Models for Learning Spatial Interactions in Natural Images for Context-Based Classification

Sanjiv Kumar

CMU-RI-TR-05-28

August, 2005

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:
Martial Hebert, Chair
Takeo Kanade
Henry Schneiderman
John Lafferty
Andrew Blake, Microsoft Research Cambridge

Keywords: Image classification, Image context, Scene labeling, Spatial interaction, Markov Random Field, Conditional Random Field, Region classfication, Object detection, Hierarchical field

To

Sri Ramakrishna Paramahamsa

Abstract

Classification of various image components (pixels, regions and objects) in meaningful categories is a challenging task due to ambiguities inherent to visual data. Natural images exhibit strong contextual dependencies in the form of spatial interactions among components. For example, neighboring pixels tend to have similar class labels, and different parts of an object are related through geometric constraints. Going beyond these, different regions e.g., sky and water, or objects e.g., monitor and keyboard appear in restricted spatial configurations. Modeling these interactions is crucial to achieve good classification accuracy.

In this thesis, we present discriminative field models that capture spatial interactions in images in a discriminative framework based on the concept of Conditional Random Fields proposed by Lafferty et al. The discriminative fields offer several advantages over the Markov Random Fields (MRFs) popularly used in computer vision. First, they allow to capture arbitrary dependencies in the observed data by relaxing the restrictive assumption of conditional independence generally made in MRFs for tractability. Second, the interaction in labels in discriminative fields is based on the observed data, instead of being fixed a priori as in MRFs. This is critical to incorporate different types of context in images within a single framework. Finally, the discriminative fields derive their classification power by exploiting probabilistic discriminative models instead of the generative models used in MRFs.

Since the graphs induced by the discriminative fields may have arbitrary topology, exact maximum likelihood parameter learning may not be feasible. We present an approach which approximates the gradients of the likelihood with simple piecewise constant functions constructed using inference techniques. To exploit different levels of contextual information in images, a two-layer hierarchical formulation is also described. It encodes both short-range interactions (e.g., pixelwise label smoothing) as well as long-range interactions (e.g., relative configurations of objects or regions) in a tractable manner. The models proposed in this thesis are general enough to be applied to several challenging computer vision tasks such as contextual object detection, semantic scene segmentation, texture recognition, and image denoising seamlessly within a single framework.

Acknowledgments

I would like to thank my advisor Martial Hebert for providing careful guidance throughout my graduate program at CMU. I have tremendously benefited from his powerful combination of mathematical rigor and intuition. I especially enjoyed his role as 'devil's advocate', in which he can be extremely exacting to promote a thorough understanding of the research issues. In spite of the divergence of our views at times, Martial has played a key role in developing not only my research abilities but also my presentation skills. I would also like to express my gratitude to my other thesis committee members John Lafferty, Andrew Blake, Takeo Kanade, and Henry Schneiderman for providing their invaluable comments on the research work as well as the organization of the thesis.

I am indebted to Vijayan Asari and Damu Radhakrishnan for persuading me to go for a doctorate in the first place. At CMU, it has been a great pleasure to exchange ideas with the members of the computer vision reading group. My thesis work owes immensely to long and fruitful discussions with Jonas August and Tom Minka. I doubt that anyone who comes in their contact can remain unaffected by their tremendous intellectual strength, enthusiasm and energy. I express my deep sense of gratitude to my 'pseudo-commitee' members: Jonas August, Chuck Rosenberg, Goksel Dedeoglu, Srinivas Narasimhan, Daniel Huber and Raju Patil. They have constantly suffered the pain of listening to my half-baked practice talks and have provided major improvements in my presentation style. Of course, I must mention Bart Nabbe, the local LaTeX guru, who has fixed my latex bugs more than once.

During my internships and at various conferences, I have enjoyed great discussions with many researchers. My interactions with Yann Lecun, Jean Ponce, Alex Loui, Alan Yuille, Martin Szummer, John Winn and many more have greatly shaped my thoughts. Thanks are also due to Antonio Torralba for generously providing me with gentle-boosting code as well as the MIT object detection database, and to Vladimir Kolmogorov and Robert Mugizi for graph min-cut code. Finally, I would like to thank many of my friends, especially Anthony Gallagher, Andreea Gallagher, Sonya Allin, Susan Dedeoglu, Tikeswar Naik, Aleksa Veselinovic and Ravi Balasubramanian, who made my stay in Pittsburgh a memorable one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

One of the fundamental problems in computer vision is that of *image understanding* or *semantic scene interpretation* i.e., to interpret the scene contained in an image as a collection of meaningful entities. This may involve parsing information in the scene at different levels. For instance, one may be interested in recognizing various regions or objects in an image e.g., whether the scene contains *sky* or a *phone*, and at what location. At a relatively higher level, one may want to find the general class of a scene e.g., the scene is an *office* or a *beach*, or the event summarizing the scene e.g., the scene is from a *birthday party*. Scene understanding in computer vision presents the paradox that, in order to recognize an object, its surroundings must be recognized first, but to recognize the surroundings, the objects must be recognized first [132]. For instance, if we can recognize that the scene contains *water* and *sand*, there is a high probability that the scene is a *beach*. Similarly, presence of a *birthday cake* is a strong indication of the scene being from a *birthday party*.

In this thesis, we address the problem of classification or labeling of various components in natural images, where a component may be an image pixel, a patch[1] (rectangular or irregularly shaped), or an object. Following the conventional usage, by *natural images* we mean non-contrived scenes that are encountered commonly in our surroundings i.e., regular indoor and outdoor scenes. These images may contain both man-made as well as natural objects such as sky, vegetation etc. occurring in

[1]In this thesis we will call a rectangular patch a *block* and an irregularly shaped patch a *region*.

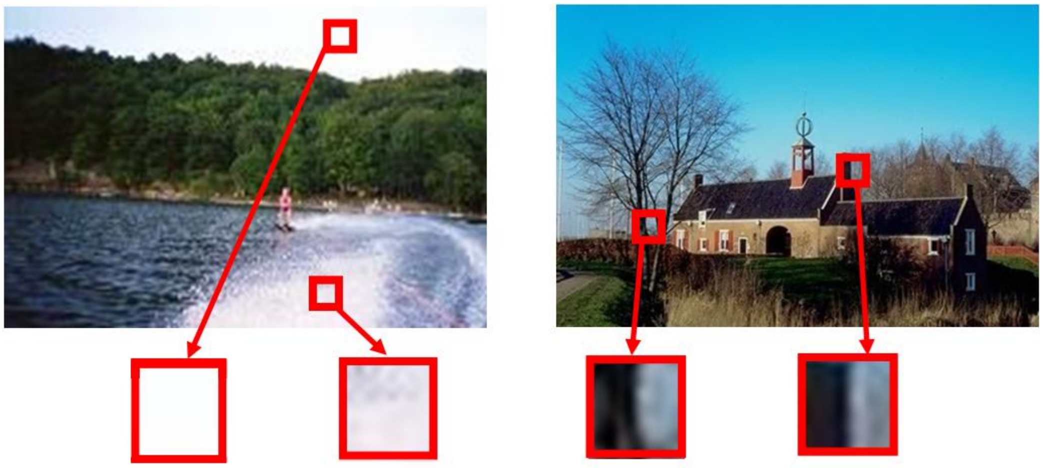


Figure 1.1: Classification of image components is difficult due to ambiguities in their appearance. In the left image, sky and water regions look similar while in the right image, tree and building regions look similar. Context can help resolve these ambiguities.

nature. In addition, we will deal with problems in which only a single static image of the scene is given, and no 3D geometric or motion information is available. This makes the classification task more difficult.

1.2 The Curse of Ambiguity

The problem of detecting and classifying regions and objects in images is a challenging task due to ambiguities in the appearance of the visual data. These ambiguities may arise either due to the physical conditions such as illumination and pose of the scene components with respect to the camera, or due to the intrinsic nature of the data itself. The use of context can help alleviate this problem significantly. For example, as shown in Figure 1.1, just on the basis of the appearance, it may be difficult to differentiate a sky patch from a water patch but their relative spatial configuration with respect to other regions removes this ambiguity. Similarly, a patch from a tree may appear locally very similar to another patch from a building (Figure 1.1, right image). But if we look at larger neighborhoods of the patch, it is easy to classify which patch is a building patch.

It is well known that natural images are not a random collection of independent pixels. To illustrate this point, a natural image is shown in Figure 1.2 (a). Figure

(a) (b) (c)

Figure 1.2: An illustration of the fact that natural images contain strong spatial dependencies rather than being a bag of random independent pixels or blocks. (a) A natural image. (b) Image obtained by randomly scrambling the pixel intensities of the original image in (a). (c) Image obtained by randomly scrambling the original image blocks.

1.2 (b) shows the image obtained by randomly scrambling the pixels of the previous image. It is obvious that the original image gives us a perception of a coherent scene because there are spatial dependencies in the image which are lost in the scrambled image. The scrambled image seems like random noise even though all the intensities, present in the original image, are also present in this image. Similarly, if one now scrambles bigger blocks instead of pixels (Figure 1.2 (c)), the coherency is still broken. This demonstrates that it is important to use the contextual information in the form of spatial dependencies for the analysis of natural images. In fact, one would like to have total freedom in modeling long range complex data interactions without restricting oneself to small local neighborhoods. This idea forms the core of the research presented in this thesis. The spatial dependencies may vary from being local to global and the challenge is how to maintain global spatial consistency using models that only need to consider relatively local dependencies.

1.3 The Nature of Contextual Interactions

There are several types of contextual interactions one would like to model to achieve robust classification in images. The simplest type of interaction is based on the notion of spatial smoothness of labels in natural images. According to this, neighboring pixels tend to have similar labels (except at the discontinuities). For example, if a pixel in left image in Figure 1.1 has label *sky*, there is a high probability that the neighboring pixels also have the same label except at the boundaries. In fact, the underlying smoothness of natural images forms the basis for recovering the true image from its noisy version in image denoising applications (Figure 1.4 (c)). These type of interactions are generally restricted to the pixel level. However, in addition to these, there exist significant interactions among bigger regions in images. In the previous example (Figure 1.1, left image), different semantic regions follow plausible spatial configurations e.g., sky tends to occur above water or sand[2].

In addition to the interaction in labels, there are also complex interactions in the observed data that might be required for classification purposes. Consider the task of detecting structured textures (e.g., man-made structures such as buildings) in a given image. The data belonging to this type of textures is highly dependent on its neighbors. This is because, in man-made structures, the lines or edges at spatially adjoining regions follow some underlying organization rules rather than being random (see Figure 1.1, right image).

Now, considering the case of parts-based object detection, one would like to detect different parts of an object to form a hypothesis about the presence of the whole object. For example, in Figure 1.3 (a), we are interested in detecting a *phone*. Different parts of the phone such as handle, keypad and front panel are related to each other through geometric and, possibly, photometric constraints. The phone can be detected in the scene if we can find the locations of these parts. However, to reliably detect these parts, we need to encode not only the appearance of each individual part but also the spatial relationships among various parts. Thus, in this case, context is applied using the mutual relationships of different parts.

Finally, the contextual interactions for object detection are not limited to the parts of a single object. These may include interactions among various objects or regions

[2] In this work we assume that the natural orientation of an image is given. This is not a very restrictive assumption since even human vision is known to be very sensitive to incorrect image orientation. One such example is shown in Appendix C (courtesy Bach [6]).

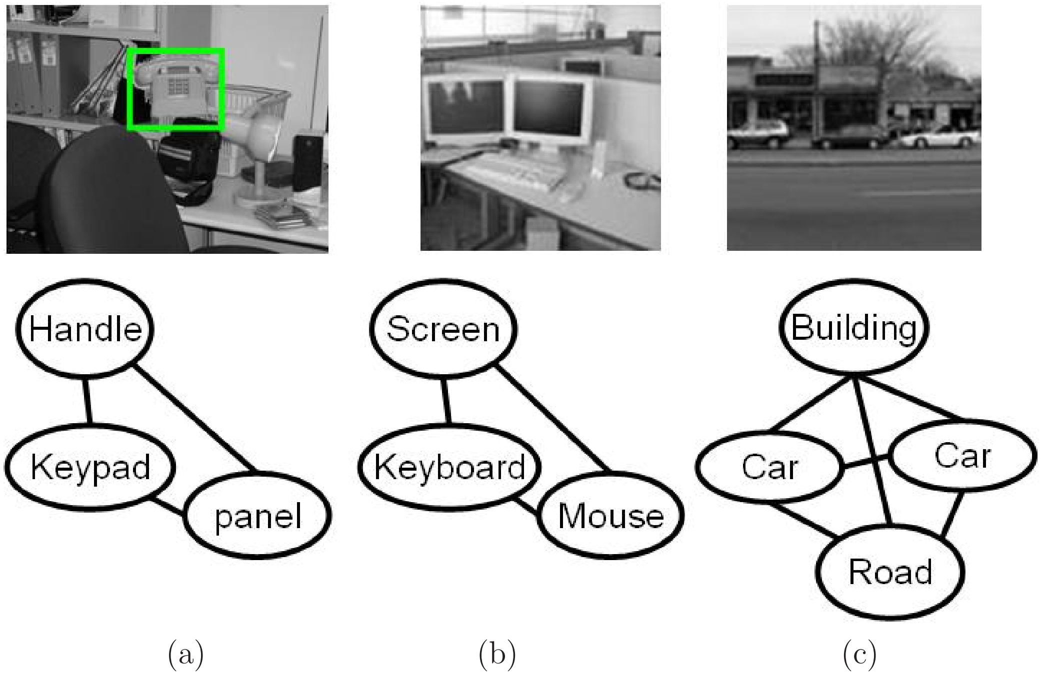


Figure 1.3: Context is important for the detection of objects in their natural surroundings. (a) Different parts of an object (phone) are related through geometric constraints that can help in robust detection of individual parts. (b) Different objects (monitor, keyboard and mouse) in a scene occur in restricted configurations which can help in detecting objects with impoverished appearance e.g., mouse. (c) Context from other regions e.g., buildings and roads can be helpful in detecting objects (cars).

in the scene. For example, as shown in Figure 1.3 (b), the presence of a monitor screen increases the probability of having a keyboard or mouse nearby. Exploiting such contextual information is crucial especially for detecting those objects that have impoverished appearances such as the mouse in this case. Similarly, the presence of regions such as buildings and roads in a scene restricts the possible locations a car can take in the image (Figure 1.3 (c)).

To summarize, context in images can be broadly divided into two categories. First, *local context* e.g., local smoothness of pixel labels in images or interactions among different parts of an object, and second, *global context* such as interaction among bigger objects and regions in images. In this thesis, we address the challenge of how to model different types of context which may include complex dependencies in the observed image data as well as the labels in a principled manner. Ideally, one would like to find a computational model that can learn all relevant types of context automatically in a single consistent framework using the training data. So far, computer vision researchers have mostly focused on modeling context at pixel or part level [49][34][40][35][146]. Recently there has been some work in modeling context at a higher level [136][56][131][77]. However, in addition to being restricted to only one type of context, these techniques are generally restricted to specific applications. In this thesis we present a principled framework that seamlessly combines apparently diverse requirements imposed by different forms of contexts for different applications in a single model.

1.4 Modeling Contextual Interactions

While modeling contextual interactions in images, it is important to take into consideration within-class statistical variations in visual data and other uncertainties due to image noise. This naturally leads to probabilistic modeling of classification problems. In probabilistic models, the final classification task can be seen as inference over these models with respect to some cost function.

As discussed before, natural images exhibit long range dependencies and manipulating these global interactions is of fundamental interest in classification. However, direct modeling of global interactions becomes computationally intractable even for a small image. On the contrary, usually it is easy to encode the structure of local dependencies in an image from which we would like to make globally consistent predictions. This paradox can be resolved to a large extent by *graphical models*. Graphical models combine two areas viz. *graph theory* and *probability theory*, and provide a powerful yet flexible framework for representing and manipulating *global* probability distributions defined by relatively *local* constraints. Graphical models are sometimes popularly referred to as *random fields*[3] in computer vision, statistical physics and several other areas.

At this stage it will be pertinent to ask the following question: Do we really need to use graphical models for modeling context? Will a simpler strategy e.g., sequential incorporation of context suffice? For example, in Figure 1.3 (b), if we can identify the keyboard first, it will be easy to locate the mouse. This approach can reduce the computational complexity significantly. However, the main problem with this approach is that gross errors will be introduced in the mouse detection if the

[3]Usually the term 'random fields' is used only for undirected graphs. However, some authors use this term even for directed graphs [16].

keyboard was not identified correctly. Ideally, both keyboard and mouse reinforce the detection of each other simultaneously and this can be modeled in a principled manner by doing inference in a graphical model. This reasoning also satisfies the general principle of *least early commitment* by postponing the hard decisions towards the end. Hence, in this thesis, we propose contextual models based on probabilistic graphical models.

1.5 Experimental Evaluation

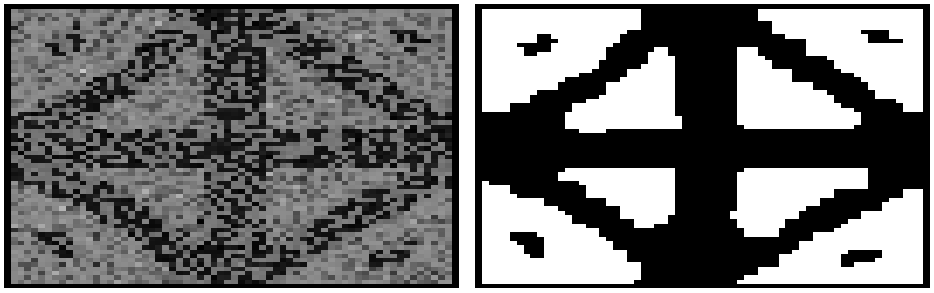
The task of labeling image regions encompasses a wide range of applications in computer vision. In this thesis, we analyze the performance of the proposed models on several datasets corresponding to different applications such as semantic segmentation, region classification, image denoising, texture recognition, and contextual object detection (Figure 1.4 and Figure 1.5). The datasets are comprised of both synthetic as well as real-world images. The synthetic images were primarily used to verify the effects of different components of the model under controlled conditions. In this thesis, experimental evaluations have been combined with the corresponding theoretical formulation in the same chapter. The experimental analysis is based on both quantitative as well as qualitative evaluation of the results.

Figure 1.4 and Figure 1.5 show some of the example results obtained using our models on different applications. Figure 1.4 (a) shows the application of semantic scene segmentation (or region classification) where we are interested in classifying different regions of the image as *sky*, *water*, *sand* and so on. This is achieved by taking into account label smoothing as well as spatial relationships of bigger regions. In Figure 1.4 (b), an application of structured texture detection (man-made structure detection) is given. For this, context in the form of spatial interactions among data from neighborhood blocks and spatial smoothness of labels was used. Figure 1.4 (c) shows an example of binary image denoising achieved using pixelwise label smoothing.

Figure 1.5 shows contextual object detection in three cases. In Figure 1.5 (a), a phone is detected by detecting various parts of the phone (shown as white squares). This is achieved using the geometric consistency between different parts as the context. Figure 1.5 (b) shows the detection of a car using the object-region interactions, i.e., relative spatial configuration of buildings, road and cars. Finally, Figure 1.5 (b) shows the detection of a mouse using the object-object interactions i.e., spatial con-

(a) Semantic Scene Segmentation

(b) Structured Texture Detection



(c) Binary Image Denoising

Figure 1.4: Various tasks in computer vision that require explicit consideration of spatial dependencies for the purpose of region labeling. The left column shows the input images and the right column shows the classification results. (a) Segmentation and labeling of input image in meaningful regions. (b) Detection of structured textures such as buildings. (c) Image denoising to restore the binary images corrupted by noise.

figurations of monitor, keyboard and mouse. Note that the detection of the mouse just on the basis of appearance is very difficult due to poor resolution. The training of all the models in the above examples was carried out in a supervised manner i.e., the models were trained using fully labeled training sets.

(a) Parts based object detection using part-part interactions

(b) Contextual object detection using object-region interactions

(c) Contextual object detection using object-object interactions

Figure 1.5: Object detection based on three different types of context. The left column shows the input images and the right column shows the detection results. (a) Detection of a phone in a cluttered scenes using geometric consistency of different parts. White squares represent different parts. (b) Car detection in an outdoor scene using interactions between car, building and road. (c) Mouse detection in an indoor scene using interactions between monitor, keyboard and mouse. Note the poor appearance of the mouse in the input image

1.6 Background Work

The issue of incorporating spatial dependencies in various image analysis tasks has been of on-going interest in vision community. In the vision literature, broadly two different types of approaches have been used to address this issue: *non-probabilistic* and *probabilistic*. We categorize a framework as non-probabilistic if the overall labeling objective is not given by a consistent probabilistic formulation even if the framework utilizes probabilistic methods to address parts of it. Among the non-probabilistic approaches, other than using weak measures to capture spatial smoothness of natural images using filters with local neighborhood supports [52][84][85], two main techniques have been used: rules-based context (Section 1.6.1) and relaxation labeling (Section 1.6.2). The probabilistic techniques have been mostly addressed under the paradigm of probabilistic graphical models (Section 1.6.3). In this section we briefly review these techniques in modeling context in computer vision.

1.6.1 Context and Early Vision

In early computer vision, extensive use of context was advocated by a large number of researchers to achieve the goal of scene understanding [40][156][47][67][54][110][62] [114][132]. The objective of most of the scene understanding systems consisted of recognizing and localizing significant objects in the scene and identifying the relevant object relationships [8]. The problem of getting semantics in the form of symbolic reasoning from the raw input images was dubbed *pixels to predicate problem* by Pentland [114].

The bottom-up schemes to recognize various objects and the scene became popular with the early work of Fischler [40]. Usually these schemes first partition a scene into regions by using general-purpose segmentation techniques. These regions are then characterized by a fixed set of attributes leading to object level labeling. The labeling process requires an inference engine to match each region to the best object model. Finally the scene itself is characterized by linking the objects together. Depending on the consistency of various objects composing the scene, object labels are refined. So, the contextual information is used in two forms: mutual relationships of objects and overall consistency of the scene. The way these systems organize and store scene knowledge is in the form of rules and graph-like structures (semantic nets, associative nets, tree-structures etc.). More details on knowledge representation in these systems are given in [25].

Along these lines, Ohta [110] used a rule-based approach to assign labels to regions obtained from a single-pass segmentation. A stumbling block in the use of rules-based approaches is their inability to deal with the statistical variations in the data. To avoid the absolute constraints imposed by the rule-based approaches, Singhal et al. [131] suggested the use of conditional histograms to make a local decision regarding assigning a label to a new region given the previous regions' labels. However, such a sequential implementation of context will suffer if an intermediate region is assigned a wrong label.

In mid-1970s, the VISIONS schema system was proposed by Hanson and Riseman [54], which provides a framework for building a general interpretation system as a distributed network of many small special-purpose interpretation systems. It introduced the notion of *schema* which embeds its own memory and procedural control strategies, acting as an expert at recognizing one type of object. The system's initial expectations about the world were represented by one or more seed schema instances. These instances predict the existence of other objects by invoking associated schema which in turn may invoke more schema. The contextual interactions and conflict resolution among various schema was again based on rule-based strategies.

Strat [132] presented a system called CONDOR to recognize natural objects for the visual navigation of an autonomous robot. The aim of this system was to utilize the context in the form of auxiliary data such as camera position and orientation, geometric horizon, date and time, weather, and digital terrain elevation data and map. This information was integrated to generate a hypothesis about scene objects which is most consistent with the global context. While analyzing generic 2D images, such meta-data is generally not available. Instead, one needs to derive the context directly from the input image itself. A comprehensive review of the use of context for recognizing natural objects in color images of outdoor scenes is given in [8].

To summarize, the main problem faced by early computer vision systems that used context for object or scene labeling was lack of principled methods to deal with uncertainty embedded inherently in image analysis applications. Attempts at using fuzzy logic [55][93] proved to be insufficient as image data usually has significant noise and other within-class variations. Even though efforts were made to represent global uncertainty using graph structures [153][121][40], the tools available for learning and inference over these structures were limited. Thus, ad-hoc procedures for resolving ambiguities using rules remained a popular strategy in early vision [25] making the resulting systems unreliable or constrained to a narrow domain.

1.6.2 Relaxation Labeling

Among the non-probabilistic approaches to modeling context, perhaps the most popular one is *Relaxation Labeling (RL)* proposed by Rosenfeld et al. [122]. This work was inspired by the work of Waltz [144] concerned with discrete relaxation on how to impose a global consistency on the labelings of idealized line drawings where the object and object primitives were assumed to be given. Since the introduction of RL, several probabilistic relaxation approaches have been suggested to provide a better explanation of the original heuristic updates of the label responsibilities [72][70][22]. In spite of successes of probabilistic RL in several applications, there are many ad-hoc assumptions in various RL frameworks [69]. For example, either the labels are assumed to be independent given the relational measurements at two or more sites [22] or conditionally independent in local neighborhood of a site given its label [70]. Probably the most important problem with RL is that the model parameters, i.e., the compatibility coefficients are chosen on a heuristic basis. In fact, it is not even clear how to interpret these coefficients [71]. The probabilistic versions of RL do allow viewing compatibilities as conditional probabilities. However, even these interpretations are valid only for the first iteration. The meaning of the probabilities yielded by subsequent iterations is increasingly speculative [71].

1.6.3 Probabilistic Graphical Models

In the probabilistic schemes, two types of graphical models, *causal* and *noncausal*, have been used extensively to incorporate spatial contextual constraints in vision problems. The causal models are directed graphs which assume that the observed image has been produced by a causal latent process. These models have been used with some success in various segmentation and labeling problems [16][21][34][152]. Our early work also explored a particular form of causal graphs [80] and the details of the model along with associated problems are discussed in Chapter 2. In this section we will focus on the background work on noncausal or undirected graphical models, which form the core of this thesis.

Markov Random Fields

Markov Random Fields (MRFs) are the most commonly used undirected graphical models in vision, which allow one to incorporate local contextual constraints in labeling problems in a principled manner. MRFs were made popular in vision by the early work of Cross and Jain [26], Geman and Geman [49], and Besag [10][11]. In this work we will focus on the use of MRFs for classification problems even though they have also been used for image synthesis problems [159][123]. MRFs are generally used in a probabilistic generative framework that models the joint probability of the observed data and the corresponding labels [94]. In other words, let $\boldsymbol{y}$ be the observed data from an input image, where $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$, $\boldsymbol{y}_i$ is the data from the i^{th} site, and S is the set of sites. Let the corresponding labels at the image sites be given by $\boldsymbol{x} = \{x_i\}_{i \in S}$. In the MRF framework, the posterior over the labels given the data is expressed using the Bayes' rule as,

$$P(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x}), \tag{1.1}$$

where the prior over labels, $P(\boldsymbol{x})$ is modeled as a MRF. For computational tractability, the observation or likelihood model, $p(\boldsymbol{y}|\boldsymbol{x})$ is assumed to have a factorized form [11][34][94][155], i.e.,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i \in S} p(\boldsymbol{y}_i|x_i). \tag{1.2}$$

An illustration of a typical MRF commonly used in computer vision is given in Figure 1.6. In MRF formulations of binary classification problems, the label interaction field, $P(\boldsymbol{x})$, is commonly assumed to be a homogeneous and isotropic Ising model (or Potts model for multiclass labeling problems) with only pairwise nonzero potentials. If the data likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ is approximated by assuming that the observed data is conditionally independent given the labels, the posterior distribution[4] over labels can be written as,

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z_m} \exp\left(\sum_{i \in S} \log p(\boldsymbol{y_i}|x_i) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \beta_m x_i x_j \right), \tag{1.3}$$

where Z_m is the normalizing constant known as the partition function, β_m is the interaction parameter of the MRF and $\mathcal{N}_i$ is the set of neighbors of site i.

However, as noted by several researchers [16][80][116][152], the assumption of con-

[4] With a slight abuse of notation, we will use the term 'MRF model' to indicate this posterior in the rest of this thesis.

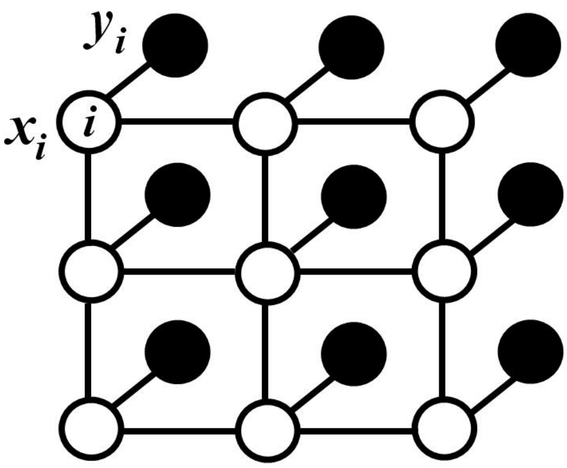


Figure 1.6: An illustration of a typical Markov Random Field (MRF) used in computer vision. The shaded circles denote the observations. At each node i, the observed data is denoted by y_i and the corresponding label by x_i. Note that the observed data is conditionally independent given the labels.

ditional independence of the data is too restrictive for several applications for the analysis of natural images. For example, consider a class that contains man-made structures (e.g. buildings). The data belonging to such a class is highly dependent on its neighbors. This is because, in man-made structures, the lines or edges at spatially adjoining sites follow some underlying organization rules rather than being random (See Figure 1.4(b)). This is also true for a large number of texture classes that are made of structured patterns, and other object detection applications where geometric (and possibly appearance) relationships between different parts of an object are crucial for its detection in cluttered scenes [146][35][33].

Some attempts have been made in the past to model the dependencies in the observed image data. In [72], a technique was presented that assumes the noise in the data at neighboring sites to be correlated, which is modeled using an auto-normal model. However, the authors do not specify a field over the labels, and classify a site by maximizing the local posterior over labels given the data and the neighborhood labels. In the context of hierarchical texture segmentation, Won and Derin [154] model the local joint distribution of the data contained in the neighborhood of a site assuming all the neighbors from the same class. They further approximate the overall likelihood to be factored over the local joint distributions. Wilson and Li [152] assume the difference between observations from the neighboring sites to be conditionally independent given the label field. In the context of multiscale random field, Cheng and Bouman [16] make a more general assumption. They assume the difference between the data at a given site and the linear combination of the data from that site's parents to be conditionally independent given the label at the current scale. All the above techniques make simplifying assumptions to get some sort of factored approximation of the likelihood for tractability. This precludes capturing stronger relationships in the observations in the form of arbitrarily complex features that might be desired to discriminate between different classes.

A novel pairwise MRF model is suggested in [116] to avoid the problem of explicit modeling of the likelihood, $p(\boldsymbol{y}|\boldsymbol{x})$. They model the joint $p(\boldsymbol{x}, \boldsymbol{y})$ as a MRF in which the label field $P(\boldsymbol{x})$ is not necessarily a MRF. But this shifts the problem to the modeling of pairs $(\boldsymbol{x}, \boldsymbol{y})$. The authors model the pair by assuming the observations to be the true underlying binary field corrupted by correlated noise. However, for most of the real-world applications, this assumption is too simplistic. In our previous work [80], we modeled the data dependencies using a pseudolikelihood approximation of a conditional MRF for computational tractability. In this thesis, we explore alternative ways of modeling data dependencies which permit eliminating these approximations in a principled manner. These models will be explained in detail in Chapter 3.

Another thing to note from Eq. (1.1) is that the interactions between labels are modeled by the term $P(\boldsymbol{x})$, which is seen as a prior in the Bayesian view. The main drawback of this view is that the label interactions do not depend on the observed data $\boldsymbol{y}$. This prohibits one from modeling data-dependent interactions in labels that are necessary for a variety of tasks. For example, while implementing local smoothness of labels in image segmentation, it may be desirable to use observed data to modulate the smoothness according to the image intensity gradients [17][13]. Further, in parts based object detection, to model interactions among object parts, we need observed data to enforce geometric (and possibly photometric) constraints. This is also the case for modeling higher level interactions between objects or regions in an image. In this thesis, we present models which allow interactions among labels based on unrestricted use of observations as necessary. This step is crucial to develop models that can incorporate contexts of different types within the same framework.

In related work, taking the non-probabilistic view of energy-based graphical model, Boykov and Jolly [17] have proposed an energy form that uses observed data to model pairwise interaction between labels. In this work, the smoothness parameter of the Ising model was modulated by a Gaussian over the intensity difference between a pair of pixels. Using such contrast-sensitive interactions, they have shown interesting results in the area of interactive image segmentation. However, their approach has two main drawbacks. Firstly, there is no direct probabilistic interpretation of their energy. As the authors themselves state, the choice of modulating the smoothing parameter by using image observations is rather 'ad-hoc'. Being non-probabilistic, parameter learning is hard in these models. The authors tune the smoothing and the modulation parameters by hand. Secondly, the non-probabilistic energy-based view eliminates the possibility of computing labels that are optimal for minimizing sitewise errors (i.e., sitewise zero-one loss function). This is due to the fact that there is no concept of marginals in energy-based view, which are required for minimizing the sitewise errors.

Recently, Blake et al. [13] have given a probabilistic interpretation of the contrast-sensitive image segmentation formulation suggested by Boykov and Jolly [17]. They have proposed to learn the observation model parameters along with the modulation parameters using pseudo-likelihood. This alleviates one of the main problems with the original non-probabilistic formulation of [17]. However, the parameters of the foreground and background models are learned separately. This forces one to use 'post-hoc' averaging schemes to estimate the modulation and the interaction parameters. Another problem with this approach is that the interactions among observed data are restricted to site pairs. On the contrary, the models proposed in this thesis allow arbitrary interactions among data from multiple sites, potentially from the whole image, without any added computational complexity.

Generative vs. Discriminative

Going back to the original aim of this work, we are interested in the classification of image sites. For classification purposes, we want to estimate the posterior over labels given the observations, i.e., $P(\boldsymbol{x}|\boldsymbol{y})$. In a generative framework, one expends efforts to model the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$, which involves implicit modeling of the observations via $p(\boldsymbol{y}|\boldsymbol{x})$. Usually, it is hard to model observations accurately, and one needs to make simplifying assumptions as discussed in the previous section. On the contrary, in a discriminative framework, one models the distribution $P(\boldsymbol{x}|\boldsymbol{y})$ directly. As noted in [34], a potential advantage of using the discriminative approach is that the true underlying generative model may be quite complex even though the class posterior is simple. This means that the generative approach may spend a lot of resources on modeling the generative models which are not particularly relevant to the task of inferring the class labels. Moreover, learning the class density models may become even harder when the training data is limited [125]. A more complete comparison between the discriminative and the generative models for the linear family of classifiers has been presented in [125][109].

In machine learning, the use of powerful probabilistic discriminative techniques is increasingly becoming common for data classification. Some examples of these techniques include simple logistic and probit classifiers and more advanced kernel classifiers such as relevance vector machine [135] and sparse classifier [38]. However, these techniques are applicable only to independently distributed data. On the other hand, as discussed before in this chapter, image data is usually not independently distributed. It contains significant contextual interactions at different levels. To incorporate these interactions using the existing graphical models, one is commonly forced to use only generative classifiers. In this thesis, we model the class conditional, $P(\boldsymbol{x}|\boldsymbol{y})$, directly as a Markov field as suggested by Lafferty et al. [86]. A crucial outcome of such models is that one can now use arbitrary discriminative classifiers even when data is not independently distributed.

Recently, there have been attempts to extend some of the popular discriminative methods such as AdaBoost [3], perceptron learning [43], and Support Vector Machines [5][134] to sequential labeling problems. However, one of the drawbacks of most of these techniques is that they develop models in a non-probabilistic setting. The reason for preferring probabilistic models is that they allow probabilistic interpretation of the outputs: in addition to predicting the best labels, one can also compute the posterior label probabilities. Yet another advantage of probabilistic models is that they allow the concept of marginalization, which is necessary to obtain optimal labels that minimize sitewise label errors. Recognizing the need of developing probabilistic discriminative models for the structured data, Altun et al. [4] have recently extended the use of Gaussian Processes (GP) to label sequences. The models presented in this thesis provide one possible strategy of exploiting arbitrary discriminative classifiers for structured data.

1.6.4 Our Approach

To summarize, the approach taken in this thesis differs from the previous efforts toward using context in that it

- derives context automatically from fully labeled training images instead of using auxiliary meta-data such as date and time of the image capture,
- avoids using rigid rule-based approach by means of statistical modeling of context,
- models context such that the labeling of different image components is done simultaneously instead of assigning labels in a sequential manner to avoid excessive dependence on previous mislabelings,
- avoids making simplistic assumptions such as conditional independence of the observed data by using discriminative models for classification instead of the generative ones,
- allows data-dependent interactions between labels by avoiding interpreting label interactions as *priors* under the Bayesian view,
- manages the exponential growth in computational complexity by leveraging efficient inference techniques based on network flow or message passing.

1.7 Thesis Contributions

Building upon the work on modeling context using undirected graphs, this thesis makes the following contributions:

- Introduces new probabilistic graphical models in computer vision that allow the use of local discriminative classifiers to incorporate contextual interactions among image components. In particular, this thesis introduces for the first time Conditional Random Field (CRF) [86] based models in computer vision.
- Develops models to capture complex spatial dependencies in labels as well as the observed data simultaneously in a principled manner on 2D lattices with cycles.
- Provides fast and robust parameter learning procedures which are applicable to even the conventional MRF models. In addition, this thesis gives an empirical comparison between different learning and inference techniques indicating coupling of learning and inference mechanisms.

- Proposes a hierarchical field formulation to model different types of contexts in images simultaneously within the same framework. The context may vary from short-range interactions between pixels to long-range interactions between objects or regions.
- Demonstrates the application of the proposed models on several challenging computer vision tasks such as contextual object detection, semantic image segmentation, texture recognition and image denoising seamlessly within a single framework.

1.8 Thesis Outline

This thesis is organized as follows:

Causal Models

In Chapter 2, we start with a discussion on how causal models are used in computer vision to learn spatial interactions in images. In particular, we focus on a popular tree-structured causal model known as Multi-Scale Random Field (MSRF). This chapter describes the formulation, parameter learning and inference in this model. At the end, we discuss several limitations of these models that prevent their use for modeling context at various levels in images. This leads to the exploration of noncausal models carried out in the next chapter.

Noncausal Models

Chapter 3 presents a noncausal discriminative field model that alleviates most of the limitations posed by the traditional MRFs. This chapter further explains the design of clique potentials, and two different methods for learning the parameters in these fields. This chapter lays the foundation of the formulations given in the rest of the thesis. Finally, it demonstrates the benefits of the discriminative fields on the applications of man-made structure detection and binary image denoising.

Despite the successes of the parameter learning procedures described in this chapter, automatic learning without any hand-tuned control knob remains a challenge in these fields. This is because exact maximum likelihood learning in these models is computationally intractable. So, the question arises: which approximation should we use to find the 'best' set of parameters? The answer to this question is explored in the next chapter.

