of simple features. In *CVPR*, 2001.

[23] M.Weber, A. Brandt and R. Basri. Unsupervised Learning of Models for Object Recognition. *PhD. thesis, Department of Computational and Neural Systems, Caltech*, 2000.

[24] R. Wilson, C. T. Li. A class of discrete multiresolution random fields and its application to image segmentation. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2004.

[25] M.H. Yang, D. Roth, and N. Ahuja. Learning to recognize 3D objects with snow. In *ECCV*, 2000.

[26] S.Yu, R. Gross and J. Shi. Concurrent Object Recognition and Segmentation by Graph Partitioning. In *NIPS*, 2002.