Approximate Parameter Learning

In Chapter 4, we present an approach for approximate maximum likelihood parameter learning in discriminative field models, which is based on approximating true expectations with simple piecewise constant functions constructed using inference techniques. Gradient ascent with these updates exhibits compelling limit cycle behavior which is tied closely to the number of errors made during inference. The performance of various approximations is evaluated with different inference techniques showing that the learned parameters lead to good classification performance so long as the method used for approximating the gradient is consistent with the inference mechanism.

Multiclass Discriminative Fields

The basic formulation of discriminative fields in Chapter 3 was developed for binary image labeling examples. To deal with more complex real-world tasks, we present its extensions to multiclass labeling problems in Chapter 5. We motivate this discussion in the context of parts-based object detection. These fields allow simultaneous discriminative modeling of the appearance of individual parts as well as the geometric relations among them. The conventional MRF formulations cannot be used for this purpose because they do not allow the use of data while modeling interaction between labels, which is crucial for enforcing geometric consistencies between parts. This chapter demonstrates the efficacy of this formulation through controlled experiments on rigid and deformable synthetic toy objects.

Hierarchical Discriminative Fields

The discussion in the thesis so far concentrates on modeling interactions in images at a pixel, a block or a patch level. Chapter 6 presents a two-layer hierarchical formulation to exploit different levels of contextual information in images for robust classification. Each layer is modeled as a discriminative field. This model encodes both the short-range interactions (e.g., pixelwise label smoothing) as well as the long-range interactions (e.g., relative configurations of objects or regions) in a tractable manner. The parameters of the model are learned using a sequential maximum-likelihood approximation. The benefits of the proposed framework are demonstrated on four different datasets on the applications of pixelwise image labeling and contextual object detection.

Conclusions and Future Work

We present the conclusions derived from the theoretical and experimental observations from this thesis in Chapter 7. Then we describe several possibilities to enhance the power of the models presented in this thesis. Finally, we wrap up the thesis with a discussion of several open issues regarding classification problems in computer vision.

Chapter 2

Causal Models

2.1 Introduction

Causal models are global probability distributions defined on directed graphs using local transition probabilities. If a causal graph is acyclic[1], the joint distribution over the node variables can be written as,

$$P(x) = \prod_i P(x_i|pa_i),$$

where pa_i is the parent of node i. Causal models are usually seen as generative models that describe how the observed data i.e., images are generated. In this chapter we will primarily discuss hierarchical causal models, in which, nodes in the last layer of the hierarchy represent actual labels on the image sites. Further, it is assumed that these hierarchical models follow the Markov Property over scales. A particular form of such models is a causal tree (Figure 2.2 (b)) in which each node has only one parent. Causal trees contain no cycles and hence allow the use of very efficient techniques for exact parameter learning and inference. Such trees have been used under the name of Multi-Scale Random Field (MSRF)[2] [16] or Tree-Structured Belief Networks [34] in image segmentation and labeling. Using their work as a basis, in our preliminary research[3], we extended the causal trees to include interactions in the observed data

[1]The directed acyclic graphs are popularly known as *Bayesian Networks*.

[2]There is some controversy over whether a directed graph should be called a random field. In this thesis, we will maintain the same terminology as suggested by Bouman and Shapiro [16].

[3]An shorter version of this work appeared in IEEE International Conference on Computer Vision

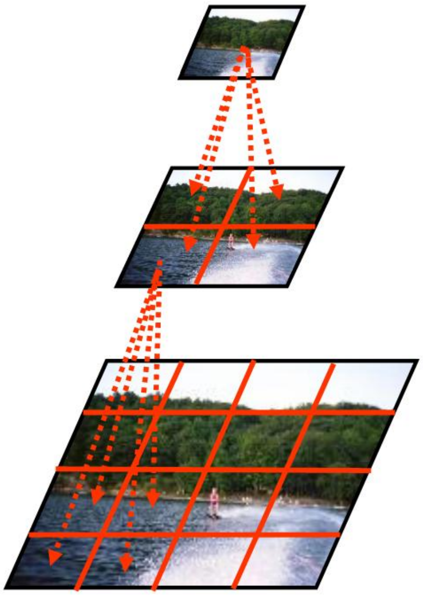


Figure 2.1: A quad-tree causal generative model of an image in which each layer has four times less nodes than the layer below. Here each node has one parent and four children except the root node which has no parent and the leaf nodes that have no children.

by using factored approximations [80] as described next.

2.2 Multi-Scale Random Field (MSRF)

Let $\boldsymbol{y}$ be the observed data associated with the input image, and $\boldsymbol{x}$ be the labels. In a MSRF model, the labels over an image are generated using Markov chains defined over coarse to fine scales. It can facilitate easy incorporation of long-range correlations in the image. In a quad-tree representation, a pyramid is built over the input image such that the number of nodes in each layer are reduced by four in comparison to the layer below as shown in Figure 2.1. Since the causal graph shown in Figure 2.1 is singly-connected, i.e., it does not have any loops, the Maximum A Posteriori (MAP) or the Maximum Posterior Marginal (MPM) inference in this graph is noniterative and the time complexity is linear in the number of image sites.

In this work we explored a slightly more complex model in which the MSRF is not a tree any more. For simplicity, a 1-D representation of the overall image generative

and Pattern Recognition (CVPR '03)[80].

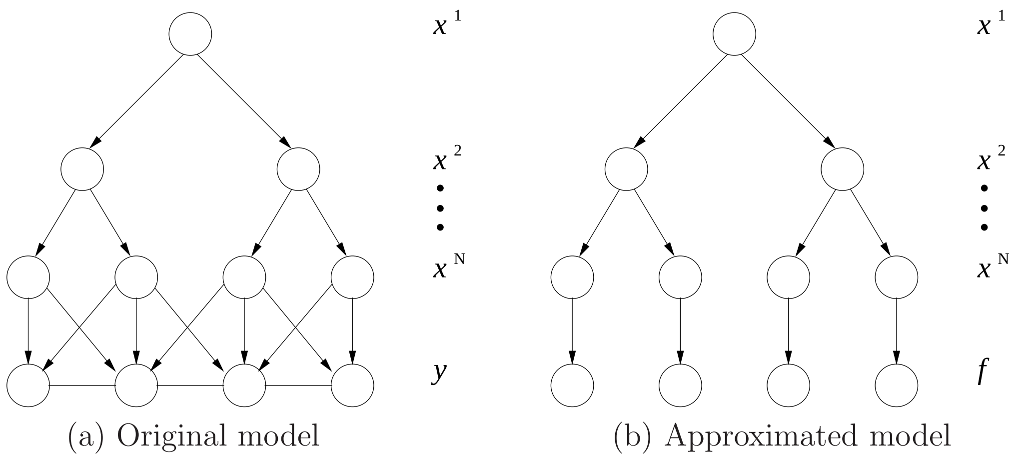


Figure 2.2: A 1-D representation of the MSRF based image generative model and its tree approximation. Note that the last layer of observed data y in the original model has been replaced by a multiscale feature vector layer f in the tree-approximated model.

model is given in Figure 2.2 (a). According to the overall image generative model, the image data $\boldsymbol{y}$ is generated from an underlying process $\boldsymbol{x}$, where $\boldsymbol{x}$ is a MSRF. The labels at N levels of the causal tree are denoted by $\boldsymbol{x} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ with $P(\boldsymbol{x}) = P(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N)$. Here $\boldsymbol{x}^n$ is the set of labels at all the nodes in level n. It can be noted that the observed image labels are nodes of the layer $\boldsymbol{x}^N$. In the Bayesian framework, given image $\boldsymbol{y}$, we are interested in finding the predictive posterior over the labels $\boldsymbol{x}^N$, which can be written as $P(\boldsymbol{x}^N|\boldsymbol{y}) \propto P(\boldsymbol{y}|\boldsymbol{x}^N)P(\boldsymbol{x}^N)$. Here $P(\boldsymbol{y}|\boldsymbol{x}^N)$ is the observation (or likelihood) model and $P(\boldsymbol{x}^N)$ is the prior model on the labels at level N.

In the MSRF model, the Markov assumption over scales implies

$$P(\boldsymbol{x}^n|\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{n-1}) = P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}) \text{ for } n = 2, \ldots, N.$$

Further, from the conditional independence assumption for the directed graphs,

$$P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}) = \prod_{i \in S^n} P(x_i^n|z_i^{n-1}),$$

where x_i^n is i^{th} node at level n, z_i^{n-1} is its parent at level $(n-1)$, and S^n is the set containing all the nodes at level n. Note that in the proposed MSRF, the observed data is *not* conditionally independent given the class labels. It interacts with the data at other sites for the purpose of classifying a certain site. To avoid dealing with an intractable true joint conditional $P(\boldsymbol{y}|\boldsymbol{x}^N)$ in this model, we assume a factored form of the observation model such that,

$$P(\boldsymbol{y}|\boldsymbol{x}^N) \approx \prod_{i \in S^N} P(y_i|y_{\mathcal{N}_i}, x_i^N), \tag{2.1}$$

where $\mathcal{N}_i$ is the neighborhood set of site i, and $y_{\mathcal{N}_i} = \{y_{i'}|i' \in \mathcal{N}_i\}$. The above approximation is similar to the pseudo-likelihood factorization in the MRF literature [94]. While making this approximation, we have ignored the fact that the observed data at each site depend on the labels of the neighboring sites as well. Thus, the overall generative model of the image can be expressed as,

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(x^1) \prod_{i \in S} P(x_i|z_i) \prod_{i \in S^N} P(y_i|y_{\mathcal{N}_i}, x_i^N), \tag{2.2}$$

where S is the set containing all the nodes in the tree $\boldsymbol{x}$ except the root node x^1. To simplify the notation, we have denoted a generic node at any level of the tree by x_i, and its parent by z_i.

We further assume the field over the data $\boldsymbol{y}$ to be homogeneous, and we approximate the conditional $P(y_i|y_{\mathcal{N}_i}, x_i^N)$ by $P(f_i|x_i^N)$, where f_i is a feature vector which encodes the dependencies of data at site i with its neighbors. This approximation allows us to model rich dependencies in the neighborhoods of a site directly through arbitrary features, which may otherwise be hard to model in $P(y_i|y_{\mathcal{N}_i}, x_i^N)$, as argued in [154]. In addition, such approximation becomes inevitable in the case of limited training data.

For our application, we need a rich representation of the data for man-made structure detection, which is inherently contained over multiple scales. In this work, we design f_i as a multiscale feature vector similar to the concept of parent vector defined by De Bonet [14], with the distinction that we compute features at a particular site by varying the size of the window around it so that the dependencies on the neighbors could be encoded explicitly. This kind of scale is also known as integration or artificial scale in the vision literature. Using the above assumptions, we can now approximate the overall image generative model as given in Figure 2.2 (b). Note that the original observation layer $\boldsymbol{y}$ has been replaced by a multiscale observation layer

$\boldsymbol{f}$. The approximated generative model is now a causal tree and the benefits of that model in terms of exact noniterative inference can now be reaped.

Finally, exploiting the assumption of homogeneity, the likelihood of the multiscale feature vector was modeled using a Gaussian Mixture Model (GMM) for each class as

$$P(f_m|x_m^N)=\sum_{\gamma=1}^{\Gamma} P(f_m|x_m^N,\gamma)P(\gamma|x_m^N), \tag{2.3}$$

where $P(f_m|x_m^N,\gamma)\sim\mathcal{N}(\mu_\gamma,\Sigma_\gamma)$, μ_γ is the mean and Σ_γ is the covariance of the γ^{th} Gaussian, and Γ is the total number of Gaussians in the GMM.

2.3 Parameter Estimation and Inference

The full image generative model has two different sets of parameters: Θ_p in the prior model, and Θ_o in the observation model. The observation model parameters consist of the mean vectors and the covariance matrices of the Gaussians, which are estimated through standard maximum likelihood formulation for GMM using Expectation Maximization (EM) [29][12]. The prior model parameter set consists of conditional transition probabilities over different links in the tree, and the prior probabilities over the root node. Let θ_{ikl} be the transition probability for node $i\in S$, defined as, $\theta_{ikl}=P(x_i=l|z_i=k)$, with the constraint $\sum_l \theta_{ikl}=1$, where $k,l\in\{0,1\}$. It simply defines the conditional distribution at i^{th} node in the MSRF given the label of its parent in the previous layer. The prior model parameters were learned using the Maximum Likelihood (ML) approach [34] by maximizing the probability of the labeled training images as,

$$\widehat{\Theta}_p^{ML}=\arg\max_{\Theta_p}\prod_{m=1}^{M} P(\boldsymbol{x}^{Nm},\boldsymbol{y}^m|\Theta_p,\Theta_o),$$

where m indexes over the training images, and M is the total number of training images. Assuming the observation model to be fixed, the ML estimate of Θ_p is simply obtained using the labeled images $\boldsymbol{x}^{Nm}$ as,

$$\widehat{\Theta}_p^{ML} = \arg\max_{\Theta_p} \prod_{m=1}^{M} P(\boldsymbol{x}^{Nm}|\Theta_p).$$

This maximization is carried out using EM, where all the nodes of MSRF from root to level $(N-1)$ are interpreted as the hidden variables. Denoting the hidden variables by $\boldsymbol{x}_h = \{\boldsymbol{x}\backslash\boldsymbol{x}^N\}$, in the E-step the lower bound is computed for the likelihood function at the current estimate of the parameters Θ'_p as the following expectation:

$$Q(\Theta_p, \Theta'_p) = \sum_{m=1}^{M} E_{\boldsymbol{x}_h^m|\Theta'_p}\left[\log P(\boldsymbol{x}_h^m, \boldsymbol{x}^{Nm}|\Theta_p)\right].$$

Computing the lower bound simply amounts to estimating the posterior probabilities over each parent-child pair,

$$P(x_i^m\!=\!l, z_i^m\!=\!k|\boldsymbol{x}^{Nm}, \Theta'_p) = \frac{\lambda(x_i^m\!=\!l)\ \theta'_{ikl}\ \pi(z_i^m\!=\!k)}{\sum_{k'} \pi(z_i^m\!=\!k')\lambda(z_i^m\!=\!k')} \prod_{u\in U(x_i^m)} \lambda_u(z_i^m\!=\!k), \tag{2.4}$$

where $U(x_i)$ is the set containing all the siblings of x_i, $\lambda(x_i)$ is the λ-value at node x_i, $\pi(z_i)$ is the π-value at node z_i, and $\lambda_u(z_i)$ is the λ-message sent from node u to z_i. Here, $\lambda(.)$ messages are defined as,

$$\lambda(x_i) = \prod_{v\in V(x_i)} \lambda_v(x_i), \tag{2.5}$$

$$\lambda_v(x_i) = \sum_k P(v = k|x_i)\lambda(v), \tag{2.6}$$

where $V(x_i)$ is the set containing all the children of the node x_i. Similarly the $\pi(.)$ messages are defined as,

$$\pi(x_i) = \sum_k P(x_i|z_i = k)\pi_{x_i}(z_i = k), \tag{2.7}$$

$$\pi_{x_i}(z_i) = \gamma\ \pi(z_i) \prod_{u\in U(x_i)} \lambda(u_z), \tag{2.8}$$

where $U(x_i)$ is the set containing all the siblings of x_i, and γ is a normalizing factor so that the values of $\pi_{x_i}(z_i)$ sum to one. More detailed derivations of these terms can be found in the seminal book on belief propagation by Pearl [113] in the context of singly-connected causal trees.

In the M-step, the new parameter values are obtained by maximizing the bound. The update of parameters for each link in the tree can be obtained as,

$$\theta_{ikl} = \frac{\sum_{m=1}^{M} P(x_i^m = l, z_i^m = k | \boldsymbol{x}^{Nm}, \Theta'_p)}{\sum_{m=1}^{M} \sum_{l'} P(x_i^m = l', z_i^m = k | \boldsymbol{x}^{Nm}, \Theta'_p)}. \tag{2.9}$$

However, in the case of limited training data, computing a different θ_{ikl} for each link is not practical. Thus, all the θ_{ikl} at each level n were forced to be the same as suggested in [34], and denoted as θ_{nkl}. Maximizing the bound defined above, subject to the constraint $\sum_l \theta_{nkl}=1$ yields for level n,

$$\theta_{nkl} = \frac{\sum_{m=1}^{M} \sum_{x_i \in S^n} P(x_i^m = l, z_i^m = k | \boldsymbol{x}^{Nm}, \Theta'_p)}{\sum_{m=1}^{M} \sum_{x_i \in S^n} \sum_{l'} P(x_i^m = l', z_i^m = k | \boldsymbol{x}^{Nm}, \Theta'_p)}. \tag{2.10}$$

The prior probabilities over the root node are simply given by the belief at that node obtained through λ-π message passing scheme of Pearl [113].

Given a new test image $\boldsymbol{y}$, the problem of inference is to find the optimal class labels over the image sites where the optimality is evaluated with respect to a particular cost function. The MAP and the MPM estimates of the labels can be obtained using the max-product and sum-product versions of the belief propagation respectively [113]. For further details on the MSRF based causal models and their experimental evaluation, we point the reader to Appendix B. Now we discuss some of the main problems associated with these models which led us to explore newer models that form the basis of this thesis.

2.4 Discussion

Even though the tree-structured models provide the advantage of exact parameter learning and inference using very efficient techniques, there are several problems that undermine their applicability to generic image analysis tasks:

1. The main problem with the fixed tree-structured models is that they suffer from the non-stationarity of the induced random field, leading to 'blocky' smoothing of the image labels [34]. This point is illustrated in Figure 2.2 (b), where in layer x^N, the first and the second node from the left have the same parents while the third node is connected to the second through a grandparent. This causes an imposed difference in the behavior of interactions between neighboring nodes at different places dictated by the tree structure even though there is no a priori reason for such a difference. This problem exists in all the tree-structured models whether causal or noncausal. One way to solve this problem is to dynamically adapt the tree-structure to a given input image. This idea was explored in *dynamic trees* [149] but the inference over tree-structure still remains an intractable problem.

2. When trained discriminatively, the causal models sometimes suffer from the *label bias* problem which unfairly favors labels with fewer successors due to the need of normalizing each link to be a proper transition probability [15][86]. On the other hand, in noncausal models this problem does not arise as one needs to define potential functions for each clique (which are not required to sum to one) and there is a universal normalizing constant for the whole distribution known as the partition function.

3. Crude approximations are usually required to make the data generative model computationally tractable to partially retain its expressive power.

4. There is no natural extension of the hierarchical tree models to more generic tasks such as object detection. In parts-based object detection problems, interactions among various parts are governed by the geometric (and possibly photometric) relationship of parts rather than some smoothing prior over them. In addition, it is also not clear how one can model the hierarchical structure for the irregular neighborhoods defined over random patches in the scene.

On a more philosophical side, it is not clear what the nodes in the various hidden layers of a hierarchical causal model really represent. The actual observed image does not have any explicit notion of label hierarchy. If it is purely a mathematical concept, the nodes in the hidden layers need not be restricted to the same cardinality as the actual label nodes. This opens several interesting issues about the selection of cardinality for the nodes at each hidden layer.

Based on the above limitations of the causal model, in this thesis we focus on noncausal models (undirected graphs) to model different types of context in images as discussed in the following chapters.

Chapter 3

Noncausal Models

3.1 Introduction

Noncausal models are global probability distributions defined on undirected graphs using local clique potentials, i.e.,

$$P(x) \propto \prod_{c \in \mathcal{C}} \psi_c(x_c), \tag{3.1}$$

where $\mathcal{C}$ is the set of all the cliques[1] in the graph, and $\psi_c(x_c)$ are clique potentials i.e., positive functions of clique variables x_c. Noncausal graphs are more suited to handle interactions over image lattices since usually there exists no natural causal relationships among image components. For instance, both keyboard and mouse mutually reinforce the chances of their occurrence in an image rather than any one of them causing the presence of the other.

One important issue that needs to be addressed while using general noncausal graphs is computational tractability in these models. One possible choice of making the computations efficient is to use tree-structured noncausal models, where parameter learning and inference can be done using efficient techniques since such graphs do not contain loops. Such models have recently been explored for image segmentation and restoration applications [61][28]. However, the tree-structured noncausal models suffer from similar problems as the causal trees described in Section 2.4 except the label-bias problem. So, in the following discussion we will explore models based on

[1] A clique is a fully connected subgraph of the original graph

arbitrary undirected graphs with loops.

Markov Random Fields (MRFs) are commonly used undirected models in computer vision. As discussed in Section 1.6.3, MRFs are modeled in a generative framework, which requires simplifying assumptions precluding the use of arbitrarily complex dependencies in the observed data that might be desired for the purpose of classification. In addition, for several vision applications, e.g. object detection, the interaction in labels is based on the observed relational data between different sites. But the traditional MRF formulation does not allow any use of data in label interactions. In MRF formulations of binary classification problems, the label interaction field, $P(\boldsymbol{x})$, is commonly assumed to be a homogeneous and isotropic MRF such as Ising model (or Potts model for multiclass labeling problems) with only pairwise nonzero potentials. If the data likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ is assumed to be conditionally independent given the labels, the posterior distribution[2] over the labels can be written as,

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z_m} \exp\left(\sum_{i \in S} \log p(\boldsymbol{s}_i(\boldsymbol{y_i})|x_i) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \beta_m x_i x_j\right), \tag{3.2}$$

where β_m is the interaction parameter of the MRF, and $\boldsymbol{s}_i(\boldsymbol{y_i})$ is the data (feature vector) at site i. Note that even though only the label prior, $P(x)$ was assumed to be a MRF, the assumption of the conditional independence of the data implies that the posterior given in Eq. (3.2) is also a MRF. This allows one to reap the benefits of readily available tools of inference over a MRF. If the conditional independence assumption is not used, the posterior will usually not be a MRF making the inference difficult. As discussed in Chapter 1, the assumption of conditional independence is usually too restrictive for many application in computer vision.

For classification purposes, we want to estimate the conditional distribution over labels given the observations, i.e., $P(\boldsymbol{x}|\boldsymbol{y})$. As explained in section 1.6.3, in a discriminative framework, one models the distribution $P(\boldsymbol{x}|\boldsymbol{y})$ directly, unlike the generative models (e.g., conventional MRFs) where one obtains this distribution indirectly by modeling the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$. This saves one from making simplistic assumptions about the data. This view forms the core theme of the models we propose in this thesis as discussed in the following sections.

[2] With a slight abuse of notation, we will use the term 'traditional MRF model' to indicate this posterior in the rest of the document.

3.2 Discriminative Random Field (DRF)

In this thesis, we present Discriminative Random Fields (DRFs)[3] based on the concept of Conditional Random Field (CRF) proposed by Lafferty et al. [86] in the context of segmentation and labeling of 1-D text sequences. The CRFs are discriminative models that directly model the conditional distribution over labels i.e., $P(\boldsymbol{x}|\boldsymbol{y})$ as a Markov Random Field. This approach allows one to capture arbitrary dependencies between the observations without resorting to any model approximations. CRFs have been shown to outperform the traditional Hidden Markov Model based labeling of text sequences [86]. Our model further enhances the 1-D CRFs described in [86] by proposing the use of local discriminative models to capture the class associations at individual sites as well as the interactions among the neighboring sites on 2-D regular as well as irregular image lattices.

We first restate in our notations the definition of CRFs as given by Lafferty et al. [86]. Let the observed data from an input image be given by $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$ where $\boldsymbol{y}_i$ is the data from i^{th} site and $\boldsymbol{y}_i \in \Re^c$. The corresponding labels at the image sites are given by $\boldsymbol{x} = \{x_i\}_{i \in S}$. In this chapter, we will be restrict ourselves to binary classification problems, i.e. $x_i \in \{-1, 1\}$. In Chapter 5, we will extend the framework to multiclass labeling problems. The random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly distributed, but in a discriminative framework, a conditional model $P(\boldsymbol{x}|\boldsymbol{y})$ is constructed from the observations and labels, and the marginal $p(\boldsymbol{y})$ is not modeled explicitly.

Definition 1. ***CRF**: Let $G = (S, E)$ be a graph such that $\boldsymbol{x}$ is indexed by the vertices of G. Then $(\boldsymbol{x}, \boldsymbol{y})$ is said to be a conditional random field if, when conditioned on $\boldsymbol{y}$, the random variables x_i obey the Markov property with respect to the graph: $P(x_i|\boldsymbol{y}, \boldsymbol{x}_{S-\{i\}}) = P(x_i|\boldsymbol{y}, \boldsymbol{x}_{\mathcal{N}_i})$, where $S - \{i\}$ is the set of all the nodes in the graph except the node i, $\mathcal{N}_i$ is the set of neighbors of the node i in G, and $\boldsymbol{x}_\Omega$ represents the set of labels at the nodes in set Ω.*

Thus, a CRF is a random field globally conditioned on the observations $\boldsymbol{y}$. The condition of positivity requiring,

$$P(\boldsymbol{x}|\boldsymbol{y}) > 0 \quad \forall \;\; \boldsymbol{x}$$

[3] An earlier version of this work appeared in International Conference on Computer Vision (ICCV '03)[79].

has been assumed implicitly. Now, using the Hammersley-Clifford theorem [53] and assuming only up to pairwise clique potentials to be nonzero, the conditional distribution over all the labels $\boldsymbol{x}$ given the observations $\boldsymbol{y}$ in a CRF can be written as,

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A_i(x_i, \boldsymbol{y}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I_{ij}(x_i, x_j, \boldsymbol{y}) \right), \tag{3.3}$$

where Z is a normalizing constant known as the partition function, and -A_i and -I_{ij} are the unary and pairwise potentials respectively. With a slight abuse of notations, in the rest of this document we will call A_i the *association potential* and I_{ij} the *interaction potential*.

There are two main differences between the conditional model given in Eq. (3.3) and the traditional MRF framework given in Eq. (3.2). First, in the conditional fields, the association potential at any site is a function of all the observations $\boldsymbol{y}$ while in MRFs (with the assumption of conditional independence of the data), the association potential is a function of data only at that site, i,e., $\boldsymbol{y}_i$. Second, the interaction potential for each pair of nodes in MRFs is a function of only labels, while in the conditional models it is a function of labels as well as all the observations $\boldsymbol{y}$. As will be shown later, these differences play a crucial role in modeling arbitrary interactions in both observed data and labels in natural images in a principled manner.

The DRF model we present in this thesis is a specific type of CRF defined in Eq. (3.3), and thus inherits all its advantages over the traditional MRF as described above. In the DRF model, we extend the specific 1-D sequential CRF form proposed in [86]. There are two main extensions: First, the unary and pairwise potentials in DRFs are designed using arbitrary local discriminative classifiers. This allows one to use domain-specific discriminative classifiers for structured data rather than restricting the potentials to a specific form. Taking a similar view, several researchers have recently demonstrated good results using different classifiers such as probit [119], boosting [136] and even neural network [56]. This view is consistent with one of the key motivations behind this work in which we wanted to develop models that allow one to leverage the power of discriminative classifiers in problems where data has interactions rather than being independent. Second, instead of being 1-D sequential models, the DRFs are defined over 2-D image lattices which generally induce graphs with loops. This makes the problem of parameter learning and inference significantly

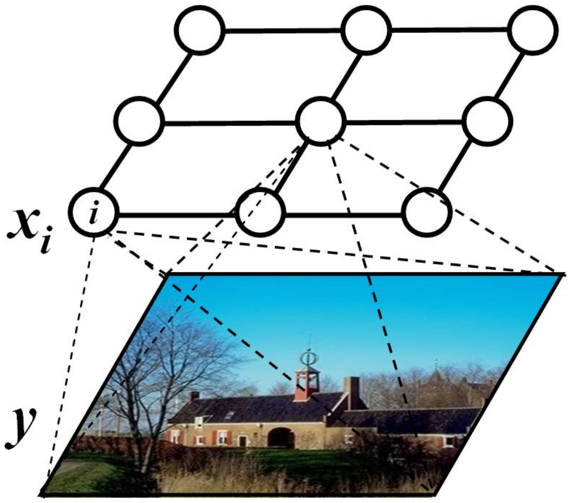


Figure 3.1: An illustration of a typical DRF for an example task of man-made structure detection in natural images. The aim is to label each site i.e., each 16×16 image block whether it is a man-made structure or not. The top layer represents the labels on all the image sites. Note that each site i can potentially use features from the whole image $\boldsymbol{y}$ unlike the traditional MRFs.

harder. To the best of our knowledge, ours is the first work that introduced CRF-based models in computer vision for image analysis. Recently, a number of researchers have demonstrated the utility of such models in various computer vision applications [108][56] [136][120][147][133][119][145].

In the rest of the thesis, we assume the random field given in Eq. (3.3) to be homogeneous i.e., the functional forms of A_i and I_{ij} are independent of the location i. In addition, we also assume the field to be isotropic implying that the label interactions are non-directional. In other words, I_{ij} is independent of the relative locations of sites i and j. Thus, subsequently we will drop the subscripts and simply use the notations A and I to denote the two potentials. Note that the assumption of isotropy can be easily relaxed at the cost of a few additional parameters. Thus, in this thesis, we will consider the models of the following form:

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A(x_i, \boldsymbol{y}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y}) \right). \quad (3.4)$$

This form of discriminative field makes it possible to treat different applications from low-level image denoising to high-level contextual object detection tasks in a single framework in a seamless fashion. Figure 3.1 illustrates a typical DRF for an example image analysis task of man-made structure detection[4]. Suppose, we are given an input image $\boldsymbol{y}$ shown in the bottom layer and we are interested in labeling each image site (in this case a 16×16 image block) based on whether it contains a man-made structure or not. The top layer represents the labels $\boldsymbol{x}$ on all the image sites. Note that each site i can potentially use features from the whole image $\boldsymbol{y}$ unlike the traditional MRFs. In addition, DRFs allow to use image data to model interactions between two neighboring sites i and j. In the following sections we discuss how the unary and the pairwise potentials are designed in DRFs.

3.2.1 Association Potential

In the DRF framework, the association potential, $A(x_i, \boldsymbol{y})$, can be seen as a measure of how likely a site i will take label x_i given image $\boldsymbol{y}$, ignoring the effects of other sites in the image (Figure 3.2). Suppose, $\boldsymbol{f}(.)$ is a function that maps an arbitrary patch in an image to a feature vector such that $\boldsymbol{f} : \mathcal{Y}_p \rightarrow \Re^l$. Here $\mathcal{Y}_p$ is the set of all possible patches in all possible images. Let $\omega_i(\boldsymbol{y})$ be an arbitrary patch in the neighborhood of site i in image $\boldsymbol{y}$ from which we want to extract a feature vector $\boldsymbol{f}(\omega_i(\boldsymbol{y}))$. Note that the neighborhood used for the patch $\omega_i(\boldsymbol{y})$ need not be the same as the label neighborhood $\mathcal{N}_i$. Indeed, $\omega_i(\boldsymbol{y})$ can potentially be the whole image itself. For clarity, with slight abuse of notation, we will denote the feature vector $\boldsymbol{f}(\omega_i(\boldsymbol{y}))$ at each site i by $\boldsymbol{f}_i(\boldsymbol{y})$. The subscript i indicates the difference just in the feature vectors at different sites, *not* in the functional form of $\boldsymbol{f}(.)$. Then, $A(x_i, \boldsymbol{y})$ is modeled using a local discriminative model that outputs the association of the site i with class x_i as,

$$A(x_i, \boldsymbol{y}) = \log P'(x_i | \boldsymbol{f}_i(\boldsymbol{y})), \tag{3.5}$$

where $P'(x_i | \boldsymbol{f}_i(\boldsymbol{y}))$ is the local class conditional at site i. This form allows one to use an arbitrary domain-specific probabilistic discriminative classifier for a given task. This can be seen as a parallel to the traditional MRF models where one can use arbitrary local generative classifier to model the unary potential. One possible choice of $P'(.)$ can be Generalized Linear Models (GLM), which are used extensively in

[4]More details on this application are given in Appendix A

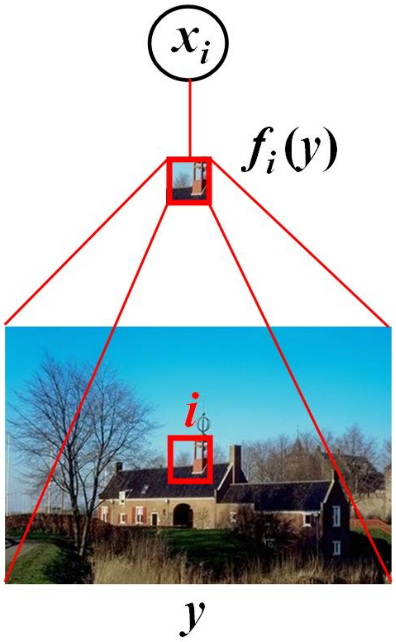


Figure 3.2: Given a feature vector $\boldsymbol{f}_i(\boldsymbol{y})$ at site i, the association potential in DRFs can be seen as a measure of how likely the site i will take label x_i, ignoring the effects of other sites in the image. Note that the feature vector $\boldsymbol{f}_i(\boldsymbol{y})$ can be constructed by pooling arbitrarily complex dependencies in the observed data y.

statistics to model the class posteriors given the observations [102]. In this work we used the logistic function[5] as a *link* in the GLM. Thus, the local class conditional can be written as,

$$P'(x_i{=}1|\boldsymbol{f}_i(\boldsymbol{y})) = \frac{1}{1+e^{-(w_0+\boldsymbol{w}_1^T\boldsymbol{f}_i(\boldsymbol{y}))}} = \sigma(w_0+\boldsymbol{w}_1^T\boldsymbol{f}_i(\boldsymbol{y})), \tag{3.6}$$

where $\boldsymbol{w} = \{w_0, \boldsymbol{w}_1\}$ are the model parameters. This form of $P'(.)$ will yield a linear decision boundary in the feature space spanned by vectors $\boldsymbol{f}_i(\boldsymbol{y})$. To extend the logistic model to induce a nonlinear decision boundary, a transformed feature vector at each site i is defined as $\boldsymbol{h}_i(\boldsymbol{y}) = [1, \phi_1(\boldsymbol{f}_i(\boldsymbol{y})), \ldots, \phi_R(\boldsymbol{f}_i(\boldsymbol{y}))]^T$ where $\phi_k(.)$ are arbitrary nonlinear functions. These functions can be seen as kernel mapping of the original feature vector into a high dimensional space. The first element of the transformed vector is kept as 1 to accommodate the bias parameter w_0. Further, since $x_i \in \{-1, 1\}$, the probability in Eq. (3.6) can be compactly expressed as,

[5]One can use other choices of link such as *probit* link.

$$P'(x_i|\boldsymbol{y}) = \sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y})). \tag{3.7}$$

Finally, for this choice of $P'(.)$, the association potential can be written as,

$$A(x_i, \boldsymbol{y}) = \log(\sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}))) \tag{3.8}$$

This transformation ensures that the DRF is equivalent to a logistic classifier if the interaction potential in Eq. (3.3) is set to zero. Note that the use of logistic function to model the discriminative classifier yields $A(.)$ that is linear in features. This is similar to the original form of the 1-D sequential CRFs of [86] with the difference that we use kernels to define this potential. Parallel to our work, researchers have proposed the use of kernels in CRF-type of models [87] [134]. Moreover, while designing graph potentials, recently other researchers have explored the use of different classifiers such as probit classifier [119][133] which will not yield a linear form of $A_i(.)$. Similarly, in Boosted Random Fields (BRFs) proposed by Torralba et at. [136], the authors design unary potential using boosting. They show good results on the application of contextual object detection using BRFs.

Note that in Eq. (3.8), the transformed feature vector at *each* site i i.e., $\boldsymbol{h}_i(\boldsymbol{y})$ is a function of the whole set of observations $\boldsymbol{y}$. This allows one to pool arbitrarily complex dependencies in the observed data for the purpose of classification. On the contrary, the assumption of conditional independence of the data in the traditional MRF framework allows one to use the data only from a particular site, i.e., $\boldsymbol{y}_i$ to get the log-likelihood, which acts as the association potential as shown in Eq. (3.2).

In related work, a neural network based discriminative classifier was used by Feng et al. [34] to model the observations in a generative tree-structured belief network model. Since the model required generative data likelihood, the discriminative output of the neural network was used to approximate the actual likelihood of the data in an ad-hoc fashion. On the contrary, in the DRF model, the discriminative class posterior is an integral part of the full conditional model in Eq. (3.4), and all the models parameters are learned simultaneously.

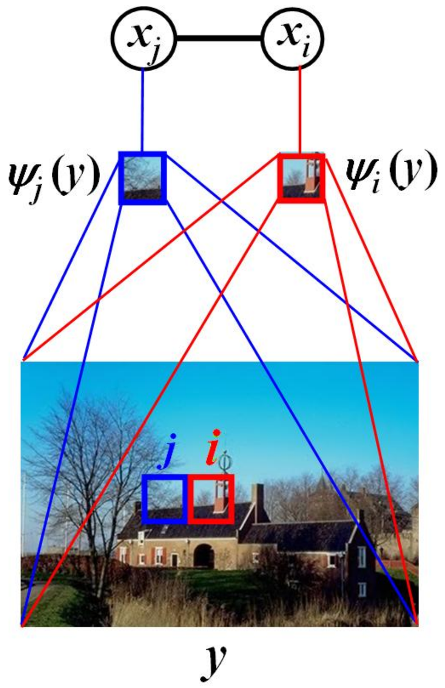


Figure 3.3: Given feature vectors $\boldsymbol{\psi}_i(\boldsymbol{y})$ and $\boldsymbol{\psi}_j(\boldsymbol{y})$ at two neighboring sites i and j respectively, the interaction potential can be seen as a measure of how the labels at sites i and j influence each other. Note that such interaction in labels is dependent on the observed image data y, unlike the traditional generative MRFs.

3.2.2 Interaction Potential

In the DRF framework, the interaction potential can be seen as a measure of how the labels at neighboring sites i and j should interact given the observed image y (Figure 3.3). To model the interaction potential, $I(.)$, we first analyze a form commonly used in the MRF framework. For the isotropic, homogeneous Ising model, the interaction potential is given as $I(.) = \beta x_i x_j$, which penalizes every dissimilar pair of labels by the cost β [64]. This form of interaction favors piecewise constant smoothing of the labels without considering the discontinuities in the observed data explicitly. Geman and Geman [49] have proposed a line-process model which allows discontinuities in the labels through piecewise continuous smoothing. Other discontinuity models have also been proposed for adaptive smoothing [94], but all of them require the labels to be either continuous or ordered. On the contrary, in the classification task there is no natural ordering in the labels. Also, these discontinuity adaptive models do not use the observed data to model the discontinuities, which may be important for several applications [17][13].

In contrast, in the DRF formulation, the interaction potential is a function of all the observations $\boldsymbol{y}$. In our preliminary work, we proposed to model $I(.)$ in DRFs using a data-dependent term along with the constant smoothing term of the Ising model. In addition to modeling arbitrary pairwise relational information between sites, the data-dependent smoothing can compensate for the errors in modeling the association potential. To model the data-dependent smoothing term, the aim is to have similar labels at a pair of sites for which the observed data supports such a hypothesis. In other words, we are interested in learning a pairwise discriminative model.

Suppose, $\boldsymbol{\psi}(.)$ is a function that maps an arbitrary patch in an image to a feature vector such that $\boldsymbol{\psi} : \mathcal{Y}_p \rightarrow \Re^{\gamma}$. Let $\Omega_i(\boldsymbol{y})$ be an arbitrary patch in the neighborhood of site i in image $\boldsymbol{y}$ from which we want to extract a feature vector $\boldsymbol{\psi}(\Omega_i(\boldsymbol{y}))$. Note that the neighborhood used for the patch $\Omega_i(\boldsymbol{y})$ need not be the same as the label neighborhood $\mathcal{N}_i$. For clarity, with slight abuse of notation, we will denote the feature vector $\boldsymbol{\psi}(\Omega_i(\boldsymbol{y}))$ at each site i by $\boldsymbol{\psi}_i(\boldsymbol{y})$. Similarly, we define a feature vector $\boldsymbol{\psi}_j(\boldsymbol{y})$ for site j. Again, to emphasize, the subscripts i and j indicate the difference just in the feature vectors at different sites, *not* in the functional form of $\boldsymbol{\psi}(.)$. Given the features at two different sites, we want to learn a pairwise discriminative model $P''(x_i\!=\!x_j|\boldsymbol{\psi}_i(\boldsymbol{y}), \boldsymbol{\psi}_j(\boldsymbol{y}))$. Note that by choosing the function $\boldsymbol{\psi}_i$ to be different from $\boldsymbol{f}_i$, used in Eq. (3.6), information different from $\boldsymbol{f}_i$ can be used to model the relations between pairs of sites.

Let t_{ij} be an auxiliary variable defined as

$$t_{ij} = x_i x_j,$$

and let $\boldsymbol{\mu}(\boldsymbol{\psi}_i(\boldsymbol{y}), \boldsymbol{\psi}_j(\boldsymbol{y}))$ be a new feature vector such that $\boldsymbol{\mu}\!:\!\Re^{\gamma}\!\times\!\Re^{\gamma} \rightarrow \Re^{q}$. Denoting this feature vector as $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ for simplification, we model the pairwise discriminatory term similar to the one defined in Eq. (3.7) as,

$$P''(t_{ij}|\boldsymbol{\psi}_i(\boldsymbol{y}), \boldsymbol{\psi}_j(\boldsymbol{y})) = \sigma(t_{ij}\boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y})), \tag{3.9}$$

where $\boldsymbol{v}$ are the model parameters. Note that the first component of $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is fixed to be 1 to accommodate the bias parameter. Now, the interaction potential in DRFs is modeled as a convex combination of two terms, i.e.

$$I(x_i, x_j, \boldsymbol{y}) = \beta \left(K x_i x_j + (1-K)(2\sigma(t_{ij}\boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y})) - 1) \right) \tag{3.10}$$

where $0 \leq K \leq 1$. The first term is a data-independent smoothing term, similar to the Ising model. The second term is a $[-1, 1]$ mapping of the pairwise logistic function defined in Eq. (3.9). This mapping ensures that both terms have the same range. Ideally, the data-dependent term will act as a discontinuity adaptive model that will moderate smoothing when the data from two sites is 'different'. The parameter K gives flexibility to the model by allowing the learning algorithm to adjust the relative contributions of these two terms according to the training data. Finally, β is the interaction coefficient that controls the degree of smoothing. Large values of β encourage smoother solutions. Note that even though the model seems to have some resemblance to the line process suggested in [49], K in Eq. (3.10) is a global weighting parameter unlike the line process where a discrete parameter is introduced for each pair of sites to facilitate discontinuities in smoothing. Anisotropy can be easily included in the DRF model by parameterizing the interaction potentials of different directional pairwise cliques with different sets of parameters $\{\beta, K, \boldsymbol{v}\}$.

To summarize the roles of the two potentials in DRFs, the association potential acts as a complex nonlinear classifier for individual sites, while the interaction potential can be seen as a data-dependent discriminative label interaction.

3.2.3 Parameter Estimation

Let θ be the set of parameters of the DRF model where $\theta = \{\boldsymbol{w}, \boldsymbol{v}, \beta, K\}$. The form of the DRF model resembles the posterior for the traditional MRF framework given in Eq. (3.2). However, in the MRF framework, the parameters of the class generative models, $p(\boldsymbol{y}_i|x_i)$ and the parameters of the prior random field on labels, $P(\boldsymbol{x})$ are generally assumed to be independent and are learned separately [94]. In contrast, we make no such assumption and learn all the parameters of the DRF model simultaneously. Nevertheless, the similarity of the forms allows for most of the techniques used for learning the MRF parameters to be utilized for learning the DRF parameters with a few modifications.

We take the standard maximum-likelihood approach to learn the DRF parameters, which involves the evaluation of the partition function Z given as,

$$Z = \sum_{\boldsymbol{x}} \exp\left(\sum_i A(x_i, \boldsymbol{y}) + \sum_i \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y})\right). \tag{3.11}$$

Since there are exponential number of label configurations in the label space $\{\boldsymbol{x}\}$, the evaluation of the sum over $\boldsymbol{x}$ in Z is a NP-hard problem. The brute-force computation is intractable even for a moderate size graph. In principle, one can use either sampling techniques or resort to some approximations e.g. mean-field or pseudo-likelihood to estimate the parameters [94]. In the first set of experiments, we used the pseudo-likelihood formulation due to its simplicity. We will discuss some more advanced techniques of maximum likelihood parameter learning in DRFs in Chapter 4. According to the pseudo-likelihood approach, the parameters are estimated by maximizing the pseudo-likelihood instead of the true likelihood as,

$$\widehat{\theta}^{ML} \approx \arg\max_{\theta} \prod_{m=1}^{M} \prod_{i \in S} P(x_i^m | \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \theta), \tag{3.12}$$
$$Subject \;\; to \;\; 0 \leq K \leq 1$$

where m indexes over the training images and M is the total number of training images, and

$$P(x_i | \boldsymbol{x}_{\mathcal{N}_i}, \boldsymbol{y}, \theta) = \frac{1}{z_i} \exp\{A(x_i, \boldsymbol{y}) + \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y})\},$$

$$\text{where} \quad z_i = \sum_{x_i \in \{-1,1\}} \exp\left(A(x_i, \boldsymbol{y}) + \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y})\right).$$

The pseudo-likelihood given in Eq. (3.12) can be maximized by using line search methods for constrained maximization with bounds [50]. Since the pseudo-likelihood in Eq. (3.12) is not a convex function of the parameters, good initialization of the parameters is important to avoid bad local maxima. To initialize the parameters $\boldsymbol{w}$ in $A(x_i, \boldsymbol{y})$, we first learn these parameters using standard maximum likelihood logistic regression assuming all the labels x_i^m to be independent given the data $\boldsymbol{y}^{\boldsymbol{m}}$ for each image m [105]. Using Eq. (3.7), the log-likelihood can be expressed as,

$$L(\boldsymbol{w}) = \sum_{m=1}^{M} \sum_{i \in S} \log(\sigma(x_i^m \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}^{\boldsymbol{m}}))). \tag{3.13}$$

The Hessian of the log-likelihood is given as,

$$\nabla^2_{\boldsymbol{w}} L(\boldsymbol{w}) = -\sum_{m=1}^{M} \sum_{i \in S} \left\{ \sigma(\boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}^{\boldsymbol{m}}))(1 - \sigma(\boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}^{\boldsymbol{m}}))) \right\} \boldsymbol{h}_i(\boldsymbol{y}^{\boldsymbol{m}}) \boldsymbol{h}_i^T(\boldsymbol{y}^{\boldsymbol{m}}).$$

Note that the Hessian does not depend on how the data is labeled and is non-positive definite. Hence the log-likelihood in Eq. (3.13) is convex (convex downward or concave), and any local maximum is the global maximum. Newton's method was used for maximization which has been shown to be much faster than other techniques for correlated features [105]. The initial estimates of the parameters $\boldsymbol{v}$ in the data-dependent term in $I(x_i, x_j, \boldsymbol{y})$ were obtained similarly.

3.2.4 Inference

Given a new test image $\boldsymbol{y}$, our aim is to find the optimal label configuration $\boldsymbol{x}$ over the image sites where optimality is defined with respect to a given cost function. Maximum A Posteriori (MAP) solution is a widely used estimate that is optimal with respect to the zero-one cost function defined as,

$$C(\boldsymbol{x}, \boldsymbol{x}^*) = 1 - \delta(\boldsymbol{x} - \boldsymbol{x}^*), \tag{3.14}$$

where $\boldsymbol{x}^*$ is the true label configuration, and $\delta(\boldsymbol{x} - \boldsymbol{x}^*)$ is 1 if $\boldsymbol{x} = \boldsymbol{x}^*$, and 0 otherwise. For binary classifications, the MAP estimate can be computed exactly for an undirected graph using the max-flow/min-cut type of algorithms if the probability distribution meets certain conditions [41][51][73]. For the DRF model, exact MAP solution can be computed if $K \geq 0.5$ and $\beta \geq 0$. However, in the context of MRFs, the MAP solution has been shown to perform poorly for the Ising model when the interaction parameter, β takes large values [51][42]. Our results in Section 3.3.2 corroborate this observation for the DRFs too.

An alternative to the MAP solution is the Maximum Posterior Marginal (MPM) solution which is optimal for the sitewise zero-one cost function defined as,

$$C(\boldsymbol{x}, \boldsymbol{x}^*) = \sum_{i \in S} (1 - \delta(x_i - x_i^*)), \tag{3.15}$$

where x_i^* is the true label at the i^{th} site. The MPM computation requires marginalization over a large number of variables which is generally NP-hard. One can use either sampling procedures [42] or Belief Propagation to obtain an estimate of the MPM solution. In our preliminary work we obtained local MAP estimates using the algorithm Iterated Conditional Modes (ICM), proposed by Besag [11] which is equivalent to zero-temperature simulated annealing. Given an initial label configuration, ICM maximizes the local conditional probabilities iteratively, i.e.

$$x_i \leftarrow \arg\max_{x_i} P(x_i | \boldsymbol{x}_{\mathcal{N}_i}, \boldsymbol{y}). \tag{3.16}$$

ICM yields local maximum of the posterior and has been shown to give reasonably good results even when exact MAP performs poorly for large values of β [51][42]. As described by Besag [11], ICM is guaranteed to converge if the updates are carried out asynchronously. However, directional effects may arise due to the selection of a fixed sequence in which the sites are updated. The spurious directional effects can be eliminated by implementing a synchronous update procedure where all the sites are updated in parallel. But convergence can no longer be guaranteed in this case and small oscillations may occur. In our ICM implementation we use a partially synchronous scheme, in which coding sets of pixels are simultaneously updated [11]. A coding sets is a set of image pixels such that each pixel in this set is a non-neighbor of any other pixel in the set. This type of update provides a useful compromise between the synchronous and the asynchronous schemes [11].

3.3 Man-made Structure Detection Task

We evaluated the performance of the proposed DRF model on the task of detecting man-made structures in natural scenes. This is a difficult task because there are significant within class variations in the appearance of data from man-made structures (*structured* class). Similarly, the data from background (*nonstructured* class) is virtually unconstrained, and there is a large overlap between these two classes. This section focuses on the detection of man-made structures, which can be characterized primarily by the presence of linear structures. A detailed account of the main issues related to this application is given in Appendix A.

The training and the test set contained 108 and 129 images respectively, each of size 256×384 pixels, from the Corel image database. Each image was divided in nonoverlapping 16×16 pixels blocks, and we call each such block an image site. The ground truth was generated by hand-labeling every site in each image as a *structured* or *nonstructured* block. The whole training set contained $36,269$ blocks from the *nonstructured* class, and $3,004$ blocks from the *structured* class. The detailed explanation of the features we used for the structure detection application is given in Appendix A. Here we briefly describe the features to set the notations. The intensity gradients contained within a window (defined later) in the image are combined to yield a histogram over gradient orientations. Each histogram count is weighted by the gradient magnitude at that pixel. To alleviate the problem of hard binning of the data, the histogram is smoothed using kernel smoothing. Heaved central-shift moments are computed to capture the the average *spikeness* of the smoothed histogram as an indicator of the *structuredness* of the patch. The orientation based feature is obtained by passing the absolute difference between the locations of the two highest peaks of the histogram through sinusoidal nonlinearity. The absolute location of the highest peak is also used.

For each image we compute two different type of feature vectors at each site. Using the same notations as introduced in Section 3.2, first a *single-site* feature vector at the site i, $\boldsymbol{s}_i(\boldsymbol{y_i})$ is computed using the histogram from the data $\boldsymbol{y}_i$ at that site (i.e., 16×16 block). Obviously, this vector does not take into account the influence of the data in the neighborhood of that site. The vector $\boldsymbol{s}_i(\boldsymbol{y_i})$ is composed of the first three moments and the two orientation based features described above. Next, a *multiscale* feature vector at the site i, $\boldsymbol{f}_i(\boldsymbol{y})$ is computed which explicitly takes into account the dependencies in the data contained in the neighboring sites. It should be noted that the neighborhood for the data interaction need not be the same as for the label interaction. To compute $\boldsymbol{f}_i(\boldsymbol{y})$, smoothed histograms are obtained at three different scales, where each scale is defined as a varying window size around the site i. The number of scales is chosen to be 3, with the scales changing in regular octaves. The lowest scale is fixed at 16×16 pixels (i.e., the size of a single site), and the highest scale at 64×64 pixels. The moment and orientation based features are obtained at each scale similar to $\boldsymbol{s}_i(\boldsymbol{y_i})$. In addition, two interscale features are also obtained using the highest peaks from the histograms at consecutive scales. To avoid redundancy in the moments based features, only two moment features are used from each scale yielding a 14 dimensional feature vector.

3.3.1 Learning

The parameters of the DRF model $\theta = \{\boldsymbol{w}, \boldsymbol{v}, \beta, K\}$ were learned from the training data using the maximum pseudo-likelihood method described in Section 3.2.3. For the association potential, a transformed feature vector $\boldsymbol{h}_i(\boldsymbol{y})$ was computed at each site i. In this work we used the quadratic transform such that the functions $\phi_k(\boldsymbol{f}_i(\boldsymbol{y}))$ include all the l components of the feature vector $\boldsymbol{f}_i(\boldsymbol{y})$, their squares and all the pairwise products yielding $l + l(l+1)/2$ features [38]. This is equivalent to the kernel mapping of the data using a polynomial kernel of degree two. Any linear classifier in the transformed feature space will induce a quadratic boundary in the original feature space. Since l is 14, the quadratic mapping gives a 119 dimensional vector at each site. In this work, the function $\boldsymbol{\psi}_i$, defined in Section 3.2.2 was chosen to be the same as $\boldsymbol{f}_i$. The pairwise data vector $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ can be obtained either by passing the two vectors $\boldsymbol{\psi}_i(\boldsymbol{y})$ and $\boldsymbol{\psi}_j(\boldsymbol{y})$ through a distance function, e.g. absolute component wise difference, or by concatenating the two vectors. We used the concatenated vector in the present work which yielded slightly better results. This is possibly due to wide within class variations in the *nonstructured* class. For the interaction potential, first order neighborhood (i.e., four nearest neighbors) was considered similar to the Ising model.

First, the parameters of the logistic functions, $\boldsymbol{w}$ and $\boldsymbol{v}$, were estimated separately to initialize the pseudo-likelihood maximization scheme. Newton's method was used for logistic regression and the initial values for all the parameters were set to 0. Since the logistic log-likelihood given in Eq. (3.13) is convex, initial values are not a concern for the logistic regression. Approximately equal number of data points were used from both classes. For the DRF learning, the interaction parameter β was initialized to 0, i.e., no contextual interaction between the labels. The weighting parameter K was initialized to 0.5 giving equal weights to both the data-independent and the data-dependent terms in $I(x_i, x_j, \boldsymbol{y})$. All the parameters θ were learned by using gradient ascent for constrained maximization. The final values of β and K were found to be 0.77, and 0.83 respectively. The learning took 100 iterations to converge in 627 s on a 1.5 GHz Pentium class machine.

To compare the results from the DRF model with those from the traditional MRF framework, we learned the MRF interaction parameter using the pseudo-likelihood formulation. The observation parameters in the MRF were learned separately using maximum likelihood. To model the observations, each class conditional density was modeled as a mixture of Gaussian. The number of Gaussians in the mixture was selected to be 5 using cross-validation. The mean vectors, full covariance matrices and the mixing parameters were learned using the standard EM technique [12]. In EM, all the parameters were initialized randomly over several trials to avoid bad local minima. The parameters that performed best on the cross-validation set were chosen to be the final parameters. For the interaction parameter, the pseudo-likelihood learning algorithm yielded β_m to be 0.68. The learning took 9.5 s to converge in 70 iterations.

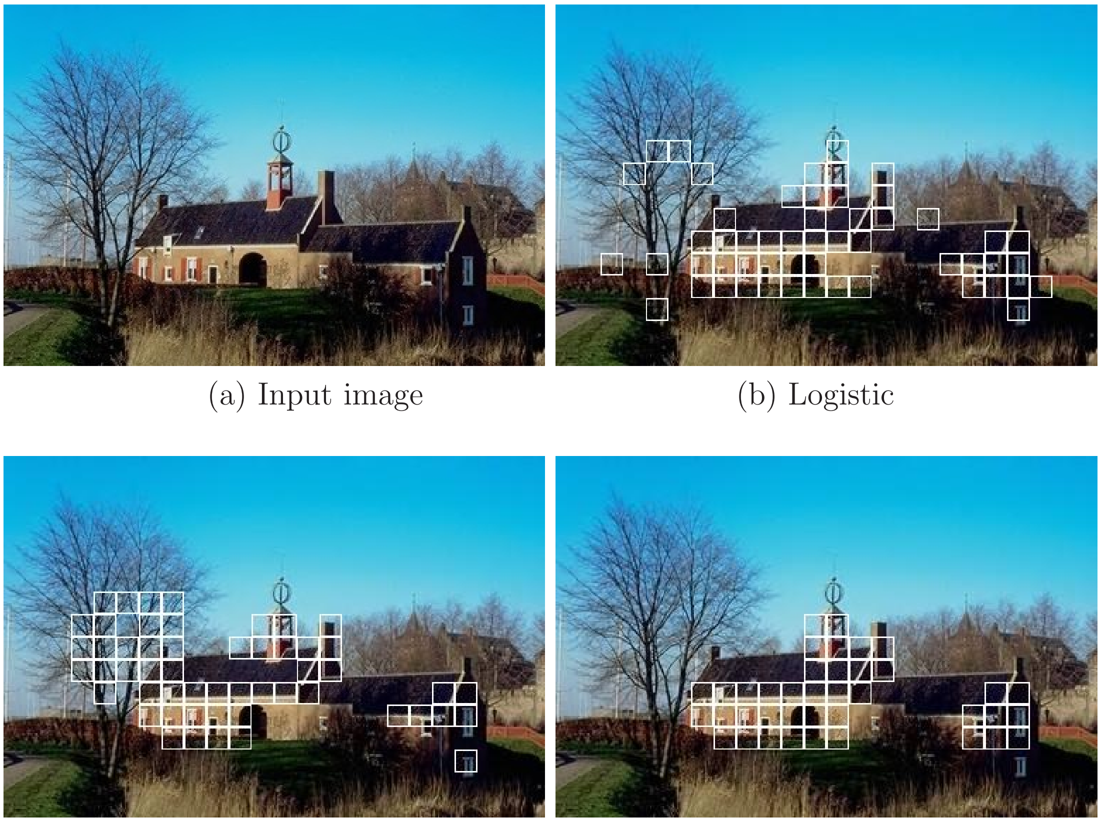
3.3.2 Performance Evaluation

In this section we present a qualitative as well as a quantitative evaluation of the proposed DRF model. First we compare the detection results on the test images using three different methods: logistic classifier with MAP inference, and MRF and DRF with ICM inference. The ICM algorithm was initialized from the maximum likelihood solution for the traditional MRF and from the MAP solution of the logistic classifier for the DRF.

Qualitative Evaluation

For an input test image given in Figure 3.4 (a), the *structure* detection results for the three methods are shown in Figure 3.4. The blocks identified as *structured* have been shown enclosed within an artificial boundary. It can be noted that for similar detection rates, the number of false positives have significantly reduced for the DRF based detection. The logistic classifier does not enforce smoothness in the labels, which led to increased isolated false positives. However, the MRF solution shows a smoothed false positive region around the tree branches because it does not take into account the neighborhood interaction of the data. Locally, different branches may yield features similar to those from the man-made structures. In addition, the discriminative association potential and the data-dependent smoothing in the interaction potential in the DRF also affect the detection results.

The ICM algorithm converged in less than 5 iterations for both the DRF and



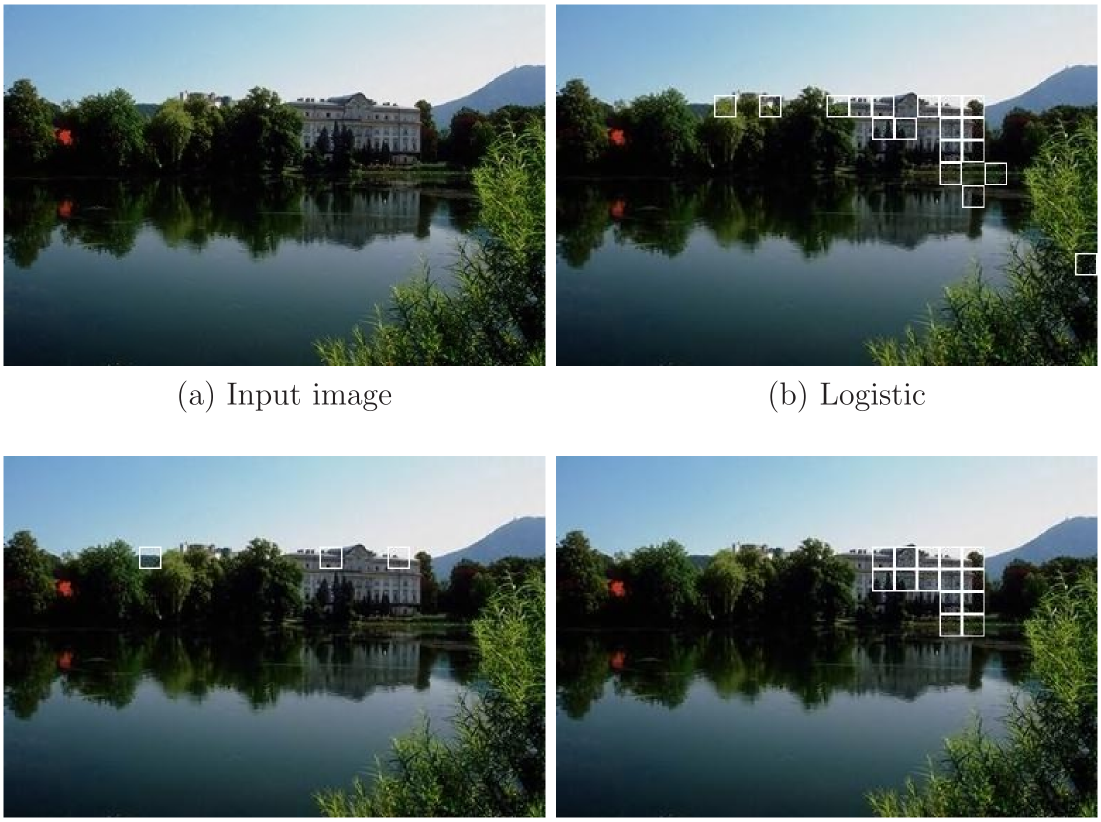
(a) Input image (b) Logistic

(c) MRF (d) DRF

Figure 3.4: Structure detection results on a test example for different methods. For similar detection rates, DRF reduces the false positives considerably.

the MRF. The average time taken in processing an image of size 256×384 pixels in Matlab 6.5 on a 1.5 GHz Pentium class machine was 2.42 s for the DRF, 2.33 s for the MRF and 2.18 s for the logistic classifier. As expected, the DRF takes more time than the MRF due to the additional computation of the data-dependent term in the interaction potential in the DRF.

Additional comparisons of the performance of the three classifiers on two test examples are given in Figure 3.5 and 3.6 respectively. Figure 3.5 shows the detection of a building in poor illumination conditions. Note that the interactions among data in larger neighborhoods beyond a single block are necessary to detect the building as shown by better detection rate of the logistic and the DRF models over the traditional MRF model. On the other hand, enforcing interactions among labels is necessary to reduce isolated false positives as shown by better performance of the DRF than the



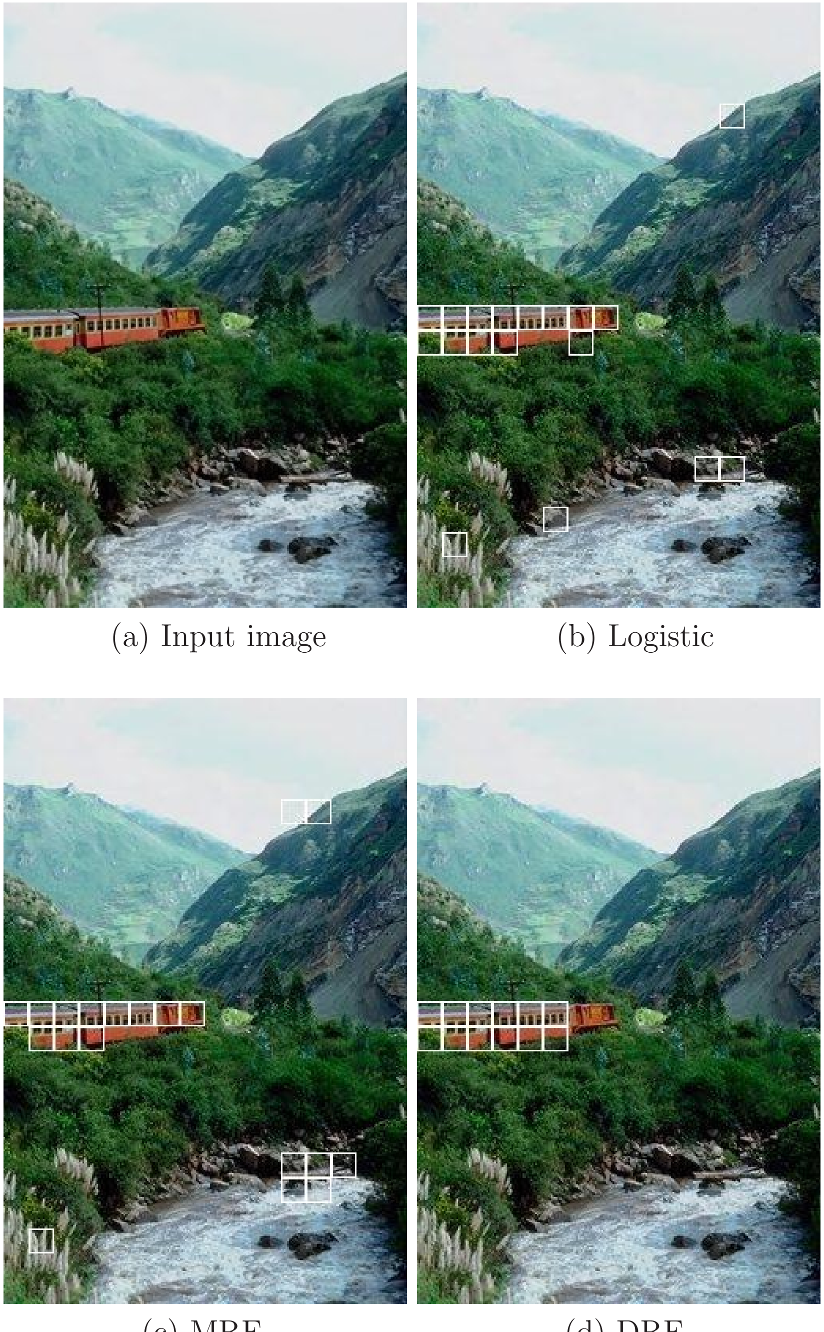
(a) Input image (b) Logistic

(c) MRF (d) DRF

Figure 3.5: Detection of a building in poor illumination conditions in a test image. The interactions among data in larger neighborhoods beyond a single block are necessary to detect the building as shown by better detection rate of the logistic and the DRF models over the traditional MRF model. On the other hand, enforcing interactions among labels is necessary to reduce isolated false positives as shown by better performance of the DRF than the logistic classifier.

logistic classifier. Figure 3.6 shows the detection of a man-made structure (not a building) in a cluttered scene. The DRF model outperforms the other two models.

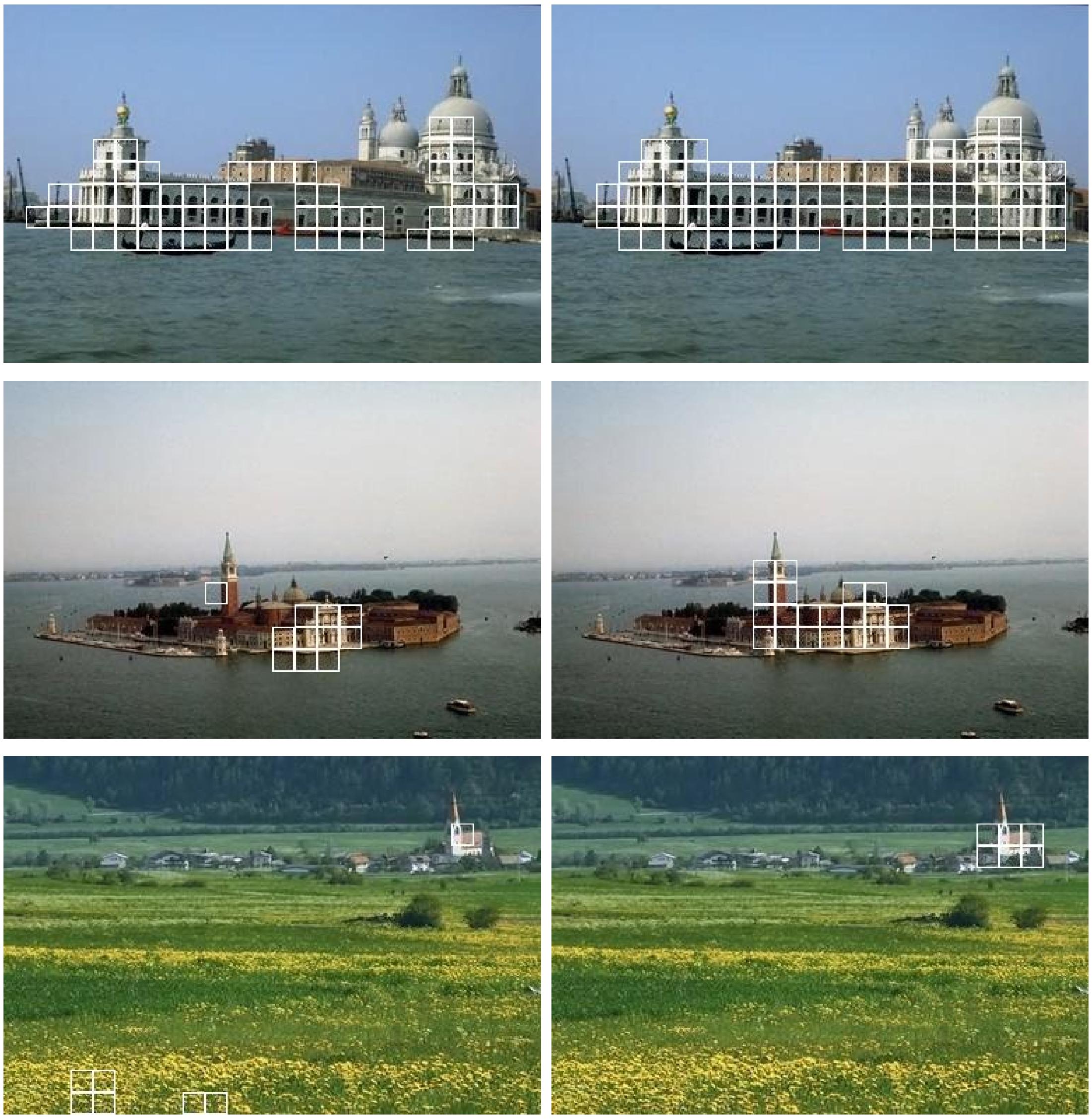
Some more examples from the test set comparing the detection performance of the MRF and the DRF models are shown in Figure 3.7 and Figure 3.8. Figure 3.7 shows the detection of structures at different scales varying from small structures to large structures. Figure 3.8 demonstrates the power of the DRF framework on complex textured structures, and edgy non-structures that locally yield features similar to those from structures. The examples indicate that the data interaction is important for both increasing the detection rate as well as reducing the false positives.



(a) Input image (b) Logistic

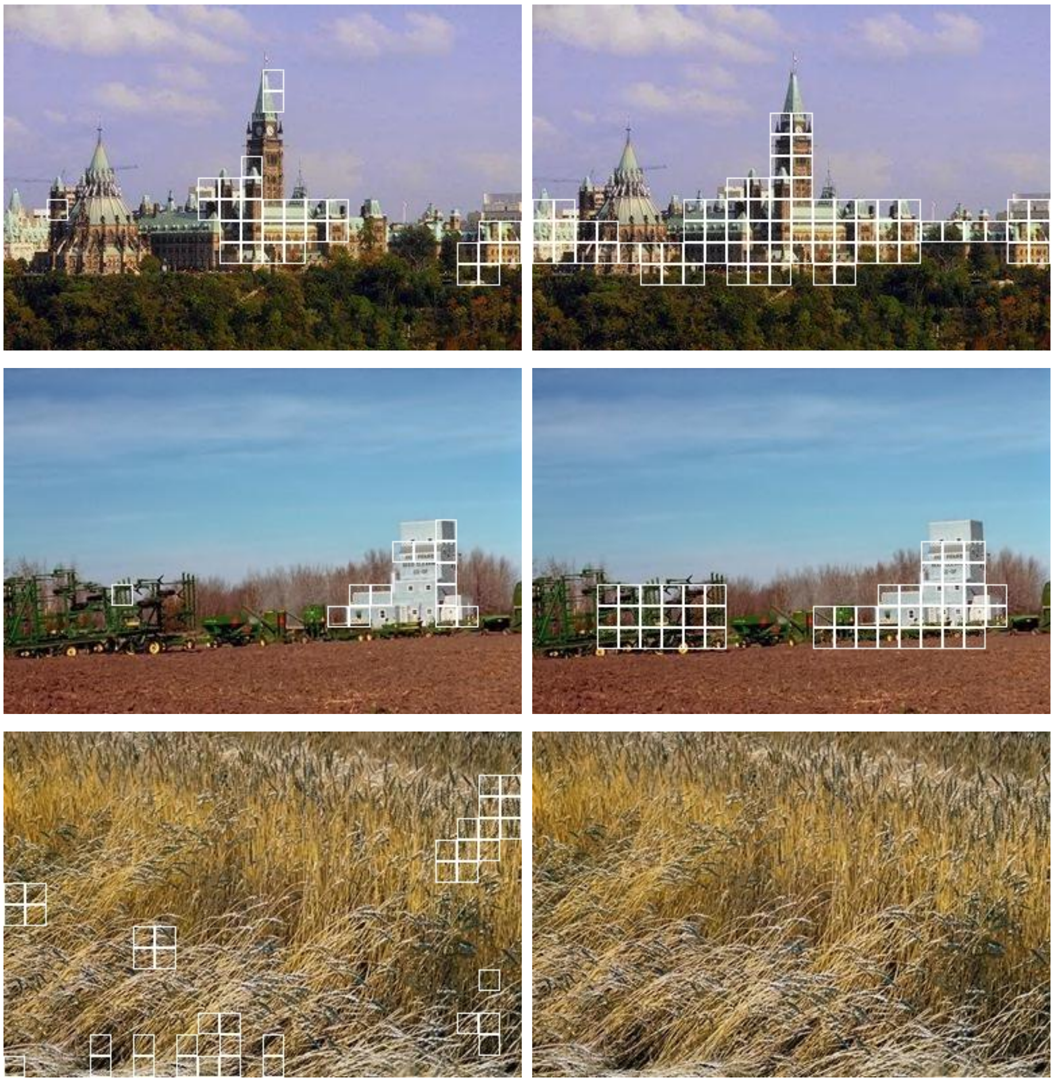
(c) MRF (d) DRF

Figure 3.6: Detection of a man-made structure in a cluttered scene from another test example. The DRF outperforms the other two models.



MRF DRF

Figure 3.7: Structure detection results from the test set at varying degree of scales with large scale structures in the top row and small scale structures in the bottom row. The DRF has higher detection rates and lower false positives in comparison to the traditional MRF.



MRF DRF

Figure 3.8: Some more examples of structure detection from test set. The DRF has higher detection rates and lower false positives in comparison to the traditional MRF. The top image contains structure with complex texture. The bottom row shows detection on *edgy* texture corresponding to clutter.

Figure 3.9: Some typical errors made by the DRF model on the test set. Top row: The tree trunks give very strong man-made structure type features. However, considering the interactions among data in larger neighborhoods, it is still possible to filter most of the false positives. Bottom left: Too small structures are hard to detect due to fixed block size. Bottom right: The DRF has detected most of the subregions of the structure but it fails on the grass-covered walls etc. Should these areas be labeled as *grass* or *man-made structure* or something intermediate?

Finally, we show some typical errors made by the DRF model on the test set in Figure 3.9. In the case of tree images, the tree trunks give very strong man-made structure type features. However, considering the interactions among data in larger neighborhoods in DRFs, it is still possible to filter most of the false positives. In addition, too small structures are hard to detect. This problem may possibly be alleviated by letting the block size be adaptive rather than being fixed at 16×16 pixels. Of course, the cost to pay would be in the form of added computations. We leave the exploration of this issue for future work. Finally, there is an interesting example in the bottom right of Figure 3.9, which raises the philosophical debate about how we define a concept 'building'. From the vision standpoint, the DRF has detected most of the subregions of the structure but it fails on the grass-covered walls etc. Should these areas be labeled as *grass* or *man-made structure* or something intermediate? To be able to infer entities at this level, clearly one has to incorporate knowledge beyond what is currently captured in the random field models. This is a topic for future research.

Quantitative Evaluation

To carry out the quantitative evaluation of our work, we compared the detection rates, and the number of false positives per image for each technique. To avoid the confusion due to different effects in the DRF model, the first set of experiments was conducted using the *single-site* features for all the three methods. Thus, no neighborhood data interaction was used for both the logistic classifier and the DRF i.e., $\boldsymbol{f}_i = \boldsymbol{s}_i$. The comparative results for the three methods are given in Table 3.1 next to 'MRF', 'Logistic$^-$' and 'DRF$^-$'. For comparison purposes, the false positive rate of the logistic classifier was fixed to be the same as the DRF in all the experiments. It can be noted that for similar false positives, the detection rates of the traditional MRF and the DRF are higher than the logistic classifier due to the label interaction. However, the higher detection rate of the DRF in comparison to the MRF indicates the gain due to the use of discriminative models in the association and interaction potentials in the DRF.

In the next experiment, to take advantage of the power of the DRF framework, data interaction was allowed for both the logistic classifier as well as the DRF. Further, to decouple the effect of the data-dependent term from the data-independent term in the interaction potential in the DRF, the weighting parameter K was set to 0. Thus, only data-dependent smoothing was used for the DRF. The DRF parameters were learned for this setting (Section 3.2.3) and β was found to be 1.26. The DRF results ('DRF($K=0$)' in Table 3.1) show significantly higher detection rate than that from the logistic and the MRF classifiers. At the same time, the DRF reduces false positives from the MRF by more than 48%. Finally, allowing all the components of the DRF to act together, the detection rate further increases with a marginal increase in false positives ('DRF' in Table 3.1). However, observe that for the full DRF, the learned value of $K(0.83)$ signifies that the data-independent term dominates in the interaction potential. This indicates that there is some redundancy in the smoothing effects produced by the two different terms in the interaction potential. This is not surprising because the neighboring sites usually have 'similar' data. In Section 3.4.1 we will describe a modified form of the interaction potential that combines these two

Table 3.1: Detection Rates (DR) and False Positives (FP) for the test set containing 129 images. FP for logistic classifier were kept to be the same as for DRF for DR comparison. Superscript $'-'$ indicates no neighborhood data interaction was used. $K = 0$ indicates the absence of the data-independent term in the interaction potential in DRF.

Method	FP (per image)	DR (%)
MRF	2.36	57.2
Logistic$^-$	2.24	45.5
DRF$^-$	2.24	60.9
Logistic	1.37	55.4
DRF ($K = 0$)	1.21	68.6
DRF	1.37	70.5

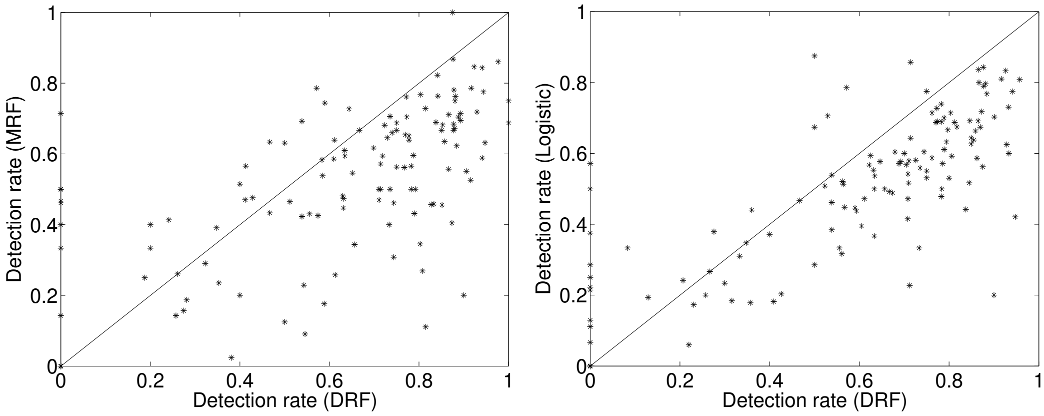


Figure 3.10: Comparison of the detection rates per image for the DRF and the other two methods for similar false positive rates. For most of the images in the test set, DRF detection rate is higher than others.

terms without duplicating their smoothing effects.

To compare per image performance of the DRF with the traditional MRF and the logistic classifier, scatter plots were obtained for the detection rates for each image (Figure 3.10). Each point on these plots is an image from the test set. These plots indicate that for a majority of the images, the DRF has higher detection rate than the other two methods.

To analyze the performance of the MAP inference for the DRF, a MAP solution was obtained using the min-cut algorithm. The overall detection rate was found

Table 3.2: Results with linear classifiers (See text for more).

Method	FP (per image)	DR (%)
Logistic(linear)	2.04	55.0
DRF (linear)	2.04	62.3

to be 24.3% for 0.41 false positives per image. Very low detection rate along with low false positives indicates that MAP is preferring oversmoothed solutions in the present setting. This is because the pseudo-likelihood approximation used in this work for learning the parameters tends to overestimate the interaction parameter β. Our MAP results match the observations made by Greig et al. [51], and Fox and Nicholls [42] for large values of β in MRFs. In contrast, ICM is more resilient to the errors in parameter estimation and performs well even for large β, which is consistent with the results of [51], [42], and Besag [11]. For MAP to perform well, a better parameter learning procedure than using a factored approximation of the likelihood will be helpful. In addition, one may also need to impose a prior that favors small values of β. These observations lay the foundation for an improved parameter learning procedure explained in Section 3.4.2.

One additional aspect of the DRF model is the use of general kernel mappings to increase the classification accuracy. To assess the sensitivity to the choice of kernel, we changed the quadratic functions used in the DRF experiments to compute $\boldsymbol{h}_i(\boldsymbol{y})$ to one-to-one transform such that $\boldsymbol{h}_i(\boldsymbol{y}) = [1 \ \boldsymbol{f}_i(\boldsymbol{y})]$. This transform will induce a linear decision boundary in the feature space. The DRF results with quadratic boundary (Table 3.1) indicate higher detection rate and lower false positives in comparison to the linear boundary (Table 3.2). This shows that with more complex decision boundaries one may hope to do better. However, since the number of parameters for a general kernel mapping is of the order of the number of data points, one will need some method to induce sparseness to avoid overfitting [38]. Lafferty et al. [87] have recently proposed a greedy method of learning the parameters that permits the use general kernels in conditional fields. However, this method is computationally expensive even for moderately large number of data points as in the case of image analysis applications. Thus, a computationally efficient procedure to learn generic kernel classification in conditional fields is still an open question.

3.4 Modified Discriminative Random Field

As explained in the previous section, there were two main reasons that prompted us to explore a modified form of the original DRF and a better parameter learning procedure:

1. The form of the interaction potential given in Eq. (3.10) has redundancy in the smoothing effects produced by the data-independent and the data-dependent terms. Also this form makes the parameter learning a non-convex problem.
2. The pseudo-likelihood parameter learning tends to overestimate the interaction coefficients which makes the global MAP estimates to be bad solutions.

In the following sections we discuss the main components of the original DRF formulation that have been modified.[6]

3.4.1 Interaction potential

For a pair of sites (i, j), let $\boldsymbol{\mu}_{ij}(\boldsymbol{\psi}_i(\boldsymbol{y}), \boldsymbol{\psi}_j(\boldsymbol{y}))$ be a new feature vector such that $\boldsymbol{\mu}_{ij} : \Re^{\gamma} \times \Re^{\gamma} \rightarrow \Re^{q}$, where $\boldsymbol{\psi}_k : \boldsymbol{y} \rightarrow \Re^{\gamma}$. Denoting this feature vector as $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ for simplification, the interaction potential is modeled as,

$$I(x_i, x_j, \boldsymbol{y}) = x_i x_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}), \tag{3.17}$$

where $\boldsymbol{v}$ are the model parameters. Note that the first component of $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is fixed to be 1 to accommodate the bias parameter. There are two interesting properties of the interaction potential given in Eq. (3.17). First, if the association potential at each site and the interaction potentials of all the pairwise cliques except the pair (i, j) are set to zero in Eq. (3.3), the DRF acts as a logistic classifier which yields the probability of the site pair to have the same labels given the observed data. Of course, one can generalize the form in Eq. (3.17) as,

$$I(x_i, x_j, \boldsymbol{y}) = \log P''(x_i, x_j | \boldsymbol{\psi}_i(.), \boldsymbol{\psi}_j(.)), \tag{3.18}$$

similar to the association potential in Section 3.2.1 and can use arbitrary pairwise discriminative classifier to define this term. Recently, a similar idea has been used by

[6]This work appeared in Advances in Neural Information Processing Systems (NIPS '03) [81].

other researchers [119][136]. The second property of the interaction potential form given in Eq. (3.17) is that it generalizes the Ising model. The original Ising form is recovered if all the components of vector $\boldsymbol{v}$ other than the bias parameter are set to zero in Eq. (3.17). Thus, the form of interaction potential given in Eq. (3.17) effectively combines both the terms of earlier model in Eq. (3.10). A geometric interpretation of interaction potential is that it partitions the space induced by the relational features $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ between the pairs that have the same labels and the ones that have different labels. Hence Eq. (3.17) acts as a data-dependent discontinuity adaptive model that will moderate smoothing when the data from the two sites is 'different'. The data-dependent smoothing can especially be useful to absorb the errors in modeling the association potential. Anisotropy can be easily included in the DRF model by parameterizing the interaction potentials of different directional pairwise cliques with different sets of parameters $\boldsymbol{v}$.

3.4.2 Parameter learning and inference

Let θ be the set of DRF parameters where $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$. As shown in Section 3.3.2, maximizing the pseudo-likelihood tends to overestimate the interaction parameters causing the MAP estimates of the field to be very poor solutions. Our experiments in Section 3.4.3 verify these observations for the interaction parameters $\boldsymbol{v}$ in modified DRFs too. To alleviate this problem, we take a Bayesian approach to get the maximum a posteriori estimates of the parameters. Similar to the concept of weight decay in neural learning literature, we assume a Gaussian prior over the interaction parameters $\boldsymbol{v}$ such that $p(\boldsymbol{v}|\tau) = \mathcal{N}(\boldsymbol{v}; \mathbf{0}, \tau^2\boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix. Using a prior over parameters $\boldsymbol{w}$ that leads to weight decay or shrinkage might also be beneficial but we leave that for future exploration. The prior over parameters $\boldsymbol{w}$ is assumed to be uniform. Thus, given M independent training images,

$$\widehat{\theta} = \arg\max_{\theta} \sum_{m=1}^{M} \sum_{i \in S} \left\{ \log \sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y})) + \sum_{j \in \mathcal{N}_i} x_i x_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}) - \log z_i \right\} - \frac{1}{2\tau^2} \boldsymbol{v}^T \boldsymbol{v}, \quad (3.19)$$

$$\text{where} \qquad z_i = \sum_{x_i \in \{-1,1\}} \exp \left\{ \log \sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y})) + \sum_{j \in \mathcal{N}_i} x_i x_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}) \right\}.$$

If τ is given, the penalized log pseudo-likelihood in Eq. (3.19) is convex with respect to the model parameters and can be easily maximized using gradient ascent.

In related work regarding the estimation of τ, Mackay [97] has suggested the use of type II marginal likelihood. But in the DRF formulation, integrating the parameters $\boldsymbol{v}$ is a hard problem. Another choice is to integrate out τ by choosing a non-informative hyperprior on τ as in [151][37]. However our experiments showed that these methods do not yield good estimates of the parameters because of the use of pseudo-likelihood in our framework. In the present work we choose τ by cross-validation. Alternative ways of parameter estimation include the use of contrastive divergence [58] and saddle point approximations resembling perceptron learning rules [23]. In chapter 4, we will discuss several techniques to learn the approximate maximum likelihood parameters in the discriminative fields, that require no hand-tuning of the parameters.

To test the efficacy of the penalized pseudo-likelihood procedure, we were interested in obtaining the MAP estimates of labels $\boldsymbol{x}$ given an image $\boldsymbol{y}$. Following the discussion in Section 3.2.4, the MAP estimates for the modified DRFs can also be obtained using graph min-cut algorithms. However, since these algorithms do not allow negative interaction between the sites, the data-dependent smoothing for each clique in Eq. (3.17) is set to be $\boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y}) = \max\{0, \boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y})\}$, yielding an approximate MAP estimate. This is equivalent to switching the smoothing off at the image discontinuities.

3.4.3 Man-made Structure Detection Revisited

The modified DRF model was applied to the task of detecting man-made structures in natural scenes. The features were fixed to be the same as used in the tests with the original DRF in Section 3.3. The penalty coefficient τ was chosen to be 0.001 for parameter learning. The detection results were obtained using graph min-cuts for both the MRF and the DRF models.

For a quantitative evaluation, we compared the detection rates and the number of false positives per image for the traditional MRF, the DRF and the logistic classifier. Similar to the experimental procedure of Section 3.3.2, for the comparison of detection rates in all the experiments, the decision threshold of the logistic classifier was fixed such that it yields the same false positive rate as the DRF. The first set of experiments was conducted using the *single-site* features for all the three methods. Thus, no neighborhood data interaction was used for both the logistic classifier and the DRF, i.e. $\boldsymbol{f}_i(\boldsymbol{y})=\boldsymbol{s}_i(\boldsymbol{y}_i)$. The comparative results for the three methods are given in Table 3.3 under 'MRF', 'Logistic$^-$' and 'DRF$^-$'. The detection rates of the traditional MRF and the DRF are higher than the logistic classifier due to the label interaction. However, higher detection rate and lower false positives for the DRF in comparison to the MRF indicate the gains due to the use of discriminative models in the association and interaction potentials in the DRF. In the next experiment, to take advantage of the power of the DRF framework, data interaction was allowed for both the logistic classifier as well as the DRF ('Logistic' and 'DRF' in Table 3.3). The DRF detection rate increases substantially and the false positives decrease further indicating the importance of allowing the data interaction in addition to the label interaction.

Now we compare the results of the modified DRF formulation with those from the original DRF. Comparing the results in table 3.1 with those in table 3.3, we find that the original DRF (with ICM inference) gave 70.5% correct detection with 1.37 average false positive per image in comparison to 72.5% correction detection and 1.76 false positives from the modified DRF (with MAP inference). Even though the results seem to be comparable for this application, we have achieved two main advantages in the modified DRFs. First, in comparison to the original DRF formulation, the modified DRF has a much simpler form of interaction potential with comparatively better behaved parameter learning problem (a convex problem). Second, clearly the experiments in this section reveal that the bad MAP solutions of DRFs were due to a particular parameter learning scheme (pseudo-likelihood) we chose in our earlier experiments. It overcomes the criticism of the original DRFs on the ground that if the global minimum of the energy ($-\log P(x|y)$) given by MAP is not an acceptable solution, it probably implies that the DRFs are not appropriate for the purpose of classification.

These results also point toward another interesting observation regarding the compatibility of a parameter learning procedure with the inference procedure. Local parameter learning (pseudo-likelihood) seems to be yielding acceptable, though usually not the best, results when used with a local inference mechanism (ICM). On the other hand, to make a global inference scheme yield good solutions, it is inevitable to use nonlocal learning procedures. We will further elaborate on such *coupling* of inference and parameter learning procedures in Chapter 4.

Finally, one common question that arises for any classifier is: how good is the generalization capability of the classifier? In other words, how well does the classifier

Table 3.3: Detection Rates (DR) and False Positives (FP) for the test set containing 129 images (49, 536 sites). FP for logistic classifier were kept to be the same as for DRF for DR comparison. Superscript $'-'$ indicates no neighborhood data interaction was used.

	MRF	Logistic$^-$	DRF$^-$	Logistic	DRF
DR (%)	58.35	47.50	61.79	60.80	72.54
FP (per image)	2.44	2.28	2.28	1.76	1.76

perform on unseen data? Our experiments with the Corel test data (which was unseen during training) indicate good generalization performance of the DRF-based classifier as described above. However, to further evaluate this capability of DRFs, we tested their performance on images that were not from the Corel set. The aim of this experiment, conducted by R. Collins at CMU, was to find man-made structures in outdoor scenes for the navigation of a robotic boat. It is worth noting that the same parameter setting was used in this experiment as learned from the Corel training set. The DRF was able to detect man-made structures with high accuracy (about 90 %) at the speed of about 5 frames (320×240 pixels each) per second.

3.4.4 Binary Image Denoising Task

We applied the DRF formulation to another task of image denoising. The aim of these experiments was to obtain true underlying images from their corrupted versions. Traditionally this application has been used heavily as a testbed by the researchers working on MRFs [10][51]. In our experiments, four base images of size 64×64 pixels each were used (top row in Figure 3.11). The training as well as the test sets were created by corrupting the base images with two different types of noises. The first noise model was a simple Gaussian noise and the second was a 'bimodal' noise (i.e., mixture of two Gaussians). For each noise model, 50 corrupted versions were generated from each base image.

In this application, each pixel was considered as an image site and the feature vector $\boldsymbol{s}_i(\boldsymbol{y}_i)$ was simply chosen to be a scalar representing the intensity at i^{th} site. No neighborhood data interaction was used for the DRFs in these experiments (i.e., $\boldsymbol{f}_i(\boldsymbol{y}) = \boldsymbol{s}_i(\boldsymbol{y}_i)$) to observe the gains only due to the use of discriminative models in the association and interaction potentials. A linear discriminant was implemented

Table 3.4: Pixelwise classification errors (%) on 150 synthetic test images. For the Gaussian noise, MRF and DRF give similar error while for 'bimodal' noise, DRF performs better. Note that only label interaction (i.e. no data interaction) was used for these tests (see text).

Noise	ML	Logistic	MRF (PL)	DRF (PL)	MRF	DRF
Gaussian	15.62	15.78	2.66	3.82	2.35	2.30
Bimodal	24.00	29.86	8.70	17.69	7.00	6.21

in the association potential such that $\boldsymbol{h}_i(\boldsymbol{y}) = [1, \boldsymbol{f}_i(\boldsymbol{y})]^T$. The pairwise data vector $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ was obtained by taking the absolute difference of $\boldsymbol{s}_i(\boldsymbol{y}_i)$ and $\boldsymbol{s}_j(\boldsymbol{y}_j)$. For the traditional MRF model, a linear discriminant was implemented by modeling each class-conditional density, $p(\boldsymbol{s}_i(\boldsymbol{y}_i)|x_i)$, as a Gaussian. The learning of parameters of a single Gaussian is a trivial convex problem. The noisy data from the left most base image in Figure 3.11 was used for training while 150 noisy images from the rest of the three base images were used for testing.

Three experiments were conducted for each noise model. (i) The interaction parameters for the DRF ($\boldsymbol{v}$) as well as for the MRF (β_m) were set to zero. This reduces the DRF model to a logistic classifier and MRF to a local maximum likelihood (ML) classifier. (ii) The parameters of the DRF, i.e. $[\boldsymbol{w}, \boldsymbol{v}]$, and the MRF, i.e. β_m, were learned using pseudo-likelihood approach without any penalty, i.e. $\tau = \infty$. (iii) Finally, the DRF parameters were learned using penalized pseudo-likelihood and the best β_m for the MRF was chosen from cross-validation. The MAP estimates of the labels were obtained using graph-cuts for both the models.

Under the first noise model, each image pixel was corrupted with independent Gaussian noise of standard deviation 0.3. For the DRF parameter learning, τ was chosen to be 0.01. The pixelwise classification error for this noise model is given in the top row of Table 3.4. Since the form of noise is the same as the likelihood model in the MRF, MRF is expected to give good results. The DRF model does marginally better than MRF even for this case. Note that the DRF and the MRF results are worse when the parameters were learned without penalizing the pseudo-likelihood (shown in Table 3.4 with suffix (PL)). The MAP inference yields oversmoothed images for these parameters. The DRF model is affected more because all the parameters in DRFs are learned simultaneously unlike MRFs.

In the second noise model each pixel was corrupted with independent mixture of Gaussian noise. For each class, a mixture of two Gaussians with equal mixing weights was used yielding a 'bimodal' class noise. The mixture model parameters (mean, std) for the two classes were chosen to be $[(0.08, 0.03), (0.46, 0.03)]$, and $[(0.55, 0.02), (0.42, 0.10)]$ inspired by [125]. The classification results are shown in the bottom row of Table 3.4. An interesting point to note is that DRF yields lower error than MRF even when the logistic classifier has higher error than the ML classifier on the test data. For a typical noisy version of the four base images, the performance of different techniques in compared in Figure 3.11. The logistic classifier gives very poor results because it classifies each pixel independently. It ignores the very basic theme of underlying smoothness of natural images due to which one can hope for recovering the true image from its noisy version. The DRF gives better performance than the MRF model.

3.5 Summary

In this chapter, we introduced noncausal Discriminative Random Field models that combine local discriminative classifiers for individual classification of image sites with interaction between neighboring sites. These models allow capturing spatial dependencies in labels and observed data simultaneously in a principled manner on 2D image lattices. Two different parameter learning procedures i.e., pseudo-likelihood and penalized pseudo-likelihood were described. The inference over these models was carried out using min-cut and ICM. We demonstrated the advantages of these models over the traditional generative MRFs on two different tasks: man-made structure detection in outdoor scenes and binary image denoising. The experiments suggest some type of coupling between parameter learning and inference methods which is explored more in the next chapter.

Several extensions are required to demonstrate the power of DRFs on other real-world classification tasks. The first natural step is to extend the proposed binary DRF model to accommodate multiclass classification problems. The multiclass extension of the binary DRFs is relatively simple and will be described in Chapter 5. The next most important challenge in the DRF framework is robust and fast learning of the model parameters. In the next chapter, we will discuss several approximate techniques to obtain the maximum likelihood parameter estimates in discriminative

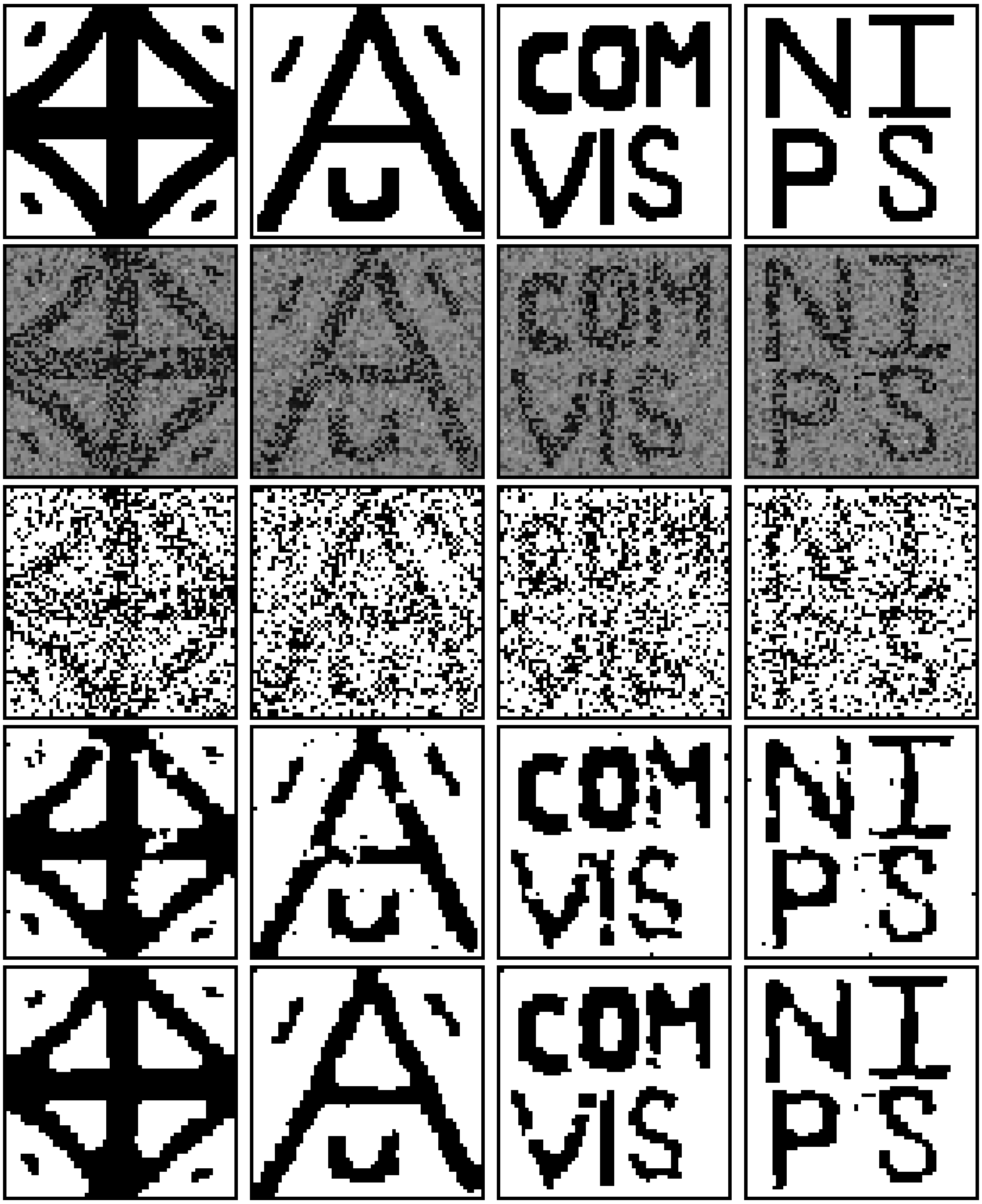


Figure 3.11: Results of binary image denoising task. From top, first row:original images, second row: images corrupted with 'bimodal' noise, third row: Logistic Classifier results, fourth row: MRF results, fifth row: DRF results.

fields.

Chapter 4

Approximate Parameter Learning

4.1 Introduction

In the previous chapter we introduced discriminative field models for the labeling of image components. One of the crucial requirements to make these models applicable to a variety of real-world tasks is accurate and efficient parameter learning in these models. For 1-D sequential CRFs proposed by Lafferty et al. [86], exact maximum likelihood parameter learning is feasible because the induced graph does not contain loops. This is because the loop-free graphs allow easy computation of the partition function using Dynamic Programming. Several efficient techniques have been proposed to learn parameters in these models, e.g., iterative scaling [86][27][9], quasi-Newton methods [129][126], conjugate gradient [143] and gradient boosting [30]. However, when a graph contains loops, it is not feasible to compute the partition function using Dynamic Programming. Thus, it is difficult to exactly maximize the likelihood with respect to the parameters. Therefore, a critical issue in applying discriminative fields is the design of effective parameter learning techniques that can operate on arbitrary graphs. The objective of this chapter is to address this central issue.

In the previous chapter we described two methods of learning the parameters in discriminative fields: pseudo-likelihood and penalized pseudo-likelihood. As discussed in Section 3.3 and Section 3.4.3, pseudo-likelihood tends to overestimate the field parameters leading to poor performance with global inference such as MAP. Penalizing the pseudo-likelihood can result in good performance but it requires ad-hoc hand-tuning of the penalizing coefficient.

In this chapter, our goal is to explore various automatic parameter learning techniques in arbitrary discriminative fields [1]. Here, we approximate the gradients of the log likelihood function directly using the inference techniques. Our experimental results may be summarized by the following two observations: First, *parameter learning* can be achieved by approximating the likelihood gradient using the label estimates obtained through methods such as Maximum A Posteriori (MAP) or Maximum Posterior Marginal (MPM) for the given conditional probability model. Second, good classification performance can be achieved by any of these approximations, so long as the method used for inference matches the method used for approximating the gradient in the parameter learning. We note that this *learning/inference coupling* is reasonable because the usual goal in classification problems is to minimize the number of errors, which is what our gradient approximation does, even though this may not necessarily maximize the likelihood. We also present a new experimental comparison of several learning and inference algorithm combinations for guiding what type of learning approximation should be adopted for a given choice of inference method.

4.2 Parameter learning approach

We take a supervised training approach to learning the parameters of the DRF model. The data required are the observed training images and their corresponding ground-truth labeling e.g., known segmentation. In this work we focus on the standard maximum likelihood approach to learning the parameters.

4.2.1 Maximum likelihood parameter learning

Let θ be the set of unknown DRF parameters where $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$. Given M i.i.d. labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood $l(\theta) = \sum_{m=1}^{M} \log P(\boldsymbol{x}^m|\boldsymbol{y}^m, \theta)$, i.e.,

[1] A shorter version of this work will appear in Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), 2005 [78].

$$\widehat{\theta}=\underset{\theta}{\operatorname{argmax}}\sum_{m=1}^{M}\left\{\sum_{i\in S^m}\log\sigma(x_i^m\boldsymbol{w}^T\boldsymbol{h}_i(\boldsymbol{y}^m))+\sum_{i\in S^m}\sum_{j\in\mathcal{N}_i}x_i^m x_j^m\boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m)-\log Z^m\right\}, \quad (4.1)$$

where the partition function for the m^{th} image is,

$$Z^m=\sum_{\boldsymbol{x}}\exp\left\{\sum_{i\in S^m}\log\sigma(x_i\boldsymbol{w}^T\boldsymbol{h}_i(\boldsymbol{y}^m))+\sum_{i\in S^m}\sum_{j\in\mathcal{N}_i}x_i x_j\boldsymbol{v}^T\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m)\right\}.$$

Note that Z^m is a function of the parameters θ and the observed data $\boldsymbol{y}^m$. For learning the parameters using gradient ascent, the derivatives of the log-likelihood are

$$\frac{\partial l(\theta)}{\partial \boldsymbol{w}}=\frac{1}{2}\sum_m\sum_{i\in S^m}(x_i^m-\langle x_i\rangle_{\theta;\boldsymbol{y}^m})\boldsymbol{h}_i(\boldsymbol{y}^m), \quad (4.2)$$

$$\frac{\partial l(\theta)}{\partial \boldsymbol{v}}=\sum_m\sum_{i\in S^m}\sum_{j\in\mathcal{N}_i}(x_i^m x_j^m-\langle x_i x_j\rangle_{\theta;\boldsymbol{y}^m})\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m). \quad (4.3)$$

Here $\langle\cdot\rangle_{\theta;\boldsymbol{y}^m}$ denotes expectation with $P(\boldsymbol{x}|\boldsymbol{y}^m,\theta)$. Ignoring $\boldsymbol{\mu_{ij}}(\boldsymbol{y}^m)$, gradient ascent with Eq. (4.3) resembles the learning problem in Boltzmann machines [2][60], with all the nodes being observed at the training stage and computing the expectations can be seen as the 'free' phase.

Generally the expectations in Eq. (4.2) and Eq. (4.3) cannot be computed analytically due to the combinatorial size of the label space. Sampling procedures such as Markov Chain Monte Carlo (MCMC) can be used to approximate the true expectations [65]. A practical use of MCMC requires the Markov process to converge sufficiently fast - in polynomial time - to the equilibrium distribution. This property known as 'rapid mixing' does not hold in general [130]. Unfortunately, MCMC techniques have two main problems: a long 'burn-in' period (which makes them slow) and high variance in estimates [58]. To avoid the MCMC drawbacks, Contrastive Divergence (CD) was proposed by Hinton [58] which is explained more in Section 4.3.1. The approximation of expectations in CD inspired the different approximations we propose in this work, as shown in Section 4.3.

4.2.2 Coupling parameter learning and inference

The approximations, commonly used in the literature, replace the exact gradient of Eq. (4.2) and Eq. (4.3) by $\boldsymbol{J(\theta)} = (\boldsymbol{J_1}(\theta), \boldsymbol{J_2}(\theta))$ where,

$$\boldsymbol{J_1}(\theta) = \frac{1}{2}\sum_m \sum_{i \in S^m} (x_i^m - f_i(\theta; \boldsymbol{y}^m))\boldsymbol{h}_i(\boldsymbol{y}^m), \tag{4.4}$$

$$\boldsymbol{J_2}(\theta) = \sum_m \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} (x_i^m x_j^m - g_{ij}(\theta; \boldsymbol{y}^m))\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m), \tag{4.5}$$

and f_i and g_{ij} are functions that approximate the true expectations in the gradient. Several approaches have been proposed that compute f_i and g_{ij} using pseudo-marginals [141][101]. In this work, we propose to directly construct f_i and g_{ij} using label estimates obtained through MAP and MPM inference at the current parameter estimates (Section 4.3.3 and 4.3.4).

However, will the gradient ascent of the likelihood with such gradients still converge? The answer is that, while the approximate gradient ascent is not strictly convergent in general, it is weakly convergent in that it oscillates within a set of good parameters, or converges to a good parameter with isolated large deviations, as shown experimentally in Section 4.4. But why should the parameters learned using a particular choice of approximating functions yield good classification performance? Informally, if we use for parameter learning the same approximating function f_i that was used for inference (e.g. MAP label estimate), then, given input training labels $\{x_i^m\}$,

$$N_E^\theta = \frac{1}{2}\sum_m \sum_{i \in S^m} |x_i^m - f_i(\theta; \boldsymbol{y}^m))| \tag{4.6}$$

can be interpreted as the number of errors in classification. Comparing Eq. (4.6) with Eq. (4.4) shows that the approximated gradient is directly related to the number of errors, so long as the *same approximation is used in both parameter learning and inference*. We provide more details in Section 4.6.1.

4.3 Candidate approximations

We first explore the form of f_i and g_{ij} based on Contrastive Divergence and pseudo-marginals, and then using two approximations directly based on two different label estimates: Maximum A Posteriori (MAP) which is optimal for 0-1 loss function, and Maximum Posterior Marginal (MPM) which is optimal for 'sitewise' 0-1 loss function. For the binary DRFs, approximate MAP estimates were obtained using the min-cut/max-flow algorithms as explained in [81]. We use the sum-product version of loopy Belief Propagation (BP) to obtain the MPM estimates [157]. The approximations described in Section 4.3.2 to Section 4.3.4 are designed to match these two classes of inference techniques.

4.3.1 Contrastive Divergence (CD)

In CD proposed by Hinton [58], only a single MCMC move is made from the current empirical distribution of the data (P^0) leading to new distribution (P^1), thus eliminating the need for running the chain beyond burn-in as in traditional MCMC approaches. According to CD,

$$\langle x_i \rangle_{\theta;\boldsymbol{y}} \approx \langle x_i \rangle^{P^1}_{\theta;\boldsymbol{y}} \quad \text{and} \quad \langle x_i x_j \rangle_{\theta;\boldsymbol{y}} \approx \langle x_i x_j \rangle^{P^1}_{\theta;\boldsymbol{y}}.$$

Even though CD is computationally simple and yields estimates with low variance, the bias in estimates can be a problem [150], which was also verified in our experiments in Section 4.5.

4.3.2 Pseudo-Marginal Approximation (PMA)

It is easy to see that if we had true marginal distributions $P_i(x_i|\boldsymbol{y}, \theta)$ at each site i, and $P_{ij}(x_i, x_j|\boldsymbol{y}, \theta)$ at each pair of sites i and $j \in \mathcal{N}_i$, we could compute exact expectations as:

$$\langle x_i \rangle_{\theta;\boldsymbol{y}} = \sum_{x_i} x_i P_i(x_i|\boldsymbol{y}, \theta) \quad \text{and} \quad \langle x_i x_j \rangle_{\theta;\boldsymbol{y}} = \sum_{x_i, x_j} x_i x_j P_{ij}(x_i, x_j|\boldsymbol{y}, \theta).$$

Since computing exact marginal distributions is in general infeasible, a standard approach is to replace the actual marginals by pseudo-marginals [101]. Here, we again used loopy BP to get these marginals. Since loopy BP assumes a tree approximation of the graph [157], it is expected to produce better approximations of these marginals than mean-field, which assumes the nodes in the graph to be disconnected. McCallum et al. [101] use a similar approximation, where pseudo-marginals estimated using Tree-based Reparametrization (TRP) were used for parameter learning in Factorial CRFs.

4.3.3 Learning with MAP inference: Saddle Point Approximation (SPA)

Here, we propose a very simple approximation inspired by CD [58], using MAP label estimates. It is based on approximating the partition function (Z) using the Saddle Point Approximation (SPA) [48]. According to SPA, Z is approximated such that the summation over all the label configurations $\boldsymbol{x}$ in Z is replaced by the largest term in the sum, which occurs at the most probable label configuration. In other words, if

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{y}, \theta),$$

then according to SPA,

$$Z \approx \exp\left\{\sum_{i\in S} \log \sigma(\hat{x}_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y})) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i} \hat{x}_i \hat{x}_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y})\right\}.$$

This leads to a very simple approximation to the expectation, i.e., $\langle x_i\rangle_{\theta;\boldsymbol{y}} \approx \hat{x}_i$ (This approximation would be exact if $\boldsymbol{x}$ were Gaussian.). If we further assume mean-field type of decoupling [115], i.e., $\langle x_i x_j\rangle_{\theta;\boldsymbol{y}} = \langle x_i\rangle_{\theta;\boldsymbol{y}} \langle x_j\rangle_{\theta;\boldsymbol{y}}$, it also follows that $\langle x_i x_j\rangle_{\theta;\boldsymbol{y}} \approx \hat{x}_i \hat{x}_j$. However, this approximation is different from that of mean-field in [115] because here expectations, $\langle x_i\rangle_{\theta;\boldsymbol{y}}$ are replaced by the modes of the exact field, instead of the means of an approximated field. It is interesting to note that with the saddle point approximation of Z, the gradient ascent updates are similar to the perceptron-learning type updates used in [91] and [23] in nonprobabilistic settings.

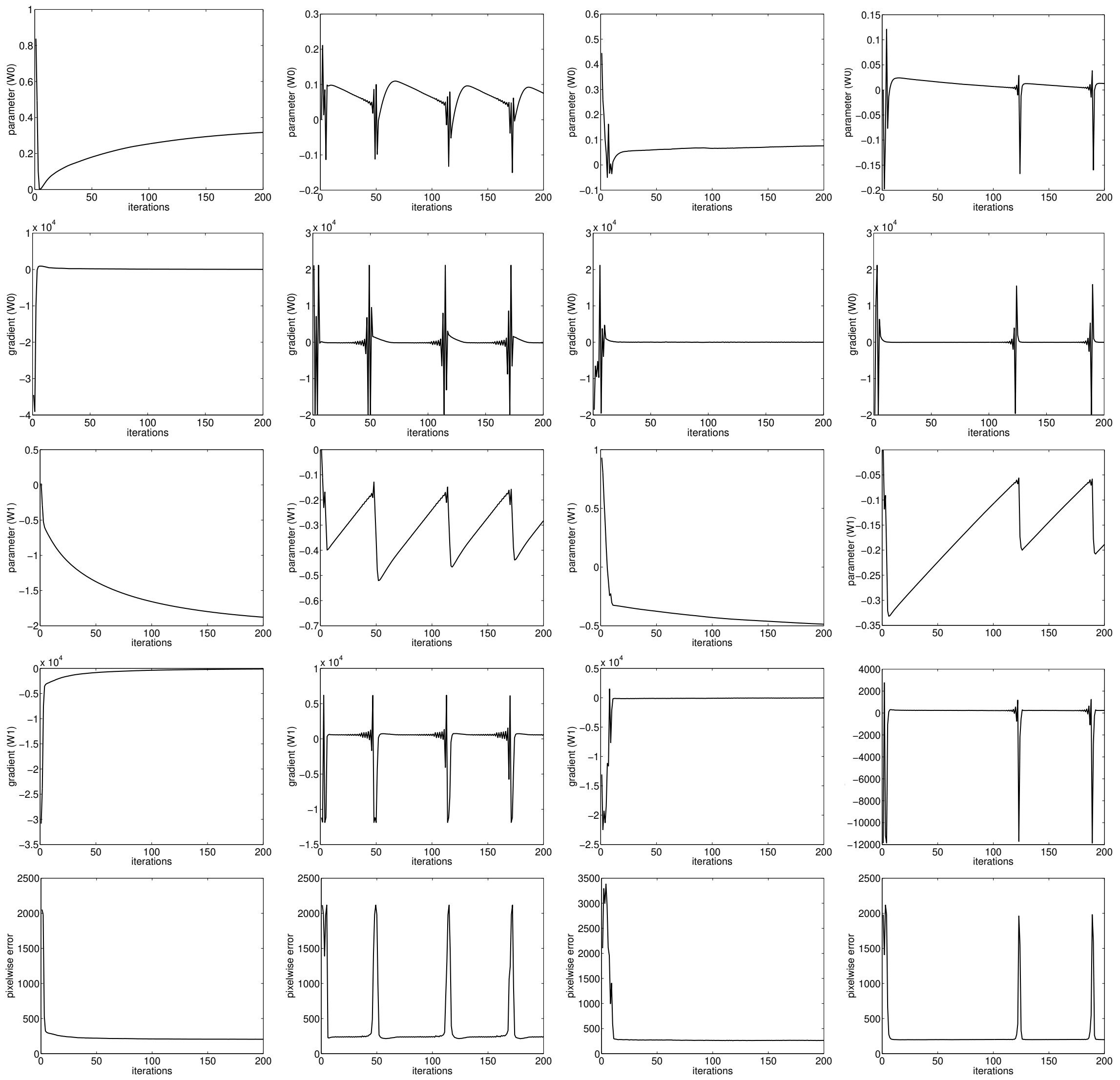
4.3.4 Learning with MPM inference: Maximum Marginal Approximation (MMA)

This is the second approximation based on BP inference in which Maximum Posterior Marginal (MPM) label estimates are used for approximating the expectations. Following the arguments of SPA-based parameter learning in the previous section, one can make a similar approximation of Z such that all the mass of Z is assumed to be concentrated on the maximum marginal configuration, $\tilde{x}_i = \arg\max_{x_i} P_i(x_i|\boldsymbol{y}, \theta)$. The expectations in this case can be written as: $\langle x_i \rangle_{\theta;\boldsymbol{y}} \approx \tilde{x}_i$ and $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}} \approx \tilde{x}_i \tilde{x}_j$. Clearly, in the binary case, maximum marginals are just the thresholded sitewise marginals. Thus, MMA can be interpreted as a discrete approximation of PMA. We experimented with both MMA and SPA in order to gain a better understanding of the consequences of discretization (see Section 4.4 and 4.6.1).

4.4 Experimental observations: parameter learning

To analyze the convergence behavior of various parameter learning procedures described in the previous section, we learned a DRF model for a binary image denoising application. The aim was to obtain true labels from corrupted binary images. A binary image (leftmost image in the top row of Figure 4.2) of size 64×64 pixels was corrupted by two types of noise: Gaussian noise and Bimodal (mixture of two Gaussians) noise. For each noise model, 10 noisy images were used as the training set for learning the parameters. The unary and pairwise features were defined as: $\boldsymbol{h}_i(\boldsymbol{y}) = [1, I_i]^T$ and $\boldsymbol{\mu}_{ij}(\boldsymbol{y}) = [1, |I_i - I_j|]^T$ respectively, where I_i and I_j are the pixel intensities at site i and site j. The details of the noise parameters for this dataset are given in [81]. Here, the parameter vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ both were two-element vectors, i.e., $\boldsymbol{w} = [w_0 \;\; w_1]$, and $\boldsymbol{v} = [v_0 \;\; v_1]$.

In all the experiments, the parameters were initialized from random values and updates were based on gradient ascent. The step size η was fixed to a small value (10^{-5}). Fig. 4.1 shows, for each approximation, plots of the approximated gradients and the parameters at each iteration for a typical run with bimodal noise. For brevity we show plots only for parameters w_0 and w_1. The other parameters behaved similarly. The last row in Figure 4.1 shows the number of training errors (N_E^θ) made at the



(a) PMA (b) SPA-1 (c) SPA-2 (d) MMA

Figure 4.1: Plots of DRF parameter (w_0) updates (top row), and the approximate gradient (second row) for different approximations. PMA shows a converging behavior while SPA shows oscillations which may be large-scale (SPA-1) or small-scale (SPA-2) depending on the initialization of the parameters. MMA shows similar behavior as SPA. Rows 3 and 4 show the analogous plots for parameter w_1. The last row shows number of errors at each parameter update. The errors are low when the gradient magnitudes are small.

current estimate of the parameters using the same inference technique on which a particular gradient approximation is based.

Since the log likelihood in Eq. (4.1) is a convex function of parameters, the final parameter values at convergence will be independent of their initialization in the true gradient ascent. For the PMA based learning, this desirable behavior was seen (Fig. 4.1 (a)). This indicates that the beliefs from loopy BP were possibly converging to reasonable estimates for this dataset.

For SPA and MMA based learning, an interesting behavior emerges since both of them make discrete approximations of the true expectations. It was found that any random initialization of the parameters in the approximate gradient ascent using SPA or MMA yields one of the two different stereotypical patterns of limit cycle convergence (see Section 4.6.1). For SPA, we denote these two patterns as SPA-1 and SPA-2. In the first pattern (Figure 4.1 (b), 'SPA-1'), the approximated gradients for all the parameters show oscillatory behavior. Initially there are large oscillations in gradients which later settle down to a low gradient zone. The gradients remain in this zone for a relatively long duration before showing large oscillations with changing sign again. Note that this will not occur for the gradient ascent with true gradients if suitably small η is chosen. One possibility of damping the oscillations is by annealing η following a decrementing schedule for η. However such ad-hoc procedures of forcing convergence lead to bias in the final parameters. In the oscillatory case, there are several commonly-used heuristics for choosing the parameters when convergence is not guaranteed, e.g., the voted perceptron used in [43] and [23]. In this work we simply used majority vote parameter setting, i.e., the parameters for which the training error was minimum.

In the second kind of SPA pattern ('SPA-2'), as seen in Figure 4.1 (c), after initial oscillations, the gradients do not show 'periodic' large oscillations again but maintain microscopic oscillations within low gradient zones (not visible in the figure due to the scale of the plots). The MMA based learning showed similar behavior as for the SPA indicating that these behaviors are related to the discrete, piecewise constant approximation of the actual expectations. An oscillating gradients case for MMA is shown in Figure 4.1 (d). In Section 4.6.1 we will discuss these limit cycle behaviors of SPA and MMA based learning procedures.

Finally, note that number of errors for all approximations is small whenever gradient magnitudes are small which indicates that all the three techniques tend to achieve parameter values that minimize the errors for that particular inference. This is especially interesting in the case of SPA and MMA because of the nature of the approximations. We will compare the performance of the parameter learning procedures with different inference techniques on a separate test set in Section 4.5.

4.5 Experimental observations: inference

The aim of these experiments was to compare the performance of different parameter learning procedures for a *fixed* inference procedure. For each noise model introduced in Section 4.4, a test set of 200 noisy images was generated using 50 noisy images each from four base images shown in top row of Figure 4.2. For comparison, we also obtain the local MAP solution using Iterated Conditional Modes (ICM) [11] which has been shown to be robust to incorrect parameter settings. In addition, we also compare results with parameters learned through pseudo-likelihood (PL), which uses a factored approximation of the partition function, Z, for tractability [81].

Figure 4.2 shows the denoising performance on four typical test images corrupted by the 'bimodal' noise. The parameters were first learned using existing techniques, i.e., pseudo-likelihood and contrastive divergence. It is clear from the figure that both the techniques give poor results with MAP or MPM inference. The MAP inference with the matched learning technique, i.e., SPA, yields good results as shown in Figure 4.3. The same is true for MPM inference with MMA learning.

Table 4.1: Pixelwise classification errors (%) on 10 **training** images (64×64 pixels each). The rows show different parameter learning procedures and the columns show different inference techniques used for two different noise models. To interpret this table, for each noise model, different parameter learning techniques should be compared by fixing a column that corresponds to a fixed inference technique.

		Gaussian noise			Bimodal noise			Learning time
Inference methods		MAP	MPM	ICM	MAP	MPM	ICM	(Sec)
	SPA	**2.89**	7.94	5.13	**5.29**	11.56	19.52	81.52
Parameter	PMA	3.62	**3.01**	4.84	5.94	**4.92**	22.70	1183.13
Learning	MMA	48.36	**3.01**	10.33	22.67	**4.83**	14.93	635.78
Methods	PL	4.58	3.80	4.76	22.74	6.95	29.38	299.75
	CD	4.40	3.39	**4.58**	7.32	5.62	**14.62**	206.93

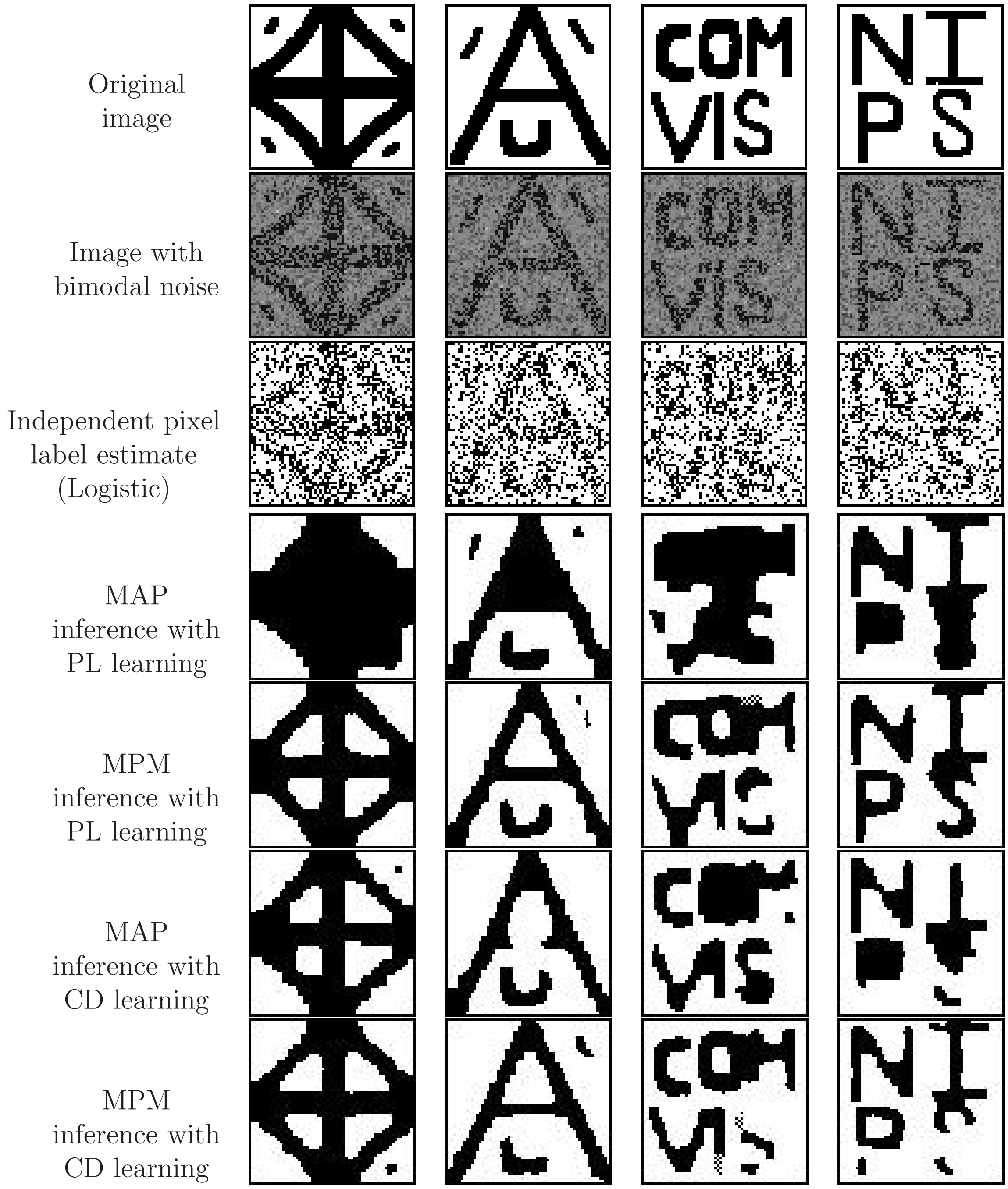


Figure 4.2: Image denoising results on synthetic images with existing parameter learning methods (MAP: Maximum A Posteriori, MPM: Maximum Posterior Marginal, PL: Pseudo-Likelihood, CD: Contrastive Divergence). Both PL and CD yield poor estimates of the parameters.

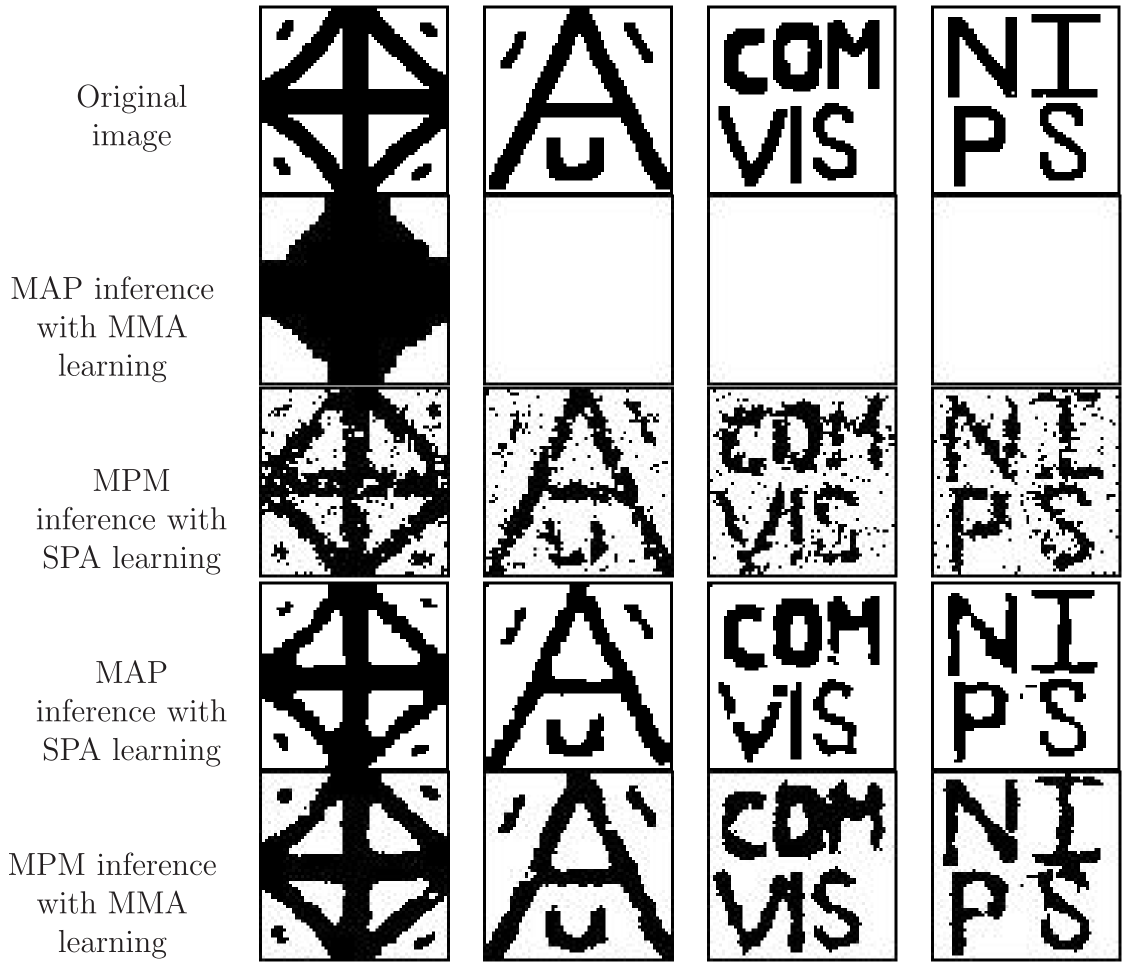


Figure 4.3: Image denoising results on the noisy images shown in Figure 4.2 (MAP: Maximum A Posteriori, MPM: Maximum Posterior Marginal, SPA: Saddle Point Approximation, MMA: Maximum Marginal Approximation.) When an inference algorithm is mismatched to a parameter learning method, the results are poor (rows 2 and 3). For example, oversmoothing is observed for MAP inference with MMA learning. MPM inference yields undersmoothed results with SPA learning. The results are good whenever the parameter learning is matched with the inference procedure (rows 4 and 5), i.e., MAP inference with SPA learning (both use min-cut) or MPM inference with MMA learning (both use BP).

The overall pixelwise errors on the training set and the test set are given in Table 4.1 and Table 4.2 respectively. There are three key observations: First, the MAP inference works best with SPA parameters (both use min-cut [51]), and MPM works best with PMA or MMA parameters (all three use BP), empirically verifying the claim of *learning/inference coupling*. Second, for the MAP inference, SPA based learning is also the most efficient approach. The SPA learning is more than 14 times faster than the next most accurate method, PMA. Last, for the training set, MMA is able to learn reasonable parameters for MPM inference equivalent to PMA (slightly better on the training set and slightly worse on the test set), at almost half the training time for PMA. Note that both PMA and MMA use BP at the learning stage and slightly better results of PMA on the test set may be because PMA returns a single converged estimate of the parameters while in MMA one has to heuristically pick the best set of parameters. Better performance may be expected if a better heuristic is used instead of picking the majority voted parameters.

Table 4.2: Pixelwise classification errors (%) on 200 **test** images (64×64 pixels each). The rows show different parameter learning procedures and the columns show different inference techniques used for two different noise models. To interpret this table, for each noise model, different parameter learning techniques should be compared by fixing a column that corresponds to a fixed inference technique.

		Gaussian noise			Bimodal noise		
Inference methods		MAP	MPM	ICM	MAP	MPM	ICM
Parameter Learning Methods	SPA	**2.49**	7.64	3.98	**5.82**	19.19	14.88
	PMA	2.73	**2.51**	3.91	6.45	**5.48**	17.39
	MMA	34.34	**2.96**	4.11	26.53	**5.70**	16.00
	PL	3.82	3.10	**3.89**	17.69	7.31	22.22
	CD	3.78	2.82	4.09	8.88	6.29	**8.92**
Inference time (Sec)		5.52	90.04	5.20	5.96	113.84	5.20

Three main observations help understand the differences between PMA and MMA. Firstly, since MMA is simply a discretized version of PMA, MMA will remain exact even if the pseudo-marginals converge to erroneous values, provided that the ranking of the labels implied by the pseudo-marginals is the same as that implied by the true marginals. This makes MMA more robust to errors in the estimate of marginals when pseudo-marginals tend to give poor estimates of the true marginals, e.g., in the presence of strong attractions or repulsion between nodes [108].

Secondly, this discretization accelerates parameter learning since we only need to run BP for enough iterations to stabilize the ranking of the labels, not the exact evaluations of the pseudo-marginals. The former is a coarse (low-resolution) computation, but the latter is a fine (high-resolution) computation. Empirically we noticed that most of the changes in the relative ranking of marginals generally occur in first few iterations. This partly explains faster learning through MMA in comparison to PMA as shown in Table 4.1.

Thirdly, while learning the parameters using gradient ascent, MMA gives rise to oscillatory non-convergent behavior. Similar to SPA, this usually requires far fewer iterations of gradient ascent, as typically the *limit-cyclic* behavior in MMA implies that we can stop the gradient ascent iterations after one or two such 'cycles' to obtain sufficiently accurate estimate of the parameters.

An interesting observation is that the MAP inference is very poor with MMA parameters and the same is true for MPM inference with SPA parameters. This further enforces the idea that learning/inference coupling is rooted in minimizing the classification error for a learning/inference pair, rather than maximizing the true likelihood.

As a by-product of this comparison, we find that MPM inference is more robust to the parameters returned by other techniques than MAP which gives significantly worse results with parameters other than SPA and PMA. In addition, the PL and CD parameters generally give bad estimates while ICM does poor inference due to the problem of label initialization. Another thing to note is that even though MPM inference gives least pixelwise classification error for bimodal noise on both training as well as test set (Table 4.1 and Table 4.2), the overall quality of results seems better for MAP inference in comparison to the MPM inference, as seen in the last two rows of Figure 4.2. Perhaps, this is because MPM inference finds the best labeling corresponding to sitewise zero-one loss function in comparison to the MAP inference, which is optimal for global zero-one loss function.

4.6 Discussion

4.6.1 Dynamics of SPA- and MMA-based learning

What is the origin of the complex dynamics of our proposed parameter learning methods (Figure 4.1)? In SPA and MMA we replace the expectations $\langle x_i \rangle_{\theta;\boldsymbol{y}}$ and $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}}$ in the true likelihood gradient with approximations $f_i(\theta;\boldsymbol{y})$ and $g_{ij}(\theta;\boldsymbol{y}) = f_i(\theta;\boldsymbol{y}) f_j(\theta;\boldsymbol{y})$ obtained from MAP and MPM label estimates. These estimates are necessarily discrete values in the set $\{-1,+1\}$, and therefore $f_i(\theta;\boldsymbol{y})$ and $g_{ij}(\theta;\boldsymbol{y})$ are piecewise constant functions of the parameter $\theta \in \Theta$. In other words, the discrete label estimates induce a partition $\{\Theta_k\}$ of parameter space Θ into a disjoint union $\cup_k \Theta_k$ where $f_i(\theta;\boldsymbol{y})$ and $g_{ij}(\theta;\boldsymbol{y})$ are constant within each cell Θ_k. By substitution, the approximate gradient $\boldsymbol{J}(\theta)$ is also piecewise constant for the same partition $\{\Theta_k\}$ of Θ.

As a consequence, integral curves through vector field $\boldsymbol{J}(\theta)$ will be piecewise linear, with “kinks” at the boundaries between cells, say between Θ_k and $\Theta_{k'}$. Our approximate gradient ascent with its finite step size will therefore result in a sequence of parameters along piecewise linear trajectories.

One cannot generally expect these trajectories to terminate, as that would require $\boldsymbol{J}(\theta)$ to be identically zero for all θ in some cell Θ_k. To understand why, consider the double sum in Eq. (4.4) as a product $\frac{1}{2}H(\boldsymbol{x}-\boldsymbol{f})$ of the matrix $H = [\boldsymbol{h}_i(\boldsymbol{y}^m)]$ with vector $\boldsymbol{x}-\boldsymbol{f}$, where $\boldsymbol{x} = [x_i^m]$ and $\boldsymbol{f} = [f_i(\theta;\boldsymbol{y}^m)]$. Now, $\boldsymbol{J}(\theta) = 0$ requires that $\boldsymbol{x}-\boldsymbol{f}$ be in the nullspace of H. Because both training labels and the label estimates are discrete, the components $x_i^m - f_i(\theta;\boldsymbol{y}^m)$ of $\boldsymbol{x}-\boldsymbol{f}$ will be one of the integers -2,0, or +2. But the class of real matrices H which have an integer vector in their nullspace has measure zero, and therefore the possibility that $(\boldsymbol{x}-\boldsymbol{f}) \in \operatorname{nullspace} H$ is both unlikely and unstable. Generally, therefore, the approximate gradient ascent using SPA or MMA will not stop.

In the simpler case of true gradient ascent, for a sufficiently small step size η, the parameter updates converge (without stopping) in a neighborhood of a stationary point of the gradient vector field where the gradient is zero. Why does this ascent converge? Because this gradient vector field is smooth and thus the gradients along the ascent become arbitrarily small near the stationary point, automatically slowing the ascent.

Although our approximate gradients $\boldsymbol{J}(\theta)$ may become small in the vicinity of the true maximum likelihood solution, they cannot become *arbitrarily* small because they are quantized, and therefore the trajectories never slow down beyond some nonzero lower bound. Indeed, our empirical results show a quasi-cyclical behavior of the parameter trajectories. Similar behavior, called *limit cycles*, is common in digital control and signal processing, and arises from quantizing states and coefficients in continuous dynamical systems. Such limit cycles have been observed with small oscillations after a single initial transient or with quasi-periodic transients followed by small oscillations. The small oscillation case corresponds to a parameter trajectory passing in a tight loop through nearby portions of abutting cells, say $\Theta_k, \Theta_{k'}$, and $\Theta_{k''}$, which all have small approximate gradients. But there is no guarantee that cells with small and large approximate gradients will not be adjacent. Thus the observed "wild" transient behavior in Figure 4.1 can arise from several adjacent cells with small approximate gradient linked by cells with large approximate gradient: most of the time is spent in the cells with small approximate gradient, but rapid change occurs in cells with large gradient. To summarize, discretization can account for these limit cycle dynamics.

4.6.2 The role of classification errors in parameter learning

Given these limit cycle dynamics, how can one choose the best parameter along the trajectory? Approximate gradients alone may be misleading, as there may be large approximate gradients nearer to the optimal solution than some small approximate gradients. In true gradient ascent, one may use the likelihood itself as "yard stick" for choosing the best parameter, e.g., at the maximal likelihood observed on the trajectory. The likelihood is also useful in diagnosing pathological dynamics from too large a step size, e.g., if the likelihood decreased significantly. From a dynamical systems perspective, the likelihood exists because the gradient is, by construction, *integrable*.

Instead we have only approximate gradients, which may not be integrable: they may not be the actual gradients of any function. In other words, there may be no approximate likelihood for our approximate gradient!

To overcome this lack of an approximate likelihood, we guide our choice of parameter using the number of classification errors, a widely-employed performance criterion in parameter learning.[2] But what inference algorithm should one use to measure these

[2]Ideally, one would like to minimize the generalization error, i.e., expected error on the test set.

classification errors? In keeping with the coupling of parameter learning and inference first discussed in Section 4.2.2, we compute the number of errors N_E^θ at parameter estimate θ using the inference method used in the gradient approximation Eq. (4.6), i.e., $N_E^\theta = (1/2)\sum_m \sum_{i\in S} |x_i - f_i(\theta)| = (1/2)\|\boldsymbol{x} - \boldsymbol{f}\|$, where $\|\cdot\|$ is the L_1 norm. Formally, this choice is motivated by the following simple bound.

Lemma 1. $\|\boldsymbol{J}(\theta)\| \leq cN_E^\theta$*, for some* $c > 0$.

In other words, the number of errors provides an upper bound on the approximate gradient. Note that matching the inference method used in both the number of errors and the approximate gradient is required in the following proof of the lemma.

Proof. Recall that $\boldsymbol{J}(\theta) = (\boldsymbol{J_1}(\theta), \boldsymbol{J_2}(\theta))$. Using the form of $\boldsymbol{J_1}(\theta)$ in Eq. (4.4),

$$\|\boldsymbol{J_1}(\theta)\| \leq R\, N_E^\theta, \quad \text{where} \quad R = \max_{i,m} \|\boldsymbol{h}_i(\boldsymbol{y}^m)\|.$$

Now, define the pairwise error N_P^θ as,

$$N_P^\theta := (1/2)\sum_m \sum_{i\in S} \sum_{j\in\mathcal{N}_i} |x_i x_j - f_i(\theta;\boldsymbol{y}^m) f_j(\theta;\boldsymbol{y}^m)|.$$

Using the form of $\boldsymbol{J_2}(\theta)$ in Eq. (4.5) with $g_{ij}(\theta;\boldsymbol{y}^m) = f_i(\theta;\boldsymbol{y}^m)f_j(\theta;\boldsymbol{y}^m)$, it is easy to see that,

$$\|\boldsymbol{J_2}(\theta)\| \leq 2Q\, N_P^\theta, \quad \text{where} \quad Q = \max_{ijm} \|\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m)\|.$$

This implies that $\|\boldsymbol{J_2}(\theta)\| \leq 2QdN_E^\theta$, since $N_P^\theta \leq dN_E^\theta$, where d is the maximum degree of the graph, i.e., $d = \max_i |\mathcal{N}_i|$. Combining these results, we have the required result,

$$\|\boldsymbol{J}(\theta)\| = \|\boldsymbol{J_1}(\theta)\| + \|\boldsymbol{J_2}(\theta)\| \leq (R + 2Qd)N_E^\theta.$$

□

This bound is useful in two ways. First, if $\|\boldsymbol{J}(\theta)\|$ is large, then N_E^θ is also large

---

This is a combination of the training error and the complexity of the learned classifier [139]. This is to make sure that overfitting does not occur while training. In our results shown in Table 4.1 and Table 4.2, both training as well as (independent) test data show similar trends for different learning-inference pairs. This indicates that overfitting was not a problem in our experiments.

as verified in the plots in Figure 4.1. Second, if at some θ, N_E^θ is small, $\|\boldsymbol{J}(\theta)\|$ will also be small. Thus, for a suitably small step size η, the parameter change will also be small. This would mean that one will stay in a low error zone for a long period as seen in Figure 4.1.

Indeed, given the importance we put in the number of classification errors, one might ask whether minimizing N_E^θ itself should be used as a starting point for *deriving* parameter learning algorithms. Unfortunately, since the number of errors is piecewise constant in the parameters, its gradient is zero except on a set of measure zero. The number of errors is therefore useless to derive a gradient-based learning algorithm as known from the perceptron learning literature [31].

4.6.3 Related Work

The problem of learning the parameters of loopy discriminative graphs has been addressed before under different paradigms. In the non-probabilistic setting, Taskar et al. [134] learn the model parameters by maximizing the margin. Lecun and Huang [92] have described the sufficient conditions for the training of energy-based (unnormalized) graphical models. In our previous work [81], we proposed the use of penalized pseudo-likelihood that gives reasonable estimates of the parameters. However, this needs hand-tuning of the regularizing constant. Finally, taking the Bayesian view, Qi et al.[118] have argued for integrating the parameters while predicting the labels on a test input instead of using a point estimate of the parameters using maximum likelihood. But integrating the parameters is generally a difficult task.

4.7 Summary

In this chapter we presented an approach for learning the parameters of discriminative field models that uses inference to approximate the gradients used in maximum likelihood learning. We showed that the proposed approximations lead to a limit cycle convergence behavior of the learning procedures. Further, the learned parameters lead to good classification performance so long as the method used for approximating the gradient is consistent with the inference mechanism. We also provided an experimental comparison of commonly used learning and inference techniques for discriminative fields. For MAP inference, SPA based learning was found to be most accurate as well as efficient. Similarly, for MPM inference, PMA and MMA performed best. Although we restricted ourselves to binary fields in this chapter, we have already used maximum marginal approximation to successfully learn more than 3000 parameters for multiclass DRFs, described in the next chapter, applied to object detection [82]. In the future, we would like to evaluate the performance of the proposed approximate parameter learning procedures with conventional MRFs.

Chapter 5

Multiclass Discriminative Fields

5.1 Introduction

So far, we have discussed the binary discriminative fields in Chapter 3, where the label at each site was restricted to be binary. There, we showed that conditional Markov models can provide a principled way of handling many shortcomings in the existing MRF models. Further, we described efficient parameter learning techniques in binary fields in Chapter 4. However, to address more complex real-world vision tasks using the discriminative field framework, we need to extend the current framework to handle mutliclass labelings problems.

There are several applications in computer vision that require the nodes in the graph to take multiple class labels. For example, in semantic scene segmentation task shown in Figure 1.4 (a), the aim it to assign each pixel into one many classes such as *sky*, *water*, *grass* etc. In the case of image denoising applied to a 256 gray-level image, each pixel may take up to 256 labels. In the part-based paradigm of object detection, usually there are more than two characteristic parts that make the full object, and the goal is to label each generic part in the scene as a specific part of the object or background.

It turns out that the extension of binary DRFs to the multiclass case is relatively straightforward. We describe the basic formulation, parameter learning and inference in these models in the following sections. We motivate this formulation in the context of parts-based object detection task. One interesting implication of this application is that the topology of the induced graphs is not fixed to be the same for all images, since in object detection task, sites are not restricted to a regular 2D lattice.

5.2 Multiclass Formulation

In the parts-based paradigm of object detection, given generic parts in the image, our aim is to label each part as a 'specific' part of the object or as a 'background' part. For example, as shown in Figure 5.1 (a) in page 99, we want to detect the phone in the image by detecting its parts. If we are given generic patches in the scene, shown in white squares in Figure 5.1 (b), the goal is to labels each such patch as certain part of the object or background. We will call each patch an image site. Let the observed data from an input image be given by $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$, where $\boldsymbol{y}_i$ is the data from i^{th} site, $\boldsymbol{y}_i \in \Re^c$, and S is the set of all the image sites (i.e. parts). The corresponding labels at the image sites are given by $\boldsymbol{x} = \{x_i\}_{i \in S}$, where $x_i \in \{1, \ldots, C\}$ and C is the number of classes. By convention, the first $(C-1)$ labels correspond to specific object parts and the C^{th} label corresponds to the background class.

We start this formulation by first restating the form of full conditional distribution in discriminative fields described in Chapter 3. We consider only up to pairwise cliques in the graph. Thus, the distribution over the labels $\boldsymbol{x}$ given the observations $\boldsymbol{y}$ can be written as,

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z} \exp\left(\sum_{i \in S} A(x_i, \boldsymbol{y}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I(x_i, x_j, \boldsymbol{y}) \right). \tag{5.1}$$

First we look at the modeling of the association potential. Following the form of the association potential for binary DRFs given in Eq. (3.8) and the associated arguments of Section 3.2.1, the association potential can be easily generalized to the multiclass case as,

$$A(x_i, \boldsymbol{y}) = \sum_{k=1}^{C} \delta(x_i = k) \log P'(x_i = k|\boldsymbol{y}), \tag{5.2}$$

where $\delta(x_i = k)$ is 1 if $x_i = k$ and 0 otherwise. Let $\boldsymbol{f}_i(\boldsymbol{y})$ be a feature vector at each site i. To extend the local discriminative classifier to induce a nonlinear decision boundary in the feature space, a transformed feature vector at each site i is defined as, $\boldsymbol{h}_i(\boldsymbol{y}) = [1, \phi_1(\boldsymbol{f}_i(\boldsymbol{y})), \ldots, \phi_R(\boldsymbol{f}_i(\boldsymbol{y}))]^T$ where $\phi_r(.)$ are arbitrary nonlinear functions.

The first element of the transformed vector is kept at 1 to accommodate the bias parameter. Note that, in the case of object detection, the vector $\boldsymbol{h}_i(\boldsymbol{y})$ encodes the appearance based features of the i^{th} site (or part). To model $P'(x_i = k|\boldsymbol{y})$, in this work we will simply use the multiclass version of the logistic form we chose for the binary DRFs as described in Chapter 3 (Section 3.2.1). This leads to the softmax function in the multiclass case where,

$$P'(x_i = k|\boldsymbol{y}) = \begin{cases} \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{h}_i(\boldsymbol{y}))}{1+\sum_{l=1}^{C-1} \exp(\boldsymbol{w}_l^T \boldsymbol{h}_i(\boldsymbol{y}))} & \text{if } k < C \\ \\ \frac{1}{1+\sum_{l=1}^{C-1} \exp(\boldsymbol{w}_l^T \boldsymbol{h}_i(\boldsymbol{y}))} & \text{if } k = C. \end{cases} \tag{5.3}$$

Here, $\boldsymbol{w}_k$ are the model parameters for $k = 1 \ldots C-1$. For a C class classification problem, one needs only $C - 1$ independent hyperplanes, which may lie in a high dimensional (kernel-projected) space inducing a non-linear decision boundary in the original feature space. Note that this choice of $P'(x_i = k|\boldsymbol{y})$ leads to the unary potential which is linear in features similar to the CRFs given in [86] with a subtle difference that the parameters $\boldsymbol{w}_k$, for $k = C$, are set to $\mathbf{0}$. Also, the form in Eq. (5.3) uses kernel functions to design the unary potential unlike the original CRFs in [86]. Note that other domain-specific choices of $P(x_i = k|\boldsymbol{y})$ are also possible as recently explored in [119]. In the application of object detection, the association potential discriminatively models the individual appearance of each part in the image.

The interaction potential in DRFs predicts how the labels at two sites interact given the observations. Generalizing the interaction potential given for binary DRFs in Chapter 3 (Section 3.2.2), interaction potential for multiclass DRFs can be written as,

$$I(x_i, x_j, \boldsymbol{y}) = \sum_{k=1}^{C} \sum_{l=1}^{C} \boldsymbol{v}_{kl}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}) \delta(x_i = k) \delta(x_j = l). \tag{5.4}$$

Here, $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is the pairwise relational vector for a site pair (i, j), and $\boldsymbol{v}_{kl}$ are the model parameters. Note that in the case of object detection, the vector $\mu_{ij}(\boldsymbol{y})$ encodes the pairwise features required for forcing geometric and possibly photometric consistency in the pair of parts. For undirected graphs, the site pairs are unordered sets implying that $\boldsymbol{v}_{kl} = \boldsymbol{v}_{lk}$ for $k, l = 1 \ldots C$. The from of interaction potential given in Eq. (5.4) is a generalization of the Potts model used commonly in computer vision problems such as image segmentation and restoration [94]. The standard Potts model can be recovered from Eq. (5.4) if $\boldsymbol{v}_{kl} = \mathbf{0}$ when $k \neq l$, and all the elements of the vector $\boldsymbol{v}_{kl}$ are set to zero except the bias term. A more specific but popular form of Potts model is achieved if the bias terms for all the vectors $\boldsymbol{v}_{kk}\ \forall\ k$ are also fixed to be the same. Similar to the interaction potential of the binary DRF, multiclass interaction potential can be seen as a pairwise discriminative model which partitions the pairwise relational feature space (induced by the features $\mu_{ij}(\boldsymbol{y})$) in $C(C+1)/2$ regions.

It is important to note that, to enforce the geometric consistency relationship between parts, the interaction between part labels has to use observed data (e.g. the location of patches). Since, in discriminative fields, the pairwise potential I is a function of observed data, these fields provide a principled way to represent relations between parts by using a random-field framework. In contrast, in the conventional generative MRFs, the conditional distribution over labels is modeled as $P(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x})$, where $P(\boldsymbol{x})$ is used for modeling the label interaction. Since $P(\boldsymbol{x})$ does not allow the use of data $\boldsymbol{y}$ while modeling label interactions, conventional forms of MRFs cannot model the geometric consistency simultaneously with appearance. In the following sections we describe how one can learn the parameters and do inference in multiclass DRFs.

5.3 Parameter Learning

Similar to the binary case, here also we resort to maximum likelihood parameter learning. Let θ be the set of DRF parameters where $\theta = \{\{\boldsymbol{w}_k\}_{k=1\ldots C-1}, \{\boldsymbol{v}_{kl}\}_{k,l=1\ldots C}\}$. Given M i.i.d. labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood,

$$l(\theta) = \sum_{m=1}^{M} \log P(\boldsymbol{x}^m|\boldsymbol{y}^m, \theta).$$

To extend the parameter learning for multiclass DRFs, i.e. $x_i \in \{1, \ldots C\}$, the derivatives of the association and the interaction potentials are given as,

$$\frac{\partial A(x_i^m, \boldsymbol{y}^m)}{\partial \boldsymbol{w}_k} = \{\delta(x_i^m = k) - P'(x_i = k|\boldsymbol{y}^m)\}\, \boldsymbol{h}_i(\boldsymbol{y}^m),$$

$$\frac{\partial I(x_i^m, x_j^m, \boldsymbol{y}^m)}{\partial \boldsymbol{v}_{kl}} = \left\{\delta(x_i^m = k)\delta(x_j^m = l)\right\} \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m).$$

To learn the parameters using gradient ascent, the derivative of the log-likelihood, after some algebraic manipulations, can be written as,

$$\frac{\partial l(\theta)}{\partial \boldsymbol{w}_k} = \sum_m \sum_{i \in S^m} \Big(\delta(x_i^m = k) - \langle \delta(x_i = k) \rangle\Big) \boldsymbol{h}_i(\boldsymbol{y}^m), \tag{5.5}$$

$$\frac{\partial l(\theta)}{\partial \boldsymbol{v}_{kl}} = \sum_m \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} \Big(\delta(x_i^m = k)\delta(x_j^m = l) - \langle \delta(x_i = k)\delta(x_j = l) \rangle\Big) \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m), \tag{5.6}$$

where $\langle . \rangle$ denotes expectation with respect to the distribution $P(\boldsymbol{x}|\boldsymbol{y}^m, \theta)$. Generally the expectation in (5.5) and (5.6) cannot be computed analytically even for moderately sized problems due to the combinatorial number of elements in the configuration space of labels $\boldsymbol{x}$. Following the discussion in Chapter 4, the expectations can be replaced by pseudo-marginals obtained from Belief Propagation (BP) leading to Pseudo-Marginal Approximation (PMA). Another approach based on Saddle Point Approximation (Section 4.3.3) is difficult to implement since the global maximum cannot be computed using graph min-cut in the multiclass case. Recently, Boykov et al. [18] have proposed promising algorithms, e.g., Alpha-Expansion to obtain good approximations of the global maximum for multiclass labeling problems. We leave further exploration of the behavior of such algorithms with SPA-based parameter learning as a future work. However, we can use the Maximum Marginal Approximation (MMA), which uses thresholded pseudo-marginals to approximate the expectations (Section 4.3.4). MMA may be preferred in comparison to PMA, due to more robustness to pseudo-marginals if they are poor estimates of the true marginals, and faster computations. According to MMA, if

$$\tilde{x}_i{}^m = \arg\max_{x_i} \log P(x_i|\boldsymbol{y}^m, \theta), \tag{5.7}$$

then the derivatives are given by,

$$\frac{\partial l(\theta)}{\partial \boldsymbol{w}_k} = \sum_m \sum_{i \in S} \Big(\delta(x_i^m = k) - \delta(\tilde{x}_i^m = k)\Big) \boldsymbol{h}_i(\boldsymbol{y}^m), \tag{5.8}$$

$$\frac{\partial l(\theta)}{\partial \boldsymbol{v}_{kl}} = \sum_m \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \Big(\delta(x_i^m = k)\delta(x_j^m = l) - \delta(\tilde{x}_i^m = k)\delta(\tilde{x}_j^m = l)\Big) \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m). \tag{5.9}$$

These approximations were used for learning several thousand parameters in the object detection tests described in Section 5.5.

5.4 Inference

As argued in Section 4.5, using the same approximation for parameter learning and inference tends to minimize the classification error. Since we use MMA for parameter learning, in this work we will focus on obtaining sitewise maximum marginal (MPM) estimates of the labels at each node. However, for graphs containing cycles, exact marginal estimates cannot be guaranteed using Belief Propagation (BP) [113]. The loopy version of BP has been shown to return reasonably good estimates for a variety of problems [45]. The sum-product version of BP remains the most popular way of computing the sitewise and pairwise marginals[1]. In loopy BP, messages are passed between each node and its neighboring nodes iteratively until convergence (if possible). The message passing updates of BP for multiclass DRFs can be given as,

$$\text{Let} \;\; \phi_i(x_i) = \exp\left(A(x_i, \boldsymbol{y})\right) \;\; \text{and} \;\; \psi_{ij}(x_i, x_j) = \exp\left(I(x_i, x_j, \boldsymbol{y})\right),$$

$$m_{ji}(x_i) = \alpha \sum_{x_j} \phi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}_j \setminus i} m_{kj}(x_j) \tag{5.10}$$

where α is the normalizer, $m_{ji}(.)$ is the message from node j to node i and $\setminus$ is set exclusion operator. Finally, the marginal probability (or belief) at each node is computed as,

$$b_i(x_i) = \kappa \phi_i(x_i) \prod_{j \in \mathcal{N}_i} m_{ji}(x_i) \tag{5.11}$$

The convergence of loopy BP updates is not guaranteed and oscillations may occur [108]. Many convergent alternatives have been proposed to BP, e.g. CCCP

[1] One can use the max-product version of loopy Belief Propagation to find the MAP estimates.

procedures from Yuille [158] and Belief Optimization from Welling and Teh [148]. One can extend the BP algorithm to a more generic algorithm called Generalized Belief Propagation (GBP) by replacing the Bethe Free Energy approximation by Kikuchi Free Energy [157]. GBP has been shown to be more accurate than BP but it is usually slower due to the presence of bigger cliques in the approximation. Regarding the appropriateness of BP, when pairwise potentials are either repulsive (i.e. potentials that prefer the neighboring nodes to take different labels) or mixed (both attractive and repulsive potentials) then BP seems to have more problems with convergence and usually returns bad estimates [142]. Several alternatives to BP have been proposed in the literature, e.g. Expectation Propagation (EP) [106] and Tree-Based Reparameterization (TRP) [142][74], which yield better results than BP under various circumstances. In this work, we primarily use BP to obtain MPM estimates of the labels and leave the exploration of other techniques as a future work.

5.5 Object Detection Task

Object detection has been a long standing problem in computer vision. Even though several promising approaches have been proposed in the literature, generic category-level object detection under complex variations in appearances, object deformations and occlusions is still a challenging problem. The proposed multiclass framework has three key advantages: First, during classification, it probabilistically enforces the appearance of individual parts and geometric consistency between parts simultaneously, thus making the classification robust to ambiguities and deformations. Second, the part appearances as well as the relations between parts are modeled using local discriminative models, thus avoiding the need of learning generative models which may be hard to learn for complex data. Last, the final classification is obtained using efficient inference over graphs carried out using existing techniques without requiring exhaustive search in the solution space.

The main approaches to solve the object detection problem can be divided broadly into two categories. Finding an object in the scene either by scanning a window over the entire image (possibly at different orientations and scales), or by detecting different parts of the object. Several detection techniques attempt to detect the object as a 'whole' by applying a classifier to a window scanned over the image [124][128][140][98] [107]. Even though these approaches have been successfully applied to detect faces and cars etc., they tend to have problems when objects are occluded, or when they undergo significant deformations or articulations. As will be explained later, these issues can be handled naturally in the multiclass DRF framework without needing any extra modeling or computational effort.

The 'parts-based' approach to object detection is based on the idea of identifying 'characteristic' parts of the object in the image. The parts-based techniques that first detect the object parts purely on the basis of their appearance and then refine these part detections using geometric reasoning [98][96][57] may yield inaccurate results if the appearance of the parts in images is noisy or ambiguous. So, it is desirable to have techniques that detect the parts not only on the basis of their individual appearance but also by enforcing geometric relationship of the parts *simultaneously*. This can be achieved by interpreting the part detection as a labeling problem in which labels (i.e. parts) of the object are dependent on other labels. Thus, this problem can be viewed as a classification problem in a *random field* framework. This idea forms the basis of this work.

A number of researchers have proposed graph-based techniques to model shape and appearance of objects simultaneously [33][24][146][35]. In [33][24], the authors assume a tree-structured graph over the object parts and look for the best possible match in the image. However, restricting the graph to a tree is generally not enough to capture the structure of the object. In addition, the tree-structured object models are unable to handle occlusions. There exist other parts-based techniques which view the detection problem as an explicit search over the image parts [146][35][32]. A graph is formed over the object parts which allows one to model appearance and relations between the parts simultaneously. But the final classification is carried out by searching the solution space which is $O(N^P)$ problem where N is the number of total parts in the image, and P is the number of object parts. For computational tractability, N and P are restricted to be small (typical choice is 20 for N and 5 for P). On the other hand, our DRF-based approach defines a graph over the image sites and detection task is seen as labeling individual image parts. At classification time, this has a computational complexity of $O(NP^2)$ which allows efficient inference even if N is in hundreds as we will show later in experimental results. In [36], the authors have recently suggested a modified star model to overcome the computational complexity $O(N^P)$ of their previous model [35]. The new model has a complexity of $O(N^2P)$, but it needs one of the parts in the image, called 'landmark' part, to be always present in the image. In fact, this model is essentially the same as the tree-structured model presented in [33], and it inherits all the drawbacks of that model.

The graph based techniques usually assume only a single instance of the object in the scene [33][24][35]. To detect multiple instances of objects in the scene, either the number of instances should be known a-priori, or a threshold needs to be applied to the candidate scores. On the other hand, the DRF based framework allows detection of multiple instances naturally without needing any such information. Finally, all the graph based techniques of object detection operate exclusively in a generative framework in which a lot of resources may be spent on modeling the generative models for complex part appearances and part relations which are not particularly relevant to final classification task. Moreover, learning realistic class density models may become even harder when the training data is limited. To the best of our knowledge this work presents the first graphical model based approach to object detection that models the part appearances and their geometric relations in a discriminative framework. This effect is unique to the DRF framework (in a random field setting) since DRFs allow the use of observed data in pairwise potentials also.

5.5.1 Experiments

We conducted experiments with object detection on synthetic data to verify the applicability of our multiclass formulation of the DRFs to object detection. We took the *part-based* approach to object detection in which at first some salient *parts*[2] are found in an input image. Then, on the basis of appearance and configuration of these parts, the parts that belong to the object are filtered from the non-object or *background* parts. These experiments were constructed to validate specific properties of the approach, such as the ability to discriminate object from background, to detect multiple instances, robustness to occlusion, and to learn deformation models.

To extend the DRF framework to this application, each part of the object is labeled as a separate class while all the parts that do not belong to the object are labeled as *background class*. Suppose the object has $C-1$ parts. Then the object detection problem can be seen as a C class classification problem where all the background parts are assigned to the C^{th} class. The shape of the object, which may be deformable, defines the statistical pairwise geometric relationship between the parts that belong to the object. But the pairs that have either one or both the parts from

[2] The term *part* will be used in this thesis to represent a patch in the image extracted using some generic interest point and region detector.

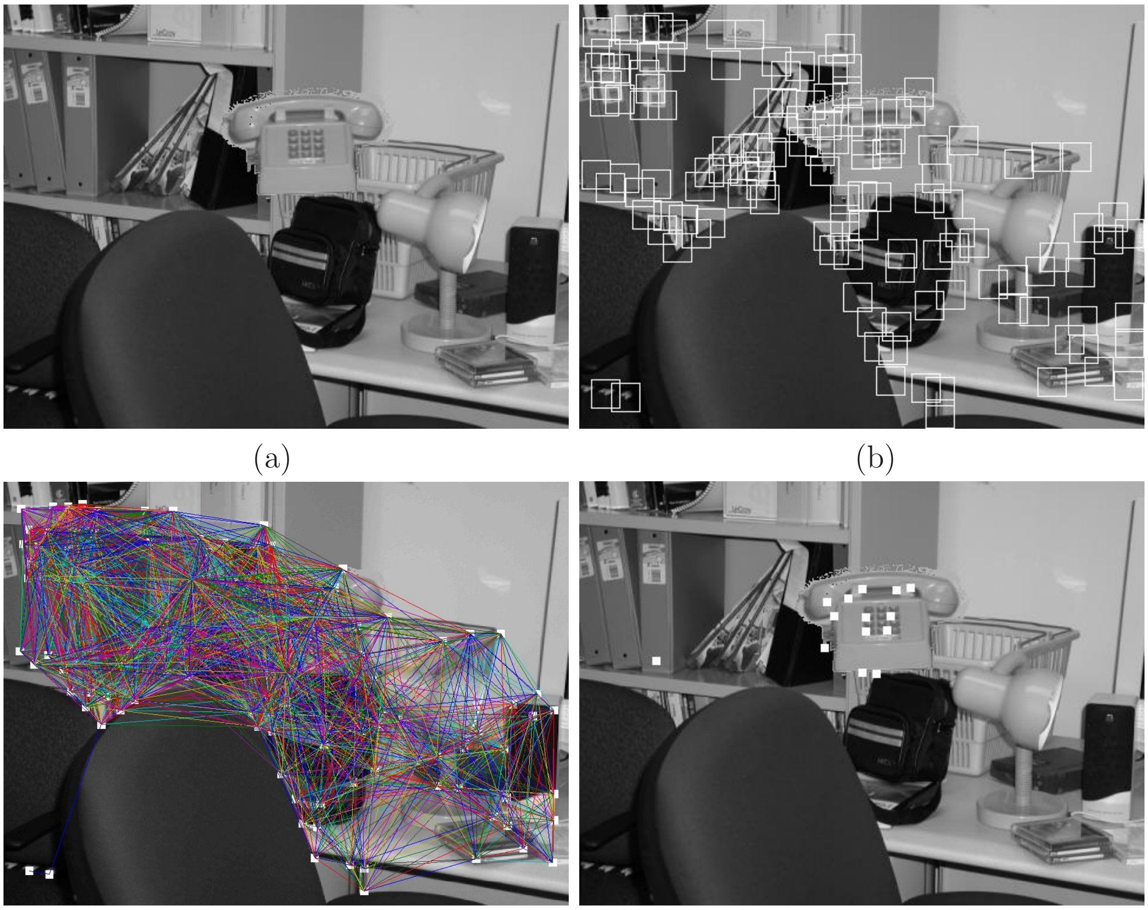
the background are geometrically unconstrained. We call such pairs as 'background pairs'. As discussed in Section 5.2, the interaction potential partitions the space of pairs by using the hyperplanes in pairwise feature space. To separate the background pairs from the object pairs using a single hyperplane we must have $\boldsymbol{v}_{kl} = \boldsymbol{v}_b$ if $k = C$ or $l = C$ or both. Further, without loss of generality $\boldsymbol{v}_b$ can be set to $\mathbf{0}$, since to partition a C class problem we need to learn only $C-1$ independent hyperplanes. The main advantage of the DRF framework is that it allows modeling of the appearance of each part (through association potential) and the geometric relationship of part pairs (through interaction potential) *simultaneously* in a random field setting. This is important for developing a robust object detection framework that can allow trade off between part appearances and their geometrical relationship in a principled manner. For real-world applications, using first appearances and then enforcing geometric constraints is usually misleading due to noise or statistical variations in images.

In the first set of experiments, the aim was to,

1. illustrate the detection framework under DRFs using a rigid object,
2. verify the performance under object occlusions, and
3. validate the capability of the framework to deal with multiple objects in the scene.

In these experiments, the task was to detect a rigid object i.e., a phone in a cluttered scene (Fig. 5.1 (a)). Synthetic training and test data sets were generated by taking a mask of the phone and embedding its affine distortions in 300 random office backgrounds. $\pm 10\%$ percent variation was allowed in scale and shear. For each training image, interest points were detected using the Harris corner detector [3] and a patch of size 25×25 pixels around each interest point was called a part as shown in Fig. 5.1 (b). A graph was generated using these patches as nodes as shown in Fig. 5.1 (c). All patches within a specified radius (135 pixels in this case) from a patch were called neighbors of that patch. Note that the resulting graph is no longer a regular grid lattice and that each node in the graph will usually have different number of neighbors. In this work, we assumed a uniform distribution over the graph structures which leads to 'averaging' over all the graphs in the training images. We intend to

[3] One may use other more powerful interest point (and interest region) detectors that are invariant to affine deformations of the object [96][66][104].



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

Figure 5.1: Detection of a rigid object (phone) in a cluttered scene. (a) Input image. (b) Patches extracted from the input image. (c) Graph joining patches with their neighbors. (d) Detection results. Patches that are classified as object parts are shown highlighted.

explore in the future if better distributions could be learned over the graph structure itself.

The appearance based features used in the association potential, $\boldsymbol{f}_i(\boldsymbol{y})$, were computed based on the gradient orientation histograms weighted by the gradient magnitude and quadratic transformations were used to compute $\boldsymbol{h}_i(\boldsymbol{y})$. The pairwise features, $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$, were just the distances between the part centers. In the future, some more features e.g., joint appearance may also be added. For this problem, the number of classes, C, was fixed to 17 based on the object part-detector output while training. The model had overall 3230 parameters which were learned successfully using the Maximum Marginal Approximation (MMA), which uses thresholded

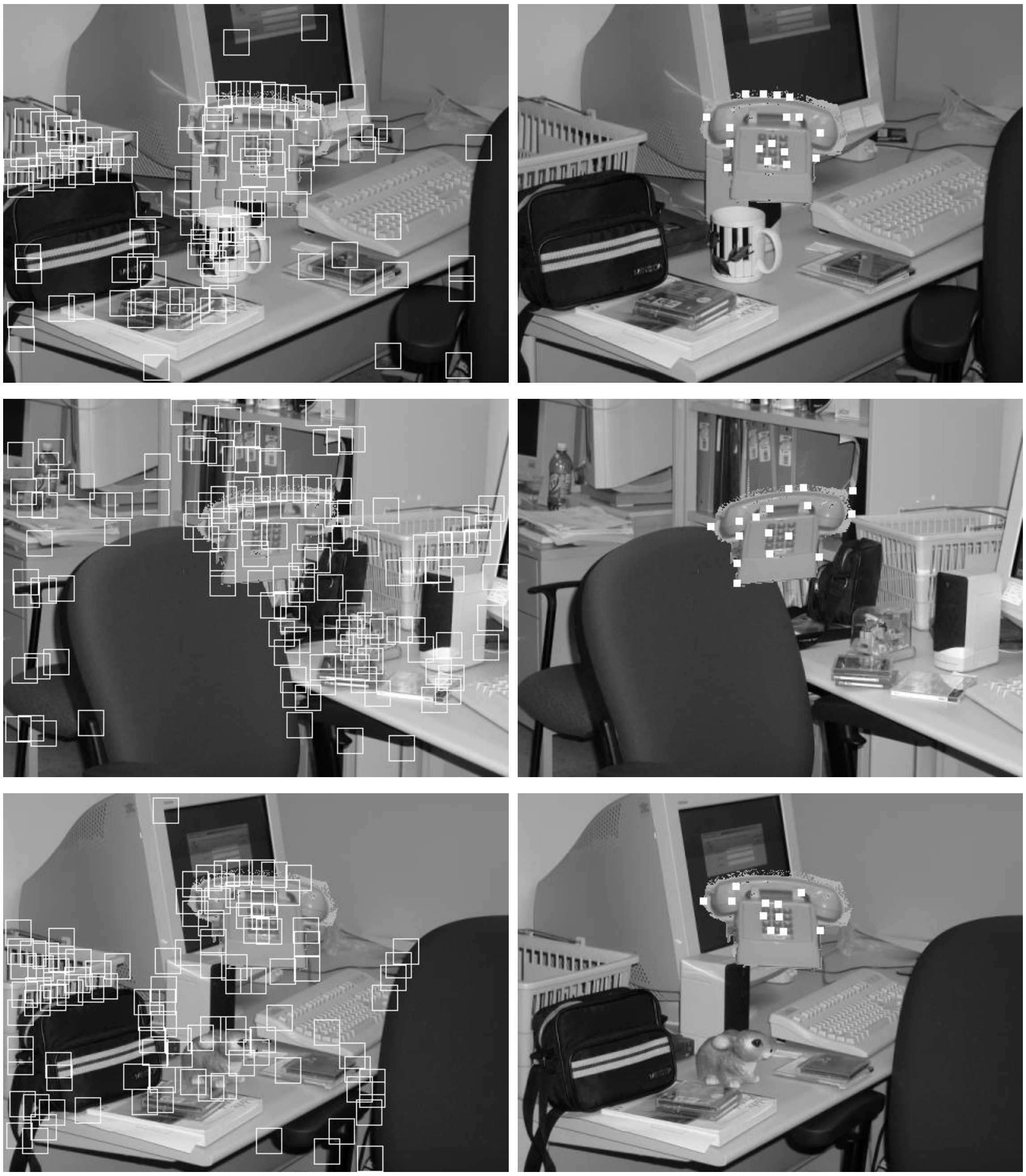


Figure 5.2: Some more examples of the phone detection with different affine transformations of the object in varying backgrounds. Left: Input images along with the extracted patches. Right: Highlighted patches that are labeled as phone parts.

pseudo-marginal estimates obtained using BP as described in Section 5.3. The association parameters $\boldsymbol{w}_k$ were initialized from the softmax classifier parameters, while the interaction parameters $\boldsymbol{v}_{kl}$ were initialized at 0. At test time, BP was used to infer the optimal labeling of the parts. In Fig. 5.1 (d) all the parts that were labeled as any of the object parts are shown highlighted. To generate the final object hypothesis, one may use a simple postprocessing step (e.g., location based clustering) to filter any isolated false positives. Training took about 50 iterations and two hours, while the average time taken for inference was 1.35 sec per image on a 2GHz machine. Some more examples of the phone detection with different affine transformations of the object in varying backgrounds are shown in Figure 5.2.

To demonstrate the effect of occlusion, we synthetically blocked the right half of the phone and the DRF detection results are shown in the left image in Fig. 5.3. To verify multiple instance detection under this framework, two affine distorted versions of the phone were embedded randomly in the scene and the corresponding detection results are shown in the right image in Fig. 5.3. Note that no information about the number of objects in the scene was known, and the same learned model described in the previous paragraph was used for detection in both experiments.

Figure 5.3: Toy examples constructed to demonstrate detection with occlusion (left), and with multiple object instances in the scene (right) using the same learned model.

In the second set of synthetic experiments, we explored the answers to two questions regarding the DRF model applied to object detection:

1. Can it learn all the deformations of a deformable object in a single model?
2. Can it automatically learn to trade off the appearance with the geometric con-

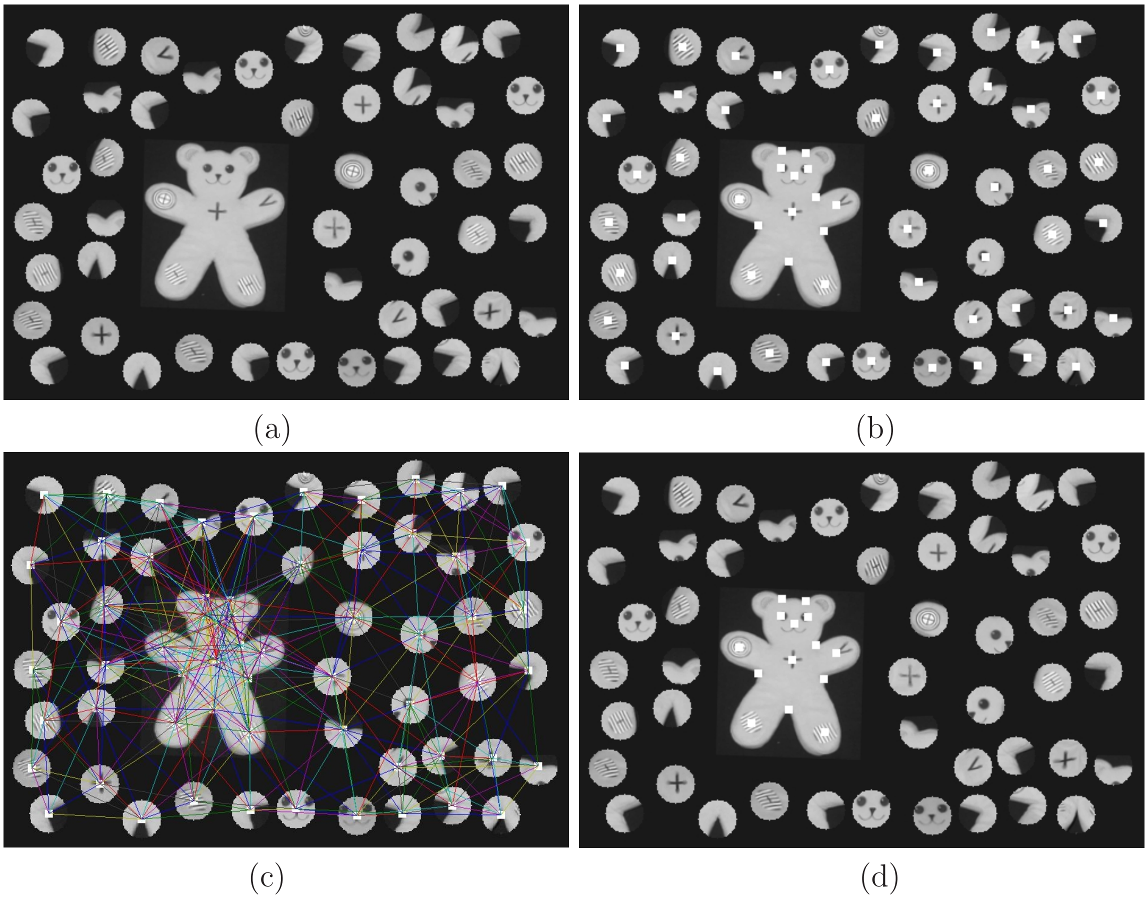


Figure 5.4: Detection of a deformable object (teddy) in a synthetic scene in which the object patches are inserted as background patches to confuse the appearance based detection. (a) Input image. (b) Interest points extracted from input image. (c) Graph joining patches with their neighbors. (d) Detection results. Patches that are classified as object parts are shown highlighted. Note that DRF was able to ignore the background patches even though their local appearances are the same as the object patches.

	object	background
object	0	4287
background	0	13136

(a) Softmax

	object	background
object	4258	29
background	30	13106

(b) DRF

Figure 5.5: Confusion matrices for all the patches in the test set using different techniques. The softmax classifier uses just the appearance of each patch while the DRF model uses both appearance and the geometric configuration between patches to classify different patches. Note that for all the affine and articulated deformations in the object, only a single DRF was learned to account for all these variations.

straints between parts in presence of ambiguities?

For these experiments, we chose an articulated toy object shown in (Figure 5.4) in which different joints of the object could be deformed independently. We generated training and test sets by embedding affine transformations of different deformed versions of the object in synthetic backgrounds. To confuse the appearance, we randomly inserted the object patches in the background (Figure 5.4 (a)). Clearly, if appearance alone were used to classify the parts, everything would be classified as background. This is because there are many more background patches than the object patches in the training set and a discriminative classifier will try to reduce the classification error by simply assigning all the object patches to the background class. However, the geometric relationship along with the appearance should be able to restrict the choice of parts being from the object. This is exactly what is exploited by the DRF as shown by the result in Figure 5.4 (d). Some more results on different deformations of the object are given in Figure 5.6 and Figure 5.7. In Figure 5.5 we show the confusion matrices displaying patchwise detection results on the test set using the multiclass softmax classifier and the DRF model. Clearly, the softmax classifier that uses only appearance classified all the patches as background while the DRF classifies the background and the object patches with very high accuracy. Note that for all the affine and articulated deformations in the object, only a single DRF was learned to account for all these variations. The training needed about 50 iterations and less than one hour while the testing took on an average 0.24 sec to process each image on a 2 GHz machine.

5.6 Summary

In this chapter, we presented an extension of binary DRFs to multiclass labeling tasks. Further, we showed the application of these fields on parts-based object detection. This application is particularly interesting as the graphs induced in this case can be of arbitrary topology instead of being restricted to 2D rectangular grids. The parameters of the field are learned using efficient maximum marginal approximations and inference is carried out using loopy belief propagation. The proposed formulation allows simultaneous discriminative modeling of the appearance of individual parts as well as the geometric relations among them. The conventional Markov Random Field (MRF) formulations cannot be used for this purpose because they do not allow the use of data while modeling interaction between labels, which is crucial for enforcing geometric consistencies between parts. The proposed technique can handle object deformations, occlusions and multiple-instance detection in a single trained model with no added computational efforts. We demonstrated the efficacy of this approach through controlled experiments on rigid and deformable synthetic toy objects. Clearly, the next important step is to apply this framework to the real-world detection tasks and compare its performance with the existing techniques. This has been left as a topic of future exploration. Scale invariance can be achieved in this framework by choosing scale invariant unary and pairwise features, or by using the cliques of size three or more in the model.

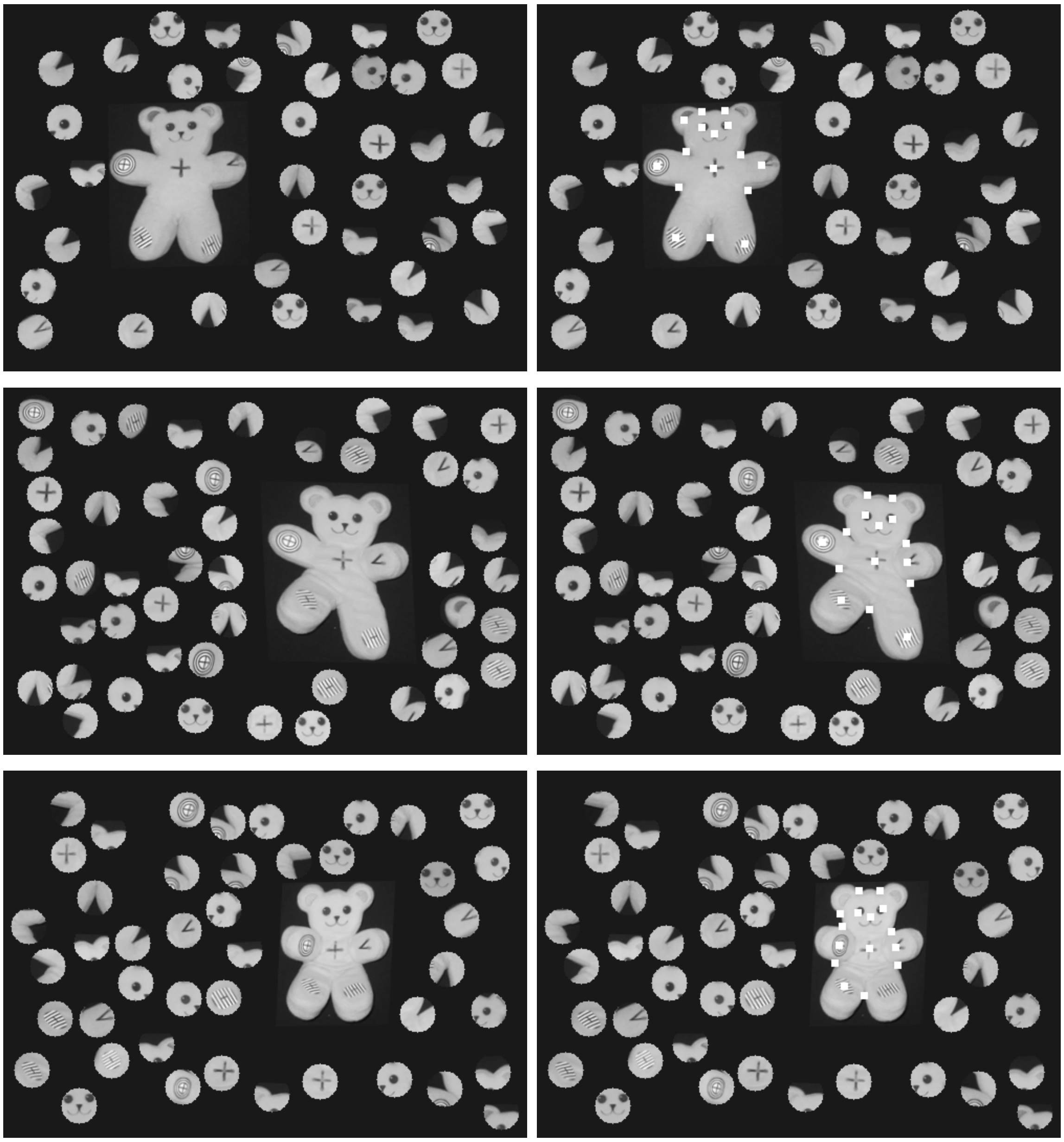


Figure 5.6: Synthetic deformable object detection experiments to verify the advantages of simultaneous modeling of appearance and spatial interactions between patches. Left column: Various deformations of the object. Right column: Corresponding DRF detection results.

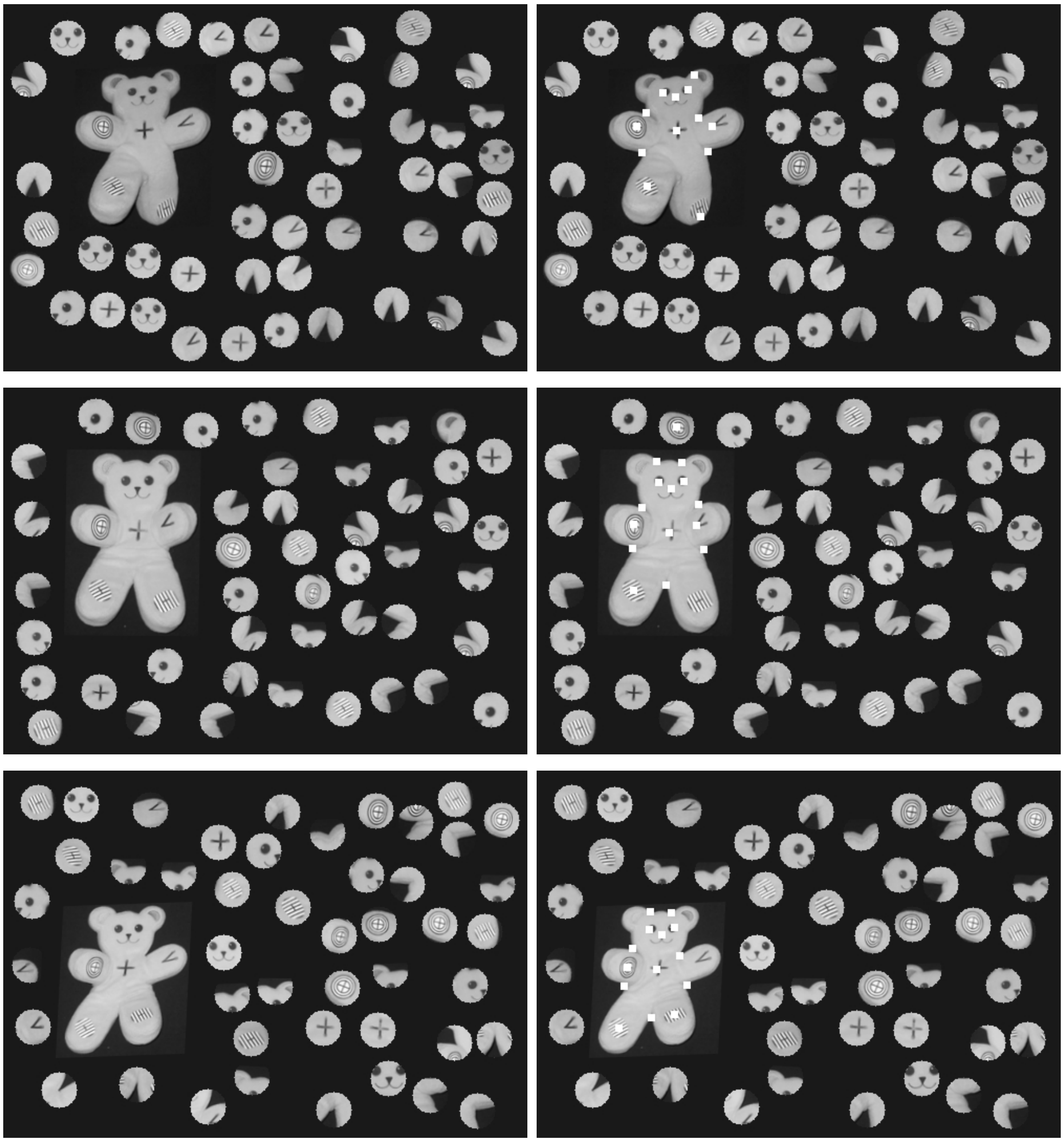


Figure 5.7: Some more example deformations of the synthetic deformable object. Left column: Various deformations of the object. Right column: Corresponding DRF detection results.

Chapter 6

Hierarchical Discriminative Fields

6.1 Introduction

So far, we have discussed spatial interactions in natural images at pixel, block or patch level for binary or multiclass classification problems. However, the problem of detecting and classifying bigger regions and objects in images is a challenging task due to ambiguities in the appearance of visual data. The use of spatial context at a larger scale can help alleviate this problem significantly. For example, in Figure 6.1, the sky and the water subregions may locally look very similar but their relative spatial configuration removes the ambiguity regarding their identity.

There are different levels of context one would like to use to improve classification accuracy. For instance, for pixelwise image labeling problem, the local smoothness of pixel labels will be a local context. On the other hand, global context will refer to the fact that the image regions follow probable configurations e.g., sky tends to occur above water or vegetation (Figure 6.1). We denote this type of global context by *region-region* interaction. Similarly, for the problem of parts-based object detection, local context will be the geometric relationship among parts of an object while the relative spatial configurations of different objects will provide the global contextual information. This type of global context is denoted by *object-object* interaction. As shown in Figure 6.1, the keyboard and the mouse may be very hard to detect because of their impoverished appearance but the relative configuration of monitor, keyboard and mouse helps disambiguate the detection. Similarly, car detection is much easier given the configuration of building and road (Figure 6.1). In this case, the global

Figure 6.1: Example images demonstrating that scene context is important in different domains to achieve good classification even though the local appearance is impoverished. From left: first and second - scene labeling (*region-region* interaction), third - *object-region interaction*, fourth - *object-object interaction*.

context is provided by *object-region* interaction.

In the past, global context has been advocated for the problems of pixelwise image labeling [131][56]. On the other hand, several techniques have been proposed that use context for object detection in images [137][39][20][136][120]. All these techniques are either specifically tuned for a certain application domain or use context only at a specific level. In this work we present a framework that provides a unified approach to incorporate the local as well as the global context of any of the three types in a single model.

In [131], Singhal et al. presented an approach for labeling each region in the scene where an input image is first processed by a number of individual material detectors (e.g., Neural Network based sky detector). These detectors give rise to belief maps indicating the likelihood of a region being a certain material. Based on the fusion of all individual belief maps, the original image is segmented into regions that are supposed to be homogeneous. This segmentation map along with the combined materials belief map is passed to the spatial context-based belief refinement module. The refinement process starts with a seed region (with the largest belief) and proceeds sequentially where each time a new region is picked (depending on its belief ranking) and a label is assigned given the labels of the previous regions. This approach has two main problems. First, since the segmentation map is fixed, the technique will give wrong results if the segmentation is erroneous. Second, since the region refinement takes place in a sequential fashion, this approach will give spurious results if the previously labeled regions were assigned wrong labels.

Markov Random Fields (MRFs) provide a sound theoretical approach to model contextual interactions among different components simultaneously [49]. However, a variety of applications require image observations to model such interactions. For example, different natural regions in a scene, or parts of an object are related through geometric constraints. Traditional MRFs do not allow the use of observed data to model interactions between labels. The conditional fields provide a principled approach to incorporate these data-dependent interactions. In our hierarchical approach, each layer is modeled as a DRF. Another advantage of DRFs over the traditional MRFs is that they use a discriminative approach for classification rather than spending the efforts in modeling the generation of the observed data.

Based on conditional fields, He et al. [56] presented an approach for labeling image pixels into a predefined set of class labels. The model is a product combination of individual models, each providing labeling information about different aspects of the image: a pixelwise discriminative classifier (Neural Network in this case) that looks at local image statistics, regional label features that look at local label patterns, and global label features that look at large, coarse label patterns. The classifier is learned separately from the label features. One problem with this approach is that the effective clique in the induced field is of the size of input image, making it computationally intractable to do learning or inference in such model. The authors use contrastive divergence to do approximate learning in this model, and inference is carried out using Gibbs sampling. Another key limitation of this approach is that it is specifically designed for pixelwise image labeling problems and it cannot handle other applications such as contextual object detection within the same framework.

Fink and Perona suggested Mutual Boosting to incorporate contextual information to augment object detection [39]. Also, Torralba et al. [136] have combined boosting with CRFs to learn the graph structure and its potentials for contextual object detection. However, neither technique provides a guiding framework for handling different levels of context for different applications in the same model.

In computer vision, various forms of hierarchical models have been suggested under both undirected [68][103] as well as directed [16][34] graph paradigms. However, these models have been restricted to simple local contextual information such as label smoothing to obtain good segmentation. They do not use any high level global context. In addition, all the previous hierarchical models were based on MRFs. To the best of our knowledge, this work presents the first attempt on modeling a hierarchy of conditional fields[1].

[1]A shorter version of this chapter will appear in IEEE International Conference on Computer Vision (ICCV), 2005 [83].

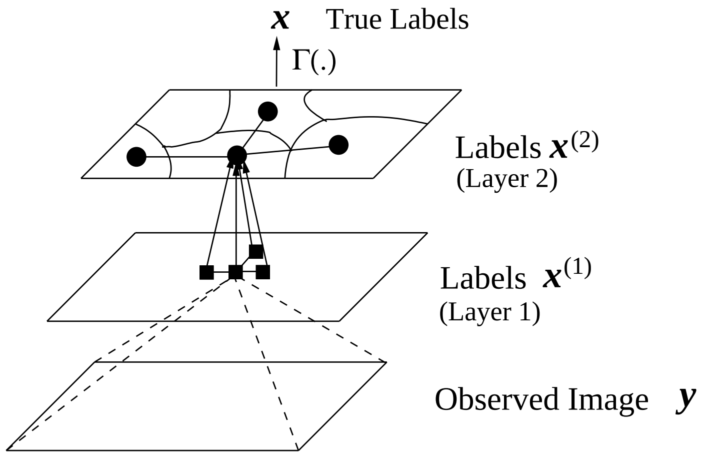


Figure 6.2: A simple illustration of the two-layer hierarchical field for contextual classification. Squares and circles represent sites at the two layers. Only one node along with its neighbors is shown for each layer for clarity. Layer 1 models short-range interactions while layer 2 long range dependencies in images. The true labels x are obtained from the top layer by a simple replication mapping $\Gamma(.)$. Note that the partition shown in the top layer is not necessarily a partition on the image.

6.2 Hierarchical Framework

In this work, we are interested in modeling interactions in images at two different levels. Thus, we propose a two-layer hierarchical field model as shown in Figure 6.2. Note that, in any of the two layers, the induced graph's topology is not restricted to regular 2D grid locations. In this model, each layer is a separate discriminative field. The first layer models short range interactions among the sites such as label smoothing for pixelwise labeling, or geometric consistency among parts of an object. The second layer models the long range interactions between groups of sites corresponding to different coherent regions or objects. Thus, this layer can take into account interactions between different objects (monitor/keyboard) or regions (sky/water).

The two layers of the hierarchy are coupled with directed links. A node in layer 1 may represent a single pixel or a patch while a node in layer 2 represents a larger homogeneous region or a whole object. Each node in the two layers is connected to its neighbors through undirected links. In addition, each node in layer 2 is also connected to multiple nodes in layer 1 through directed links. In the present work we restrict each node in layer 1 to be connected to only one node in the layer above. As noted by Hinton et al. [59], with respect to hierarchical MRFs, the use of directed links between the two layers, instead of the undirected ones, avoids the intractability of dealing with a large partition function. Each layer being a discriminative field, any node in layer 1 can potentially use arbitrary features from the whole image to compute its bias. The top layer uses the output of layer 1 as input through the directed links.

6.2.1 Basic Formulation

Let the observed data from an input image be given by $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$, where $\boldsymbol{y}_i$ is the data from i^{th} site, $\boldsymbol{y}_i \in \Re^c$, and S is the set of all the image sites. We are interested in finding the labels, $\boldsymbol{x} = \{x_i\}_{i \in S}$, where $x_i \in \mathcal{L}$ and $|\mathcal{L}|$ is the number of classes. For image labeling, a site is a pixel and a class may be *sky*, *grass* etc., while for contextual object detection, a site is a patch and a class may refer to objects e.g., *keyboard* or *mouse*. The set of sites in layer 1 is $S^{(1)}$ such that $S^{(1)} = S$, while that in layer 2 is denoted by $S^{(2)}$. The nodes in layer 2 induce a partition over the set $S^{(1)}$ such that a subset of nodes in layer 1 correspond to one node in layer 2. Formally, a partition h is defined as $h : S^{(1)} \rightarrow S^{(2)}$ such that, if $S_r^{(1)}$ is a subset of nodes in layer 1 corresponding to node $r \in S^{(2)}$, then $S^{(1)} = \bigcup\limits_r S_r^{(1)}$ and $S_r^{(1)} \cap S_s^{(1)} = \phi \;\; \forall\, r, s \in S^{(2)}$. Let the space of all partitions be denoted as $\mathcal{H}$. This partition should not be confused with an image partition, since it is defined over the sites in $S^{(1)}$, which may not correspond to the image pixels (e.g., in object detection, where sites are random image patches). Let the labels on the sites in the two layers be given by $\boldsymbol{x}^{(1)} = \{x_i^{(1)}\}_{i \in S^{(1)}}$ and $\boldsymbol{x}^{(2)} = \{x_r^{(2)}\}_{r \in S^{(2)}}$, where $x_i^{(1)} \in \mathcal{L}^{(1)}$ and $x_r^{(2)} \in \mathcal{L}^{(2)}$, where $\mathcal{L}^{(2)} = \mathcal{L}$. The nodes in layer 1 may take pseudo-labels that are different from the final desired labels. For instance, in object detection, a node at layer 1 may be labeled as 'a certain part' of an object rather than the object itself. In fact, the labels at this layer can be seen as noisy versions of the true desired labels .

Given an image $\boldsymbol{y}$, we are interested in obtaining the discriminative distribution $P(\boldsymbol{x}|\boldsymbol{y})$ over the true labels. Given $\boldsymbol{y}$, let us define a space of valid partitions, $\mathcal{H}_v$, such that $\forall \;\; h \in \mathcal{H}_v$, $x_i = x_r^{(2)} \;\; \forall\, i \in S_r^{(1)}$, where $r = h(i)$. This implies that multiple nodes in layer 1 form a hypothesis about a single *homogeneous* region or an object in layer 2. An example illustrating the idea of valid partitions is given in Figure 6.3. Further, we define a replication mapping, $\Gamma(.)$, which takes any value (discrete or continuous) on node r and assigns it to all the nodes in $S_r^{(1)}$. Thus, given a partition $h \in \mathcal{H}_v$, and the corresponding labels $\boldsymbol{x}^{(2)}$, the labels $\boldsymbol{x}$ can be obtained simply by replication. This implies, $P(\boldsymbol{x}|\boldsymbol{y}) \equiv P(\boldsymbol{x}^{(2)}|h, \boldsymbol{y})$ if $h \in \mathcal{H}_v$. However, given an observed image $\boldsymbol{y}$, the constraint $h \in \mathcal{H}_v$ is too restrictive. Instead, we define a distribution, $P(h|\boldsymbol{y})$,

Figure 6.3: An example illustrating the idea of valid partition space, $\mathcal{H}_v$. The partition shown in the left image represents a valid partition because each region contains all the sites (pixels in this case) from a single class. Since it is not true for the partition shown in the right image, it is not a valid partition. Clearly, it is highly improbable that a random partition will be a valid partition.

that prefers partitions in $\mathcal{H}_v$ over all possible partitions, and,

$$
\begin{aligned}
P(\boldsymbol{x}|\boldsymbol{y}) &\cong \sum_{h\in\mathcal{H}} P(\boldsymbol{x}^{(2)}|h,\boldsymbol{y})P(h|\boldsymbol{y}) \\
&= \sum_{h\in\mathcal{H}}\sum_{\boldsymbol{x}^{(1)}} P(\boldsymbol{x}^{(2)}|h,\boldsymbol{x}^{(1)})P(h|\boldsymbol{x}^{(1)})P(\boldsymbol{x}^{(1)}|\boldsymbol{y}),
\end{aligned} \tag{6.1}
$$

where both $P(\boldsymbol{x}^{(1)}|\boldsymbol{y})$ and $P(\boldsymbol{x}^{(2)}|h,\boldsymbol{x}^{(1)})$ are modeled as discriminative fields which will be explained in Sections 6.2.2 and 6.2.3. In Eq. (6.1), computing the sum over all the possible configurations of $\boldsymbol{x}^{(1)}$ is a NP-hard problem. One naive way to reduce complexity is to do inference in layer 1 until equilibrium is reached and then using this configuration $\hat{\boldsymbol{x}}^{(1)}$ as input to the next layer, i.e., $P(\boldsymbol{x}^{(1)}|\boldsymbol{y}) = \delta(\boldsymbol{x}^{(1)} - \hat{\boldsymbol{x}}^{(1)})$. However, by doing this, one loses the power of modeling the uncertainty associated with the labels in layer 1, which was included explicitly in Eq. (6.1) through $P(\boldsymbol{x}^{(1)}|\boldsymbol{y})$. In principle, one can use Monte Carlo sampling or a variational approach to approximate the sum in Eq. (6.1), but they may be computationally expensive. In this work, instead, we wanted to examine what could be achieved by making a very simplifying assumption, where along with the equilibrium configuration, we also propagate the uncertainty associated with it to the next layer. We use the site-wise maximum marginal configuration as $\hat{\boldsymbol{x}}^{(1)}$. Let the marginals at each site i be

$b_i(x_i^{(1)}) = \sum_{\boldsymbol{x}^{(1)}\backslash x_i^{(1)}} P(\boldsymbol{x}^{(1)}|\boldsymbol{y})$, and $\boldsymbol{b}(\boldsymbol{x}^{(1)}) = \{b_i(x_i^{(1)})\}_{i\in S^{(1)}}$. The belief set, $\boldsymbol{b}(\boldsymbol{x}^{(1)})$ is propagated as an input to the next layer. Note that the configuration $\hat{\boldsymbol{x}}^{(1)}$ can be obtained directly from $\boldsymbol{b}(\boldsymbol{x}^{(1)})$ by taking its sitewise maximum configuration. Thus, in the future, we will omit explicit conditioning on $\hat{\boldsymbol{x}}^{(1)}$. Now, we can write,

$$P(\boldsymbol{x}|\boldsymbol{y}) \approx \sum_{h\in\mathcal{H}} P(\boldsymbol{x}^{(2)}|h, \boldsymbol{b}(\boldsymbol{x}^{(1)}))P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)})). \tag{6.2}$$

Note that both terms in the summation implicitly include the transition probabilities $P(x_r^{(2)}|\hat{x}_i^{(1)})$. For the first term, these are absorbed in the unary potential of the discriminative field in layer 2 as explained in Section 6.2.3. Section 6.2.4 will describe a simple design choice for $P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)}))$. We first describe the modeling of the discriminative field in layer 1.

6.2.2 Discriminative Field - Layer 1

The conditional distribution of the labels given the observed data, i.e., $P(\boldsymbol{x}^{(1)}|\boldsymbol{y})$ is directly modeled as a multiclass DRF described in Chapter 5, Eq. (5.1). According to this, the unary potential can be written as,

$$A^{(1)}(x_i^{(1)}, \boldsymbol{y}) = \sum_{k\in\mathcal{L}^{(1)}} \delta(x_i^{(1)} = k) \log P'(x_i^{(1)} = k|\boldsymbol{y}), \tag{6.3}$$

where $\delta(x_i^{(1)}\!=\!k)$ is 1 if $x_i^{(1)} = k$ and 0 otherwise, and $P'(x_i^{(1)} = k|\boldsymbol{y})$ is an arbitrary domain-specific discriminative classifier. This form of unary potential gives us the desired flexibility to integrate different applications preferring different types of local classifiers in a single framework. Let $\boldsymbol{h}_i(\boldsymbol{y})$ be a feature vector (possibly in a kernel-projected space), that encodes appearance based features for the i^{th} site (a pixel, a patch or an object). To model $P'(x_i^{(1)} = k|\boldsymbol{y})$, in this work we use a softmax function as in Eq. (5.3) as,

$$P'(x_i^{(1)} = k|\boldsymbol{y}) = \begin{cases} \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{h}_i(\boldsymbol{y}))}{1+\sum_{l=1}^{|\mathcal{L}^{(1)}|-1}\exp(\boldsymbol{w}_l^T \boldsymbol{h}_i(\boldsymbol{y}))} & \text{if } k < |\mathcal{L}^{(1)}| \\ \\ \frac{1}{1+\sum_{l=1}^{|\mathcal{L}^{(1)}|-1}\exp(\boldsymbol{w}_l^T \boldsymbol{h}_i(\boldsymbol{y}))} & \text{if } k = |\mathcal{L}^{(1)}| \end{cases},$$

where, $\boldsymbol{w}_k$ are the model parameters for $k = 1 \ldots |\mathcal{L}^{(1)}| - 1$. For a $|\mathcal{L}^{(1)}|$ class classification problem, one needs only $|\mathcal{L}^{(1)}| - 1$ independent hyperplanes.

The pairwise potential predicts how the labels at any two neighboring sites should interact given the observations. The interaction potential in defined similar to Eq. (5.4) as,

$$I^{(1)}(x_i^{(1)}, x_j^{(1)}, \boldsymbol{y}) = \sum_{k,l \in \mathcal{L}^{(1)}} \boldsymbol{v}_{kl}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}) \delta(x_i^{(1)} = k) \delta(x_j^{(1)} = l), \tag{6.4}$$

where, $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is the pairwise feature vector, and $\boldsymbol{v}_{kl}$ are the model parameters. For example, in the case of object detection, the vector $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ encodes the pairwise features required for modeling geometric and possibly photometric consistency of a pair of parts. On the other hand, in the pixelwise image labeling application, the sitewise label smoothing can be achieved by forcing $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ to be 1.

6.2.3 Discriminative Field - Layer 2

The formulation of the discriminative field for layer 2 can be obtained in the same way as described in the previous section by changing the observations to $\boldsymbol{b}(\boldsymbol{x}^{(1)})$, the set of sites to $S^{(2)}$, and the label set to $\mathcal{L}^{(2)}$. The main difference lies in the form of the unary potential. Each node $r \in S^{(2)}$ in this layer receives beliefs as input from the nodes contained in set $S_r^{(1)}$ from the layer below. Taking into consideration the transition probabilities on the directed links between node r and all the nodes in $S_r^{(1)}$, the unary potential can be written as,

$$A^{(2)}(x_r^{(2)}, \boldsymbol{b}(\boldsymbol{x}^{(1)})) = \sum_{k \in \mathcal{L}^{(2)}} \Big\{ \delta(x_r^{(2)} = k)$$
$$\Big(\log P'(x_r^{(2)} = k | \boldsymbol{b}(\boldsymbol{x}^{(1)})) + \frac{1}{|S_r^{(1)}|} \sum_{i \in S_r^{(1)}} \log P(x_r^{(2)} = k | \hat{x}_i^{(1)}) \Big) \Big\}. \tag{6.5}$$

Here, the first term in parentheses on the right hand side involves local classifier $P'(.)$, which is again modeled as a softmax function. It takes features as input, which are constructed using the beliefs $\boldsymbol{b}(\boldsymbol{x}^{(1)})$ at layer 1. The second term arises due to the directed connections between each node $r \in S^{(2)}$ in layer 2 to all the nodes in the set

$S_r^{(1)}$ in layer 1. The effect of this term can be understood by switching the first term off along with the interaction potential. This will lead to the intuitive reasoning of assigning node r that label which maximizes the joint transition probability (computed by assuming each site in $S_r^{(1)}$ to be independent) given a label configuration, i.e., $\hat{\boldsymbol{x}}^{(1)}$ at layer 1. The term, $|S_r^{(1)}|$ acts as a normalizer that takes into account the different cardinalities of sets $S_r^{(1)}$. In the interaction potential for this layer, the features $\boldsymbol{\mu}_{ij}(.)$ are designed such that they capture relative configurations of two regions or objects.

6.2.4 Modeling Partitioning

The distribution $P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)}))$ should be designed such that it gives high weight to a partition $h \in \mathcal{H}_v$, given the belief set from layer 1. Since a good partition should drive all the nodes in a set $S_r^{(1)}$ to take the same true labels, the conditional distribution over the partitions is modeled as,

$$P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)})) \propto \left\{ \prod_{r \in S^{(2)}} \left[\max_{x_r^{(2)} \in \mathcal{L}^{(2)}} \prod_{i \in S_r^{(1)}} \sum_{x_i^{(1)} \in \mathcal{L}^{(1)}} \left(b_i(x_i^{(1)}) P(x_r^{(2)}|x_i^{(1)}) \right) \right]^{1/|S_r^{(1)}|} \right\}^{1/|S^{(2)}|} . \tag{6.6}$$

The term in the product over i is the probability that the node r in layer 2, connected to site i in layer 1, will take label $x_r^{(2)}$. The maximum operator ensures that the maximum possible value of this probability is used for any $x_r^{(2)}$. Finally, the product of these maximum probabilities for all the sites in layer 2 gives a reasonable estimate of the homogeneity of a given partition. Here, $|S_r^{(1)}|$ and $|S^{(2)}|$ compensate for the differences in the number of nodes in set $S_r^{(1)}$ and the overall number of nodes induced by the partition respectively.

6.3 Parameter Learning and Inference

The set of parameters Θ, to be learned in the hierarchical model, includes the parameters of the discriminative fields at layer 1 and layer 2, and the transition probability matrices $P(x_r^{(2)}|\hat{x}_i^{(1)})$. The field parameters for each layer are the parameters of the unary and pairwise potentials i.e., $\theta^{(\alpha)} = \left\{ \boldsymbol{w}_k^{(\alpha)}, \boldsymbol{v}_{kl}^{(\alpha)} \right\}_{\forall k,l}^{\alpha=1,2}$.

Given M i.i.d. labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood $L(\Theta) = \sum_{m=1}^{M} \log P(\boldsymbol{x}^m|\boldsymbol{y}^m, \Theta)$, where the conditional distribution in the sum for each image m is given by Eq. (6.1). Since this likelihood is hard to evaluate, following the assumption made in Section 6.2.1, we use a sequential learning approach in which, first the parameters of layer 1 are estimated separately. Fixing these estimates, the parameters of the next layer and the transition matrices are estimated by maximizing the likelihood for the conditional distribution given in Eq. (6.2). Although suboptimal, the drawbacks of the sequential approach are somewhat moderated by the fact that the partition functions for the fields in the two layers are decoupled due to the directed connections.

Starting with parameter learning in layer 1, since the labels at this layer are not known, we assign pseudo-labels $\boldsymbol{x}^{(1)}$ on S using the true labels $\boldsymbol{x}$. In the image labeling applications, since the nodes at both the layers take the labels from the same set, one can assume the pseudo-labels to be the same as the true labels. For object detection, where the labels at layer 1 are part identifiers rather than being object identifiers, one possible way to generate the pseudo-labels will be to use soft clustering on the object parts and assign a part label to each node as in [82]. It is clear that the labels generated in this way are going to be noisy. That is where the hierarchical model becomes more relevant, where the top layer refines the label estimates from the layer below and the directed connections incorporate the transition probabilities from the noisy labels to the true labels.

To learn the parameters of the discriminative field in layer 1 using gradient ascent, the derivative of the log-likelihood from the distribution $P(\boldsymbol{x}^{(1)}|\boldsymbol{y}, \theta^{(1)})$ can be written as,

$$\frac{\partial l(\theta^{(1)})}{\partial \boldsymbol{w}_k^{(1)}} = \sum_m \sum_{i \in S^{(1)}} \Big(\delta(x_i^{(1)m} = k) - \Big\langle \delta(x_i^{(1)} = k) \Big\rangle \Big) \boldsymbol{h}_i(\boldsymbol{y}^m), \tag{6.7}$$

$$\begin{aligned} \frac{\partial l(\theta^{(1)})}{\partial \boldsymbol{v}_{kl}^{(1)}} = \sum_m \sum_{i \in S^{(1)}} \sum_{j \in \mathcal{N}_i} \Big(& \delta(x_i^{(1)m} = k)\delta(x_j^{(1)m} = l) \\ & - \Big\langle \delta(x_i^{(1)} = k)\delta(x_j^{(1)} = l) \Big\rangle \Big) \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m), \end{aligned} \tag{6.8}$$

where $\langle . \rangle$ denotes expectation with respect to the distribution $P(\boldsymbol{x}^{(1)}|\boldsymbol{y}^m, \theta^{(1)})$. Generally the expectation in Eq. (6.7) and Eq. (6.8) cannot be computed exactly due to the exponential number of configurations of $\boldsymbol{x}^{(1)}$. In this work, for layer 1, we estimate expectations using the pseudo-marginals returned by loopy Belief Propagation (BP) [45] as done in the PMA approximation described in Chapter 4. However, as will be discussed in Section 6.4.1, we found that for layer 2, the Maximum Marginal Approximation (MMA) yields better performance, where thresholded beliefs are used to estimate the required expectations (Section 4.3.4).

The transition probability matrices were assumed to be the same for all the directed links in the graph to avoid overfitting. The entries in this matrix were estimated using the normalized expected counts of transition from $\hat{x}_i^{(1)}$ to $x_r^{(2)}$, which are known at the training time. Note that the counts are computed using the refined label estimates $\hat{x}_i^{(1)}$ obtained directly from $\boldsymbol{b}(\boldsymbol{x}^{(1)})$.

Given $\boldsymbol{b}(\boldsymbol{x}^{(1)})$ and $P(x_r^{(2)}|\hat{x}_i^{(1)})$, the field parameters of layer 2 i.e., $\theta^{(2)}$ were obtained by maximizing the lower bound on the log likelihood of Eq. (6.2),

$$l'(\theta^{(2)}) \geq \sum_m \sum_h P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)m})) \log P(\boldsymbol{x}^{(2)m}|h, \boldsymbol{b}(\boldsymbol{x}^{(1)m}), \theta^{(2)}) \tag{6.9}$$

The derivatives of the above lower bound also have similar forms as in Eq. (6.7) and Eq. (6.8) except that the gradients are now the expectations with respect to $P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)})$. In addition, the gradient for the unary parameters $w_k^{(2)}$ at a site r will have the features scaled by the product of transition probabilities for all the nodes in $S_r^{(1)}$. To deal with the problem of summing over h, in principle, one can use full MCMC sampling. However, by using a data-driven heuristic described in Section 6.4, samples from high probability regions of $P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)})$ can be obtained using local search. Usually, the resulting partitions will not be restricted to the valid space $\mathcal{H}_v$. In that case, the training label at node r in layer 2 is obtained by using a majority vote of labels at the nodes in $S_r^{(1)}$.

For inference, in this work we used the sum-product version of loopy BP to find the maximum marginal (MPM) estimates of the labels on the image sites. BP was chosen to find marginals following the argument of *learning-inference coupling* presented in Chapter 4, since BP was also used to approximate the expectations while learning the parameters. The desired label estimates for each node i in set S can be obtained as,

$$\hat{x}_i = \arg\max_k \sum_{h,r:i\in S_r^{(1)}} P_r(x_r^{(2)} = k|h, \boldsymbol{b}(\boldsymbol{x}^{(1)}))P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)})), \tag{6.10}$$

where the sum is carried out over all h by picking the site $r : i \in S_r^{(1)}$ for each h, and $P_r(.)$ is the marginal for site r in layer 2 estimated using loopy BP. Note that the final label at each site is obtained by averaging the beliefs at that site for different partitions, weighted by the goodness of each partition.

6.4 Experiments and Discussion

We conducted experiments to test the capability of the proposed hierarchical approach to incorporate three different types of contextual interactions i.e., *region-region*, *object-region* and *object-object*, as described in Section 6.1. Four datasets for two different applications (image labeling and contextual object detection) were used for testing. For the object detection experiments, the aim was to investigate if the performance of the existing classifiers could be improved by feeding their outputs in the hierarchical model.

6.4.1 Region-Region Interactions

The first set of experiments was conducted on the 'Beach' dataset from [85], which contains a collection of consumer photographs. The goal was to assign each image pixel one of the 6 class labels: $\{$*sky*, *water*, *sand*, *skin*, *grass*, *other*$\}$. This dataset is particularly challenging due to wide within-class variance in the appearance of the data due to drastic illumination conditions (see Figure 6.5 or [85] for more images). Another characteristic of this dataset which makes it difficult is that, for most of the images, a significant area belongs to none of the semantic classes (i.e., falls under the *other* category). Traditionally it has been hard to model this category because any pixel in this category can virtually have unconstrained appearance. Finally, since this dataset contains beach images, there is significant mixing of the water and the sand regions in them, making it hard to separate these two classes. The dataset contained 123 images, each of size 124×218 pixels. This set was randomly split into a training set of 48 images and a test set of 75 images.

The layer 1 of the proposed hierarchical model implemented the smoothness of pixel labels as the local context. Hence, the sites in layer 1 were the image pixels and the neighborhood was defined to be the 4-nearest neighbors on a grid. Similar to [85], three HSV color features and two texture features, based on the eigenvalues of the

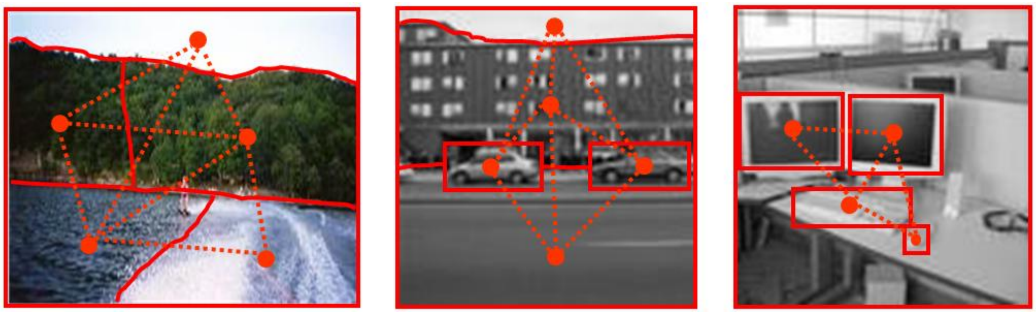


Figure 6.4: An illustration of global interactions of different types in layer 2. Each circle denotes a node corresponding to a region or object. Left: Region-Region interactions. Middle: Object-Region interactions. Right: Object-Object interactions.

second moment matrix, gave a 5 dimensional unary feature vector. Further, we used a quadratic kernel to obtain a 21 dimensional feature vector $\boldsymbol{h}_i$. To implement label smoothing, the pairwise feature vector $\boldsymbol{\mu}_{ij}$ was set to 1, resulting in a Potts model i.e., $\boldsymbol{v}_{kl} = 0$ if $k \neq l$. The parameters of layer 1 i.e., $\theta^{(1)} = \{\boldsymbol{w}_k^{(1)}, \boldsymbol{v}_{kk}^{(1)}\}_{\forall k}$ were all learned simultaneously using the maximum likelihood procedure described Section 6.3. The training time was about 10 min on a 2.8 GHz Pentium class processor.

Before proceeding to layer 2, we describe how we do local sampling of partition h in a high probability region of $P(h|\boldsymbol{b}(\boldsymbol{x}^{(1)}))$. As explained in Section 6.2.4, good partitions are those that promote homogeneous labeling within a region. So, given the beliefs from layer 1, first a binary map is generated for each class by thresholding the pixelwise beliefs at a small value. Then, a partition is obtained by simply intersecting these binary maps for all the classes, i.e., by dividing bigger regions into smaller ones whenever there is an overlap between regions from any two maps. By varying the threshold for generating the binary maps, one can have the desired number of samples. We observed that even less than 5 samples were sufficient to give good results. This was because the beliefs from layer 1 are smoothed due to message passing between the nodes in this layer while implementing the local context.

The layer 2 encodes interactions among different regions given the beliefs at layer 1 and a partition. Each region of the partition is a site in layer 2. An example illustration of the interactions in layer 2 is shown in Figure 6.4, left image. Note that the sites are not placed in a regular grid as in layer 1. For this dataset, the number of sites at layer 2 varied from 13 to 49 for different images. Since we want every region in the scene to influence every other region, each node in the graph was connected to every other node. The computations over these complete graphs are still efficient because of the small number of nodes in the graph. The unary feature vector for each node r consists of normalized product of beliefs from all the sites i in $S_r^{(1)}$ and the normalized centroid location of the region r. This gives an 8 dimensional feature vector. Further, quadratic transforms were used to obtain a 44 dim vector $\boldsymbol{h}_i$. Similar to [131], we use pairwise features between regions to be binary indicator attributes. These were: a region is *above*, *beside* or *enclosed* within another region. The PMA based learning did not perform well on this layer. This may be either due to message passing on a complete graph, or strong attractive or repulsive potentials on the induced graph in layer 2, leading to erroneous estimates of the marginals. On the other hand, we found that MMA performed well on this layer. This is possible since MMA is not affected by the errors in the estimates of the true marginals if the ranking of the marginals remains unaltered (refer to Section 4.5). The maximum likelihood learning with MMA approximation took about 5 minutes.

A few example results from the test set are shown in Figure 6.5. The softmax classifier (second column, Figure 6.5) does not perform well because it classifies each pixel independently without considering interactions in the labels. There are two main problems with the softmax classification. First, several large regions in images are assigned wrong labels e.g., sand regions have been assigned label *water* (rows 1, 2 and 6 from top) or vice versa (bottom row). There is also substantial confusion in water and sky regions (rows 3, 4, 7 from top), and sand and sky regions (row 5 from top). These errors are not surprising if we rely just on the local appearance of the image pixels. Second, the labels are not smooth due to small 'pixelated' label errors giving the resulting classification a dithered appearance. The smoothness of labels can be achieved (third column, Figure 6.5) by implementing smoothing interaction potential in the first layer of the hierarchical model. However, the errors in the larger regions are not eliminated. But, when the full hierarchical model is applied where the second layer enforces the spatial configuration of the regions, most of the errors are eliminated. There are several images that contain pixels which do not belong to any of the semantic classes (e.g., clothing, chairs, houses etc. in top two rows). It is worth noting that good accuracy is obtained even for these pixels, which have traditionally been hard to model because of large within class variations. The pixelwise beliefs for the final output of the hierarchical model are shown in the right most column of Figure 6.5.

Table 6.1 gives a quantitative comparison of the results on the test set. The use

Table 6.1: Pixelwise classification accuracy (%) for scene labeling on two different datasets. Final results of the hierarchical approach are shown in bold. The column 'Others' gives the results for the techniques proposed by other researchers.

Datasets	Softmax	Layer1	Full	MRF	Others
Beach	62.3	63.8	**74.0**	61.5	64.0 [85]
Sowerby	85.4	85.8	**89.3**	81.8	89.5 [56]

Table 6.2: Confusion matrix for the full hierarchical model on the Beach test dataset. Each entry represents the classification rate in percentage of all the pixels in the test set. Rows indicate the true class. The right most column shows the fraction (in %) of all the pixels from the test set corresponding to each class.

	sky	water	sand	skin	grass	other	% of each class in the test set
sky	23.12	2.30	0.68	0.01	0.01	1.33	27.45
water	2.00	28.60	3.38	0.11	0.46	2.43	36.98
sand	0.87	2.10	9.91	0.18	0.02	2.13	15.21
skin	0.00	0.00	0.04	0.34	0.00	0.68	1.06
grass	0.06	0.45	0.16	0.01	3.12	1.07	4.87
other	0.77	2.36	1.69	0.58	0.23	8.80	14.43
Total							100.00

of the local context (label smoothing) improves the accuracy slightly ('Layer 1' in Table 6.1) over the softmax which uses no context. However, the main use of the local context is to propagate improved beliefs and partitions to layer 2. The full hierarchical model ('Full' in Table 6.1) performs significantly better than the others[2]. The confusion matrix showing the detailed performance of the full hierarchical model on different classes is given in Table 6.2. The time taken for inference was about 6 sec for each image. For the MRF, results were obtained using the Potts model.

Next, the hierarchical model was applied to the standard Sowerby dataset. The dataset contained 104 images (64×96 pixels). The training and the test set contained 60 and 44 images respectively. As used by [56], the CIE Lab color features and oriented DoG filters based texture features gave a 30 dim feature vector that was used as input

[2]We implemented the technique proposed in [85] to get the results shown in the table as that work reported only qualitative results.

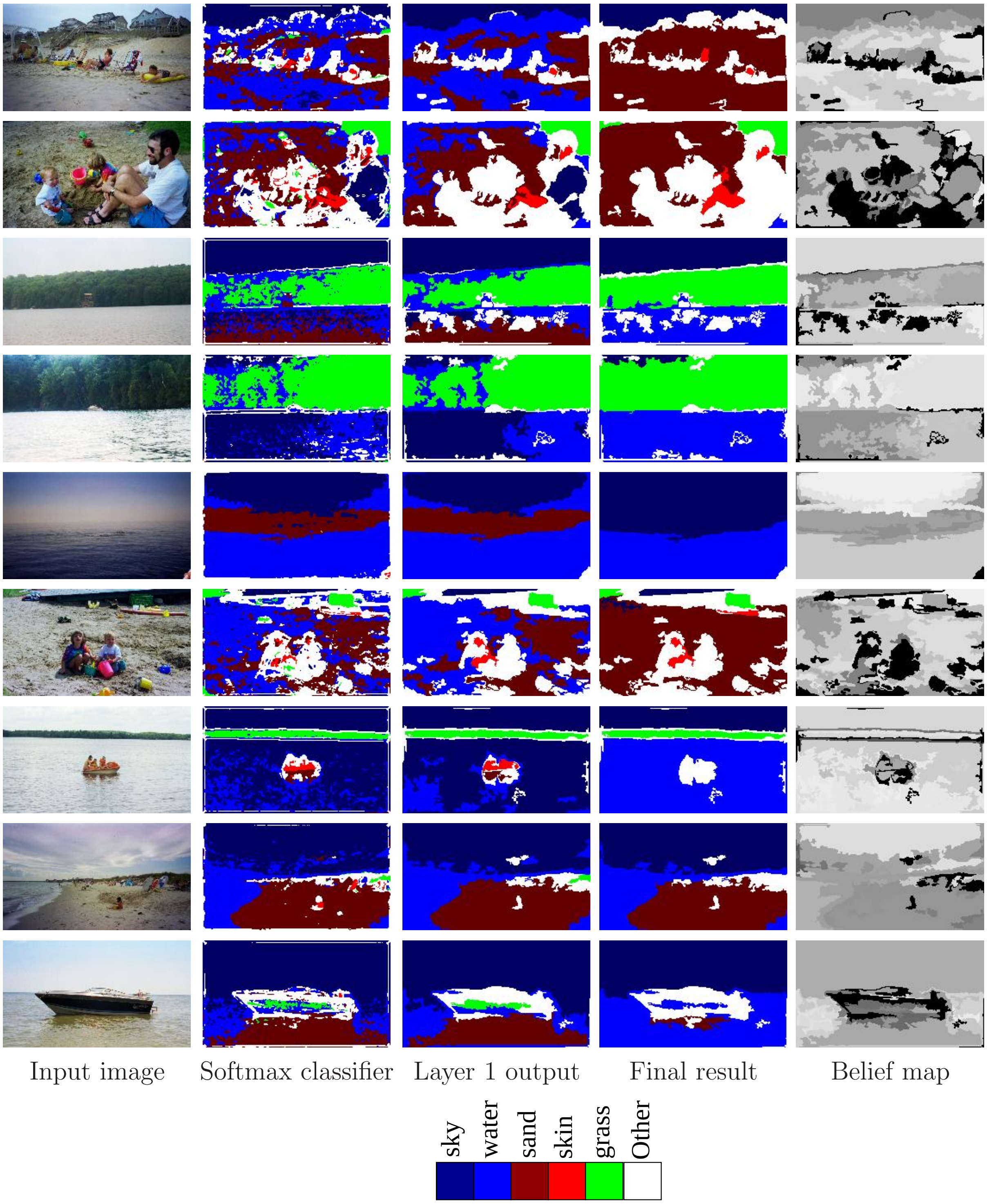


Figure 6.5: Pixelwise classification results on the Beach dataset using context based on *region-region* interactions. 'Layer 1 output' shows the result of implementing label interactions through layer 1 only. Label smoothing is achieved but many large regions are labeled wrong in this output. 'Final result' shows the final classification using both the layers in the hierarchical model which eliminates most of the errors. 'Belief map', shows the pixelwise belief for the final output. Higher intensity indicates higher confidence.

Table 6.3: Confusion matrix for the full hierarchical model on the Sowerby test dataset. Each entry represents the classification rate in percentage of all the pixels in the test set. Rows indicate the true class. The right most column shows the fraction (in %) of all the pixels from the test set corresponding to each class. 'veg' - vegetation, 'rd mark' - road marking, 'rd surf' - road surface, 'bldg' - buildings.

	sky	veg	rd mark	rd surf	bldg	st obj	car	% of each class in the test set
sky	11.93	0.44	0.00	0.22	0.00	0.00	0.00	12.59
veg	0.99	35.72	0.00	1.19	0.57	0.00	0.00	38.47
rd mark	0.01	0.01	0.04	0.14	0.00	0.00	0.00	0.20
rd surf	0.02	0.98	0.01	40.41	0.04	0.00	0.00	41.46
bldg	0.13	4.62	0.00	0.38	0.95	0.00	0.00	6.08
st obj	0.01	0.33	0.00	0.06	0.02	0.02	0.00	0.44
car	0.02	0.42	0.00	0.30	0.01	0.00	0.01	0.76
Total								100.00

to layer 1. The rest of the features, parameter learning and inference were the same as for our implementation on the Beach dataset. Figure 6.6 shows a few typical test results. Note that road markings in images in row 4 and 7 from top are preserved in the final result even though layer 1 tends to smooth it out. The quantitative comparisons are given in Table 6.1. Note that we achieve almost the same accuracy as reported in [56] even though their technique is specifically tuned for the image labeling problems, while our approach is more general, integrating different applications in a single framework.[3]. The confusion matrix showing the detailed performance of the full hierarchical model on different classes is given in Table 6.3.

6.4.2 Object-Region Interactions

We conducted the next set of experiments on a building/road/car dataset from [136].[4] The aim was to detect objects (cars) and regions (building and road) in the images. The dataset contained 31 images, each of size less than 100×100 pixels. The size and pose of the object (car) was roughly the same in all the images. As shown in Figure 6.7, the local appearance of cars is impoverished due to low resolution, making the

[3]Direct comparison with [131] could not be made because their dataset is not available in the public domain

[4]Only a partial dataset was available in the public domain.

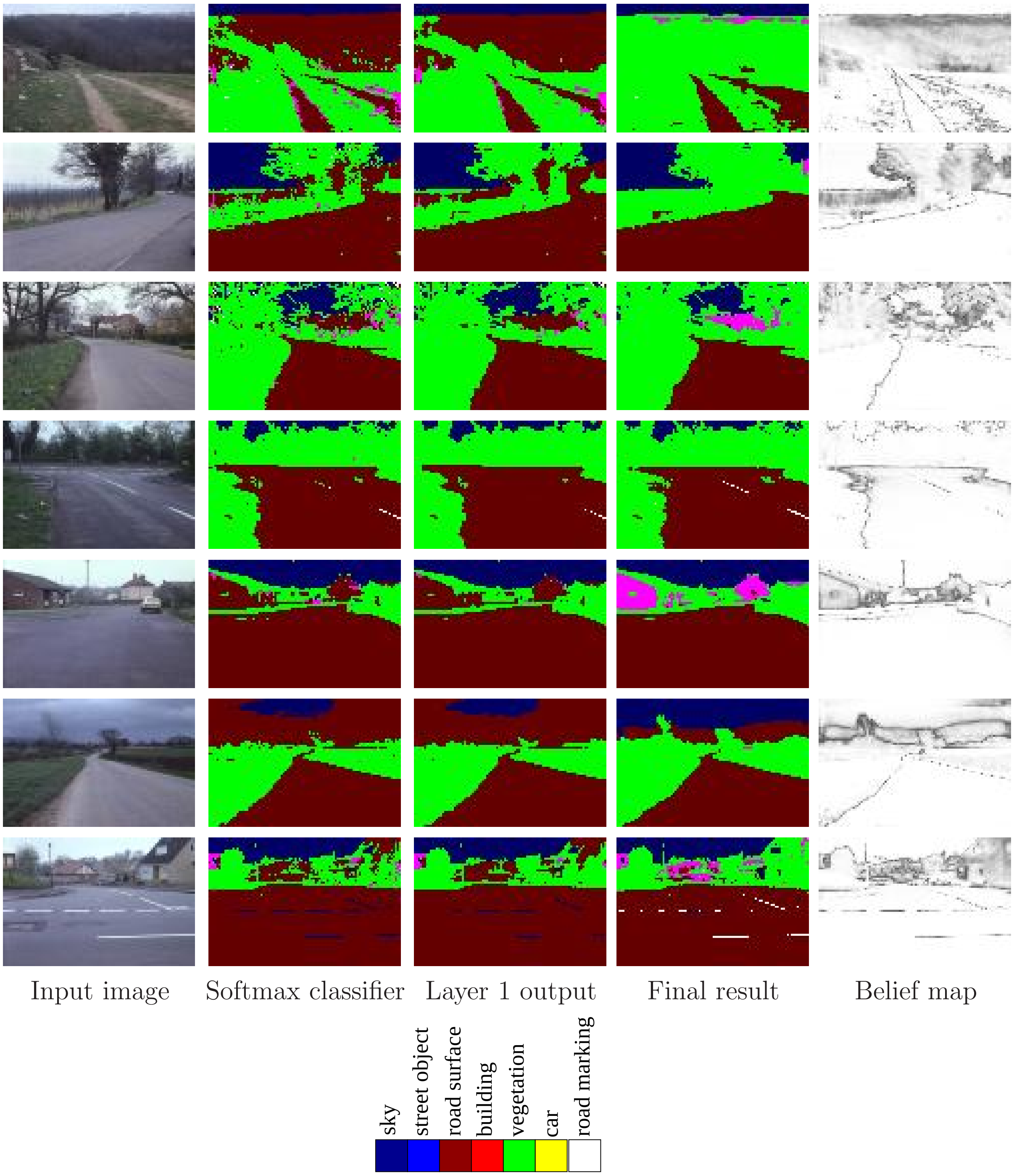


Figure 6.6: Pixelwise classification results on the Sowerby dataset using context based on *region-region* interactions. 'Layer 1 output' shows the result of implementing label interactions through layer 1 only. 'Final result' shows the final classification using both the layers in the hierarchical model. 'Belief map', shows the pixelwise belief for the final output. Higher intensity indicates higher confidence. Note that road markings are preserved in the final result in rows 4 and 7 from top.

car detection hard using stand-alone detectors. In addition, high variability in the appearance of the building data also makes it difficult to disambiguate them from roads just on the basis of intensity and texture features. However, the relationships among the object (car) and the two regions (building and road) provide strong context to improve the detection of all the three entities simultaneously.

For object detection, layer 1 models the relationship among parts of an object. Ideally, in layer 1 one can implement a DRF on object parts similar to the approach explained in Chapter 5. However, to investigate if our framework can be used for improving the performance of a standard boosting-based detector, we use the detector output in layer 1. Rectangular patches centered at the locations that have a score above a threshold are designated as sites for both layer 1 and 2. The threshold is chosen to be small enough to make the false negatives relatively rare. Of course, it will increase the false positives considerably. So, the real question is: can our framework handle a large number of false positives?

In the hierarchical model, the set of sites $S^{(1)}$ in layer 1 contains all the image pixels and the object patches. The neighborhood structure for the pixels was 4 nearest neighbors. Since each object patch represents a possible hypothesis about the full object, there is no interaction among these patches in layer 1. The set of sites in layer 2, $S^{(2)}$, consists of image regions and the same object patches as in layer 1. Note that the sites in $S^{(2)}$ induce a partition on the nodes in $S^{(1)}$. The label sets $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ for the sites in the two layers were the same as $\{$*building*, *road*$\}$ for pixels and regions, and $\{$*car*, *background*$\}$ for the patches. It is interesting to note that, in this application, we have a mixture of two different type of site systems within the same graph, where each system has its own label set. This does not pose any additional computational burden as it can be dealt within the same framework.

The features used by layers 1 and 2 for image pixels and regions were the same as described for the Sowerby dataset in the previous section. The output of the object detector was used as a feature for a patch in layer 2. All the nodes in layer 2 were connected with each other inducing a complete graph. The pairwise features between the object patches and the regions in layer 2 were simply the difference in the coordinates of the centroids of a region and a patch. An example illustration of the interactions in layer 2 is shown in Figure 6.4, middle image.

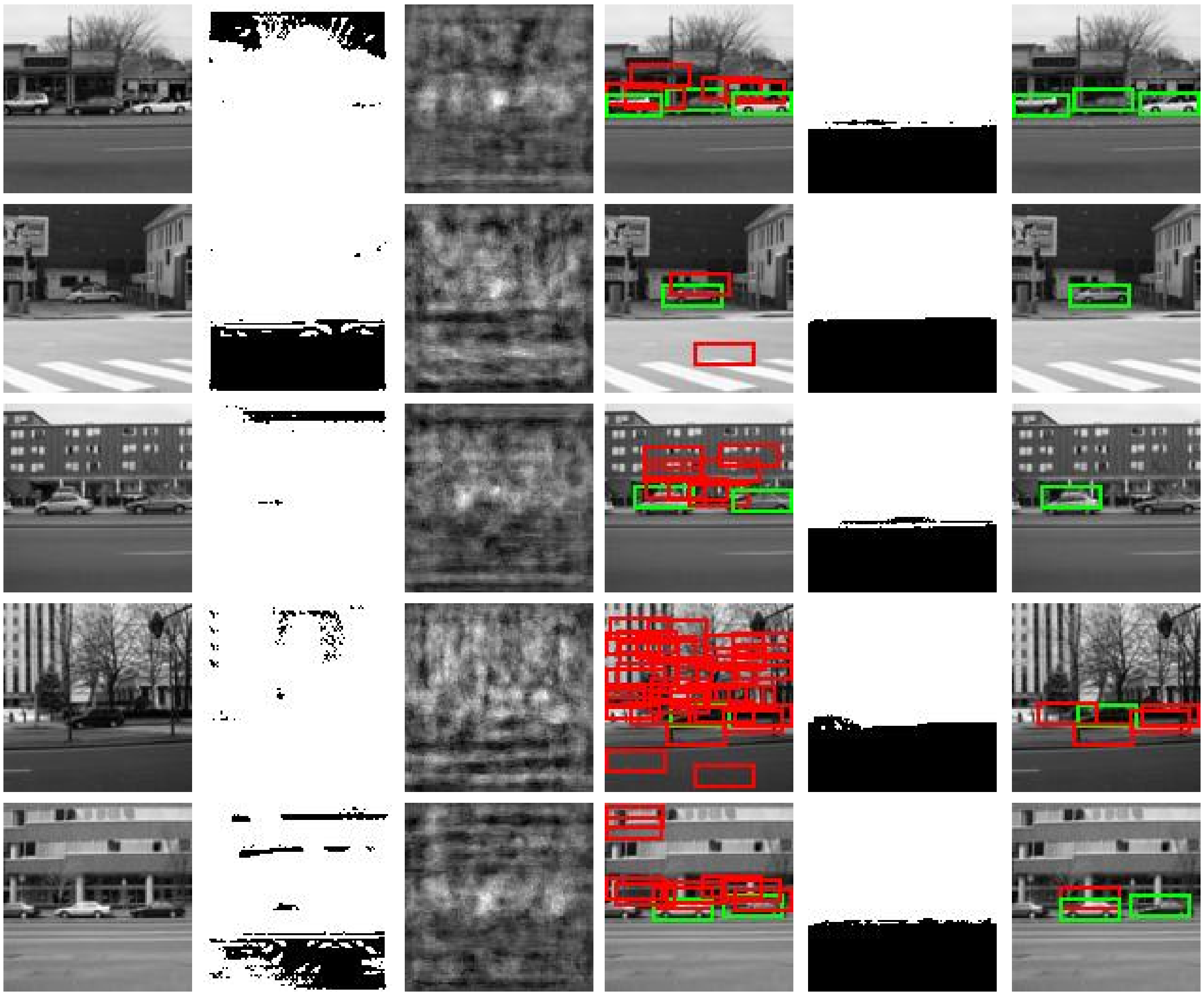
In all the experiments we used a Gentle Boosting based state-of-art detector as a base detector, similar to Torralba et al. [136]. Different versions of boosting algorithms such as Discrete and Real AdaBoost, proposed by Freund and Schapire [44], have been used extensively in computer vision. Gentle boosting is a procedure of fitting an additive logistic regression model by minimizing the exponential loss function [46]. It is similar to Real AdaBoost except that the exact optimization of Real AdaBoost is replaced by the Newton steps, which makes Gentle Boost numerically more stable.

The classification results for a few typical examples from the test set are given in Figure 6.7. The detection of building and road is very error-prone when no context is used in the softmax classification (column 'Bld/road(NC)'). This is because the pixelwise intensity and texture features are usually not sufficient to separate buildings from roads. But when context is used, the model is able to separate both these regions accurately, since buildings tend to occur above roads. Car detection using the boosting-based detector gives many false positives due to poor appearance of the cars. Simultaneous use of context between car, roads and buildings eliminates most of these false positives. The classification accuracy of building and road detection goes up from 70.66% to 98.05% as shown in the confusion matrices for the two regions in Figure 6.8. Also, the ROC curve (Figure 6.8, left) for the car detection shows that the number of false positives is reduced considerably compared to the base detector.

6.4.3 Object-Object Interactions

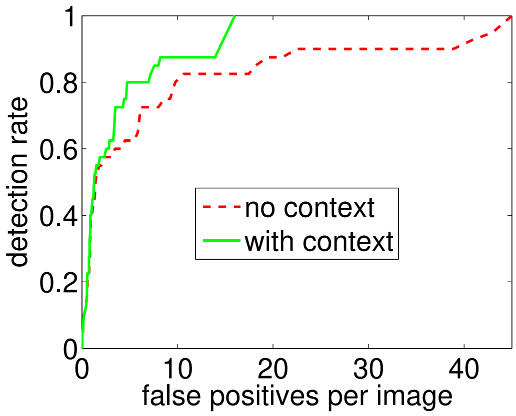
The final set of experiments was conducted on the monitor/keyboard/mouse dataset from [136], which contained 164 images of size less than 100×100 pixels each. The dataset was randomly split in half to generate the training and the test sets. The main challenge in the dataset was the detection of the keyboard and the mouse, which spanned only a few pixels in the images. In this section, we show that by taking interactions among the three objects, one can decrease the false alarms in detection significantly.

For each object, we use a detector which was also trained using gentle boosting as the base detector. Since the size of the mouse in the input images was very small (on average about 8×5 pixels), the boosting based detector could not be trained for the mouse. Instead, we implemented a simple template matching detector by learning a correlation template from the training images. A patch at a location where the output of any of the three detectors is higher than a threshold, represents a site in $S^{(1)}$. The set of sites $S^{(2)}$ in layer 2, was the same as in layer 1 , indicating a trivial



Input image Bld/road(NC) Detector score Car(NC) Bld/road(WC) Car(WC)

Figure 6.7: Detection results for buildings, road and car using context based on *object-region* interactions. 'Bld' - Building, NC - No Context, WC - With Context. Detector score shows the output of a standard boosting-based detector. Black indicates 'road' and white 'buildings'. Green and red indicate true detections and false alarms respectively.



		R	B
No Context	R	14.12	22.86
	B	6.48	56.54
		R	B
With Context	R	36.02	0.96
	B	0.99	62.03

Figure 6.8: Left: The ROC curves for contextual car detection compared to a boosting based detector. Right: Confusion matrices (as % of overall pixels) for building and road detection. Rows contain the ground truth. No context implies the output of the Softmax classifier.

partition. The label set for the sites in $S^{(1)}$ and $S^{(2)}$ was $\{$*monitor*, *keyboard*, *mouse*, *background*$\}$. Since layer 1 uses the output of a standard object detector, interactions among sites take place only at layer 2. An example illustration of the interactions in this layer is shown in Figure 6.4, right image.

The unary features at layer 2 consisted of the score from each detector yielding a 3 dimensional feature vector. The difference of coordinates of the patch centers resulted in a 2 dimensional pairwise feature vector. Each node was connected to every other node in this layer. Figure 6.9 shows a typical result from the test set. It is clear that the false alarms were reduced considerably in comparison to the base detector. The use of context did not change the results for the monitor, since the base detector itself was able to give good performance. This is reasonable because one hopes that context will be more useful when the local appearance of an object is more ambiguous. The ROC curves for the keyboard and the mouse detection are compared with the corresponding base detectors in Figure 6.10. For the mouse detection, even though our approach was able to reduce the false positives significantly, the number of false alarms per image is still high. This is understandable because the size of mouse was very small in all the images. One can hope for context to improve detection only if there is at least 'bare-minimum' appearance based evidence for that object in images.

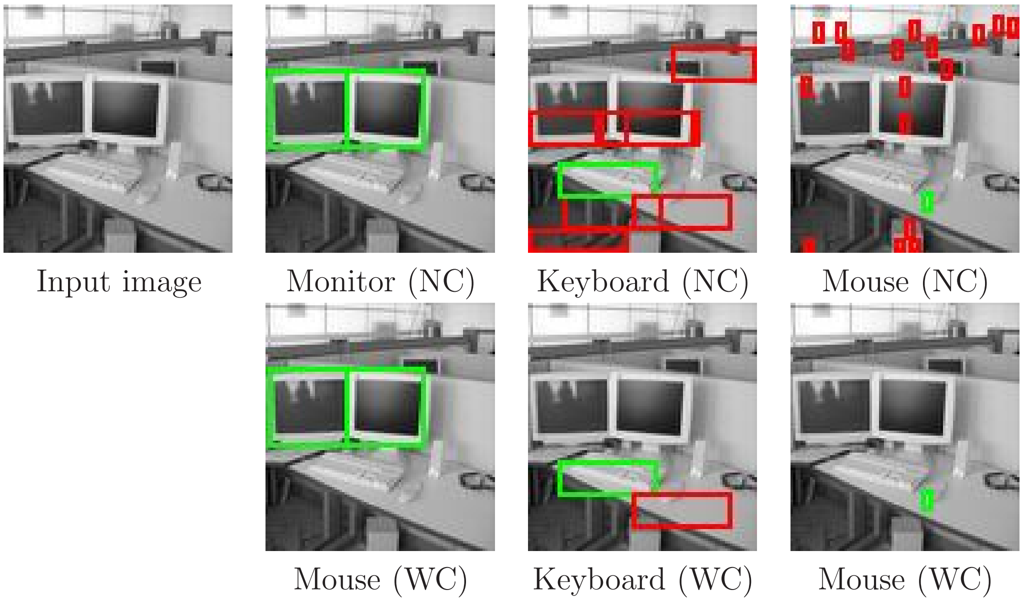


Figure 6.9: Detection results for monitor, keyboard and mouse using context based on *object-object* interactions. NC - No Context, WC - With Context. Monitor detection was good with the base detector itself due to less appearance ambiguity. Note the impoverished appearances of the keyboard and the mouse. Green and red indicate true detections and false alarms respectively.

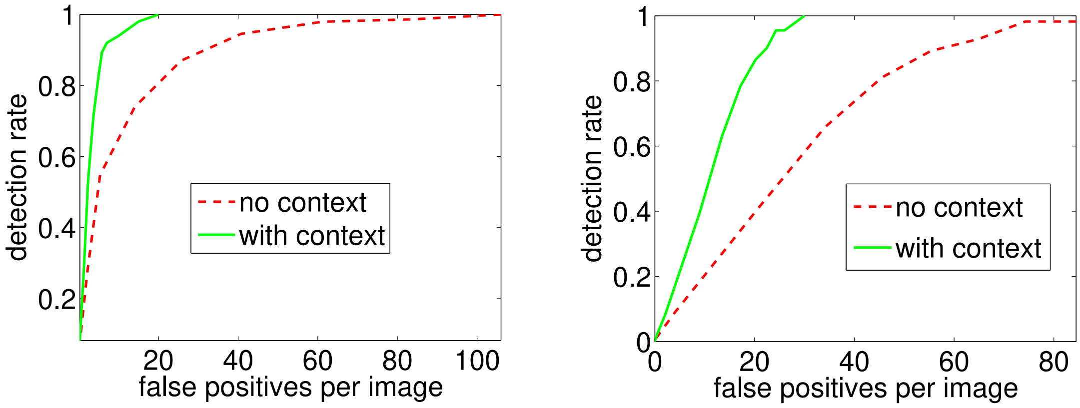


Figure 6.10: The ROC curves for the detection of keyboard (left) and mouse (right). Relatively high false alarm rates for the mouse were due to very small size of mouse (about 8×5 pixels) in the input images.

6.5 Summary

In this chapter, we have presented a two-layer hierarchical formulation to exploit different levels of contextual information in images for robust classification. Each layer is modeled as a discriminative field that allows one to capture arbitrary observation-dependent label interactions. The proposed framework has two main advantages. First, it encodes both the short-range interactions (e.g., pixelwise label smoothing) as well as the long-range interactions (e.g., relative configurations of objects or regions) in a tractable manner. Second, the formulation is general enough to be applied to different domains ranging from pixelwise image labeling to contextual object detection. The parameters of the model are learned using a sequential maximum-likelihood approximation. The benefits of the proposed framework, in spite of a few simplifying assumptions, were demonstrated on four different datasets for image labeling and contextual object detection tasks..

In the future, it will be interesting to explore the use of variational approximations to relax some of the assumptions made in this work, and also to develop efficient ways of learning the parameters of the two layers simultaneously. Finally, it is worthwhile to explore the issue of possible addition of other layers in the hierarchy, which could encode more complex relations between different scenes in a video, leading to event or activity recognition.

Chapter 7

Conclusions and Future Work

7.1 Contributions

In this thesis, we have addressed the problem of incorporating different types of context in computer vision for robust classification of image components including pixels, regions or objects. Towards this, the thesis makes the following key contributions:

- Introduces new probabilistic graphical models in computer vision that allow the use of local discriminative classifiers to incorporate contextual interactions among image components. In particular, this thesis introduces for the first time Conditional Random Field (CRF) [86] based models in computer vision. The use of arbitrary discriminative classifiers for the structured data opens a new channel, alternative to the traditional use of generative classifiers in MRF formulations for images.

- Develops models to capture complex spatial dependencies in labels as well as the observed data simultaneously in a principled manner on 2D lattices with cycles. A principal outcome of such models is the freedom to model interactions among labels using the observed data, which was not possible in the conventional MRFs. This is the main factor that allows the discriminative fields to incorporate different types of image context within the same framework.

- Provides fast and robust parameter learning procedures which are applicable even to the conventional MRF models. In addition, this thesis gives an empirical comparison between different learning and inference techniques indicating coupling of learning and inference mechanisms. This observation is of direct practical use as a guide to which parameter learning procedure one should use given a certain inference technique.

- Proposes a new hierarchical field formulation to model different types of contexts in images simultaneously within the same framework. The context may vary from short-range interactions between pixels to long-range interactions between objects or regions. This is the first work that presents the idea of a hierarchy of CRF-based 2D fields.

- Demonstrates the application of the proposed models on several challenging computer vision tasks such as contextual object detection, semantic scene segmentation, texture recognition and image denoising seamlessly within a single framework.

7.2 Key Observations

In this thesis, we explored various models for incorporating contextual interactions in the classification of image components. Further, we observed the performance of these models on several computer vision tasks. On the basis of the theoretical and the experimental observations made in this thesis, we summarize the key insights as follows.

In addition to the local statistics of a component to be labeled in an image, pooling evidence from all the contextual sources (e.g., neighboring pixels, regions and/or objects) is critical to build an efficient and reliable scene recognition system in the future. This reaffirms the view taken by early scene understanding researchers in computer vision.

Even though the tree-structured causal models provide the advantage of exact parameter learning and inference using very efficient techniques, there are several problems such as non-stationarity, label-bias and restricted modeling power that undermines their capacity to incorporate generic context in natural images.

Among the noncausal models, traditional generative MRF formulations are too restrictive to model the rich interactions in data and labels required for a variety of classification tasks in computer vision. In particular, MRFs do not allow data-dependent label interactions that are critical for many vision applications such as parts-based object detection.

The discriminative fields overcome most of the limitations posed by the traditional MRF frameworks by allowing arbitrary interactions in the observed image data, and data-dependent interactions in the labels. The experimental results conducted on hundreds of real-world as well as synthetic images verify the power of these fields on several applications spanning low-level image denoising to high-level contextual object detection. In addition, the experimental comparisons also indicate that the use of discriminative classifiers is more beneficial than the generative ones in several application domains.

Efficient and robust parameter learning in discriminative fields is possible by exploiting inference to approximate the gradients used in maximum likelihood learning. Further, the learned parameters lead to good classification performance so long as the method used for approximating the gradient is consistent with the inference mechanism.

For robust classification, both the short-range context (e.g., pixelwise label smoothing) as well as the long-range context (e.g., relative configurations of objects or regions) in images must be exploited. This can be achieved by modeling context at different levels through a hierarchy of 'flat' discriminative fields. Even suboptimal learning and inference can give substantial improvements in the classification accuracy.

7.3 Limitations and Future Extensions

There are several limitations of the models presented in this thesis. We divide the future work into two parts to address these limitations as well as to explore further model extensions. The first part, described in this section, discusses some specific issues worth exploring to enhance the power of these models, while the second part (Section 7.4) poses some broad open questions.

Robust parameter learning. By far, the most important challenge in making the discriminative fields applicable to a wide range of classification tasks in computer vision is robust and efficient parameter learning in these fields. We have described several methods of learning parameters in this thesis. However, there are several issues that need further investigation.

One interesting question that emerges from our discussion on using discrete approximations of true expectations in Section 4.3 is: Will any arbitrary choice of a discriminative classifier to design the field potentials give rise to the perceptron-type behavior or it is true only for the potentials that are linear in features? This issue needs further experimental evaluation.

In this thesis, we took the maximum-likelihood view of learning the model parameters. One possible drawback of this view is model overfitting, especially in the presence of limited training data. This may lead to poor generalization performance on the unseen test data. In the future, it is worth experimenting with regularized maximum likelihood, a commonly used procedure in machine learning to alleviate this problem. In this, the growth of the parameters is penalized using a shrinkage prior (e.g., a zero mean Gaussian) over the parameters.

Another important direction to pursue will be to take a full Bayesian view similar to Qi et al.[118], where all the model parameters are integrated while predicting the class labels, instead of computing the point estimate of the parameters as in maximum likelihood learning. Of course, integrating the parameters in discriminative fields is generally a difficult task, requiring several model approximations.

Kernel classification. Kernels have been used extensively in machine learning to yield powerful classifiers. In this work, we showed the use of simple polynomial kernels in designing the clique potentials in the discriminative fields. One of the further enhancements will be to extend the framework to general kernel mappings to increase the classification accuracy. However, since the number of parameters for a general kernel mapping is on the order of the number of data points, one will need some method to induce sparseness to avoid overfitting [135][38]. Recently, several researchers have proposed learning with general kernels on CRF-type of models e.g., max-margin learning by Taskar et al. [134], and a greedy selection approach by Lafferty et al. [87]. Investigation of other techniques based on the extensions of sparse priors [135][38] will be of interest.

Improved learning in the hierarchical framework. In this work, we used simplifying assumptions for parameter learning and inference in the hierarchical framework. In the future, it is essential to explore the use of variational approximations to learn the parameters of the two layers simultaneously. Also, experimentally it will interesting to compare what practical gains are made by employing more computationally complex techniques.

Non-homogeneous and anisotropic fields. In this thesis, we assume the discriminative fields to be homogeneous and isotropic. Homogeneity indicates that the functional form and the parameters of the potentials are not dependent on the image location. On the other hand, isotropy indicates that all the neighbors of a site are treated equally irrespective of the location of the neighbor with respect to the site. Relaxing the conditions of homogeneity and isotropy may be useful for several vision applications. However, the number of parameters will grow rapidly, and large amounts of training data will be required to learn the parameters reliably. Of course, depending on the application, a partial relaxation of these assumptions may be useful.

Arbitrary graph topology. In this thesis, we presented applications such as object detection and semantic segmentation, where the induced graphs had arbitrary topology instead of being fixed as a rectangular grid. For simplicity, we assumed a uniform distribution over all the graph structures. It will be worth examining the possibility of learning a distribution over these structures.

Higher order cliques. One interesting direction would be to enhance the DRF framework to incorporate more than pairwise interactions (i.e., cliques of three nodes or more). This will be useful in dealing with large affine or scale variations in objects leading to the goal of generic object detection. The potentials of these bigger cliques can be modeled such that the potentials remain invariant to scale or affine transformations.

Effects of imbalance in training data. In several cases, discriminative classifiers may generalize badly when the number of training data from different classes is very different [1]. This problem is known as *imbalanced training data* problem. For example, in object detection problems, the number of object patches in training images is much lower than the background patches. The data imbalance problem becomes more serious when the data from different classes has high degree of overlap in the feature space. In our experiments with the DRFs, data imbalance was not found to have any limiting influence. However, examining this issue by using controlled examples will lead to a better understanding of its effects in discriminative fields.

Unsupervised or semisupervised learning. Fully labeled training data is usually more expensive than unlabeled or partially labeled data. As the scope of computer vision expands to handle more complex objects and scenes, it will be increasingly hard to get enough fully labeled training samples. Thus, the development of unsupervised or semisupervised learning methods in these models is important for their wide applicability. Recently, attempts have been made in this direction for the application of object detection [35][120].

Extended experiments on large datasets. Clearly, the testing of object detection formulation on semi-synthetic examples is not satisfying enough. It will be desirable to do extensive testing of these models on large real-world object detection datasets containing object deformations and occlusions, and to compare their performance with the existing techniques.

In the hierarchical formulation, it will be worth exploring the issue of possible addition of other layers in the hierarchy, which could encode more complex relations between different scenes in a video, leading to event or activity recognition.

Evaluation with MRF models. Finally, evaluating the performance of the parameter learning procedures presented in this thesis on conventional MRF models will have great potential. If found suitable, they will provide efficient alternative conditional techniques for learning parameters in MRFs.

7.4 Open Issues

Feature extraction. One valid criticism of the discriminative field models is that they do not eliminate the need for carefully crafted application-dependent features. Although there has been some work in selecting important generic features from a large pool of features [100][117], one still needs to engineer the low-level image features to get good results. To make these random field models applicable to generic real-world tasks, it is unavoidable to incorporate methods for automatically extracting features from the raw image data. In convolutional neural networks this is done by feeding raw pixel data in the bottom most layer of the model [90][89]. An interesting future direction would be to see if similar ideas can be incorporated in the discriminative field models that will reduce the effects of arbitrary preprocessing step of feature extraction.

Uncertainty vs complexity. Turning back to the early work on context based object and scene recognition in '70s and '80s, the main flaw of those techniques was excessive reliance on ad-hoc rules-based reasoning. This makes it hard to compensate for uncertainties and ambiguities inherent to visual data. The models proposed in this thesis mostly overcome that problem but the complexity of these models can grow quickly as evident from the two-layer hierarchical formulation explained in Chapter 6.

As more and more layers are added to extract information at different semantic levels in a scene (e.g., recognition of the whole scene), or in a video (activity recognition over multiple scenes), the complexity of these models is bound to grow rapidly. Thus, in the spectrum of techniques for modeling context, on one end we have techniques that are computationally simple but lack flexibility, and on the other end, techniques that are very flexible but computationally difficult. To build a successful recognition system in the future, the question worths answering is: Is it possible to find a mixed strategy that uses the two ends as required to deal with uncertainty while at the same time being efficient?

Is visual data sufficient? The aim of a generic scene understanding system is to recognize various components of the scene. Whether it can be done purely using the visual data depends on the level of semantics at which we want to parse the scene. For instance, in Figure 7.1, the grass covered wall will be labeled as grass. Actually the correct semantics would be 'grass covered building'. Can it be learned by using just the visual data or some other knowledge source needs to be integrated to produce such deductions?

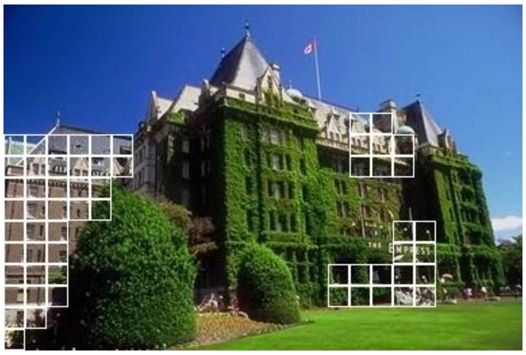


Figure 7.1: An example of building detection in images. The DRF fails on the grass-covered walls etc. Should these areas be labeled as *grass* or *building* or something intermediate?

Appendix A

Man-Made Structure Detection

A.1 Introduction

Automatic detection of man-made structures in ground-level images is useful for scene understanding, robotic navigation, surveillance, image indexing and retrieval etc. In addition, this application provides an ideal testbed to compare various classification techniques because there are significant within class variations in the appearance of data from man-made structures (*structured* class). Similarly, the data from background (*nonstructured* class) is virtually unconstrained, and there is a large overlap between these two classes.

Here we focus on the detection of man-made structures that can be characterized primarily by the presence of linear structures. The detection of such a constrained set of man-made structures from a single static ground-level image is still a non-trivial problem due to three main reasons. First, the realistic views of a structured object captured from a ground-level camera are unconstrained unlike the aerial views, which complicates the use of predefined models or model-specific properties in detection. Second, no motion or stereo information is available, precluding the use of geometrical information pertaining to the structure. Finally, the images of natural scenes contain large amount of clutter, and the edge extraction is very noisy. This makes the computation of the image primitives such as junctions, angles etc., which rely on explicit edge or line detection, prone to errors.

Buildings are one possible instance of man-made structures and some of the related work on structure detection exists for buildings [99][95][76][63][19]. A majority of the techniques for building detection from aerial imagery try to generate a hypothesis on the presence of building roof-tops in the scene [99]. This is usually attained by first detecting low-level image primitives e.g., edges, lines or junctions, and then grouping these primitives using either geometric-model based heuristics [95], or a statistical model e.g., Markov Random Field (MRF) [76]. For the ground-level images, the detection of roof-tops is not feasible and shadows do not constrain the detection problem unlike the aerial images.

Perceptual Organization based building detection has been presented in [63] for image retrieval. In [127] a technique was proposed to learn the parameters of a large perceptual organization using graph spectral partitioning. However, these techniques also require the low-level image primitives to be computed explicitly, and to be relatively noise-free. There has been some recent research work regarding the classification of a whole image as a landscape or an urban scene [111][138]. Oliva and Torralba [111] obtain a low-dimensional holistic representation of the scene using principal components of the power spectra. We found the power spectra based features to be noisy for our images, which contain a mixture of both the landscape and man-made regions within the same image. It might be due to the fact that a 'single' image (or a region contained in it) may not follow the assumption that the power spectra falls with a form $f^{-\alpha}$ where f is spatial frequency [88]. Vailaya et al. [138] use the edge coherence histograms over the whole image for the scene classification, using edge pixels at different orientations. Olmos and Trucco [112] have recently proposed a system to detect the presence of man-made objects in underwater images using properties of the contours in the image. The techniques which classify the whole image in a certain class implicitly assume the image to be exclusively containing either man-made or natural objects, which is not true for many real-world images.

The techniques described in [34][75] perform classification in outdoor images using color and texture features, but employ different classification schemes. These papers report poor performance on the classes containing man-made structures since color and texture features are not very informative for these classes [138]. In addition, in comparison to the Sowerby database used by them, we use a more diverse set of images from the Corel database for training as well as testing.

In this work, we propose to detect man-made structures in a 2D image, located at medium to long distances from the camera. To visualize the problems with low-level primitives using edges, an input image and the corresponding edge image obtained from the Canny edge detector are shown in Figure A.1. It is clear that detection

(a) Input image (b) Edge image

Figure A.1: A natural image and the corresponding edge image obtained using Canny edge detector to illustrate that reliable extraction of low-level image primitives, e.g., lines, edges or junctions for man-made structure detection is hard in natural images.

based on these primitives is going to be a daunting task for this type of images. Instead, in the present work we propose a hybrid approach which uses the bottom-up approach of extracting generic features from the image blocks, followed by the top-down approach of classifying image blocks based on statistical distribution of the features learned from the training data.

A.2 Feature Set Description

The choice of appropriate features without relying on ad-hoc heuristics is important for a generic structure detection system. On the other hand, given a small training set, task dependent feature extraction becomes unavoidable to efficiently encode the relevant task information in a limited number of features. There is currently no formal solution to deriving optimal task-dependent features. In this section, we propose a set of multiscale features that captures the general statistical properties of the man-made structures over spatially adjoining sites.

For each site in the image, we compute the features at multiple scales, which capture intrascale as well as interscale dependencies. The multiscale feature vector at site m is then given as: $f_m = \left[\{f_m^j\}_{j=1}^J, \{f_m^\rho\}_{\rho=1}^R\right]$ where, f_m^j is j^{th} intrascale feature and f_m^ρ is ρ^{th} interscale feature.

A.2.1 Intrascale Features

As mentioned earlier, here we focus on those man-made structures which are primarily characterized by straight lines and edges. To capture these characteristics, at first, the input image is convolved with the derivative of Gaussian filters to yield the gradient magnitude and orientation at each pixel. Then, for an image site m, the gradients contained in a window W_c at scale c ($c=1,\ldots,C$) are combined to yield a histogram over gradient orientations. However, instead of incrementing the counts in the histogram, we weight each count by the gradient magnitude at that pixel as in [7]. It should be noted that the weighted histogram is made using the raw gradient information at *every* pixel in W_c without any thresholding. Let E_δ be the magnitude of the histogram at the δ^{th} bin, and Δ be the total number of bins in the histogram. To alleviate the problem of hard binning of the data, we smoothed the histogram using kernel smoothing. The smoothed histogram is given as,

$$E'_\delta = \frac{\sum_{i=1}^{\Delta} K((\delta - i)/h) E_i}{\sum_{i=1}^{\Delta} K((\delta - i)/h)}, \tag{A.1}$$

where K is a kernel function with bandwidth h. The kernel K is generally chosen to be a non-negative, symmetric function.

If the window W_c contains a smooth patch, the gradients will be very small and the mean magnitude of the histogram over all the bins will also be small. On the other hand, if W_c contains a textured region, the histogram will have approximately uniformly distributed bin magnitudes. Finally, if W_c contains a few straight lines and/or edges embedded in smooth background, as is the case for the *structured* class, a few bins will have significant peaks in the histogram in comparison to the other bins. Let ν_0 be the mean magnitude of the histogram such that

$$\nu_0 = \frac{1}{\Delta} \sum_{\delta=1}^{\Delta} E'_\delta.$$

We aim to capture the average 'spikeness', of the smoothed histogram as an indicator of the 'structuredness' of the patch. For this, we propose heaved central-shift moments for which p^{th} order moment ν_p is given as,

$$\nu_p = \frac{\sum_{\delta=1}^{\Delta}(E'_\delta - \nu_0)^{p+1} H(E'_\delta - \nu_0)}{\sum_{\delta=1}^{\Delta}(E'_\delta - \nu_0) H(E'_\delta - \nu_0)}, \tag{A.2}$$

where $H(x)$ is the unit step function such that $H(x) = 1$ for $x > 0$, and 0, otherwise. The moment computation in Eq. (A.2) considers the contribution only from the bins having magnitude above the mean ν_0. Further, each bin value above the mean is linearly weighted by its distance from the mean so that the peaks far away from the mean contribute more. The moments ν_0 and ν_p at each scale c form the gradient magnitude based intrascale features in the multiscale feature vector.

Since the lines and edges belonging to the *structured* regions generally either exhibit parallelism or combine to yield different junctions, the relation between the peaks of the histograms must contain useful information. The peaks of the histogram are obtained simply by finding the local maxima of the smoothed histogram. Let δ_1 and δ_2 be the ordered orientations corresponding to the two highest peaks such that $E'_{\delta_1} \geq E'_{\delta_2}$. Then, the orientation based intrascale feature β^c for each scale c is computed as $\beta^c = |\sin(\delta_1 - \delta_2)|$. This measure favors the presence of near right-angle junctions. The sinusoidal nonlinearity was preferred to the Gaussian function because sinusoids have much slower fall-off rate from the mean. The sinusoids have been used earlier in the context of perceptual grouping of prespecified image primitives [76]. We used only the first two peaks in the current work but one can compute more such features using the remaining peaks of the histogram. In addition to the relative locations of the peaks, the absolute location of the first peak from each scale was also used to capture the predominance of the vertical features in the images taken from upright cameras.

A.2.2 Interscale features

We used only orientation based features as the interscale features. Let $\{\delta_1^c, \delta_2^c, \ldots, \delta_P^c\}$ be the ordered set of peaks in the histogram at scale c, where the set elements are ordered in the descending order of their corresponding magnitudes. The features between scales i and j, β_p^{ij} were computed by comparing the p^{th} corresponding peaks of their respective histograms i.e., $\beta_p^{ij} = |\cos 2(\delta_p^i - \delta_p^j)|$, where $i, j = 1, \ldots, C$. This measure favors either a continuing edge/line or near right-angle junctions at multiple scales.

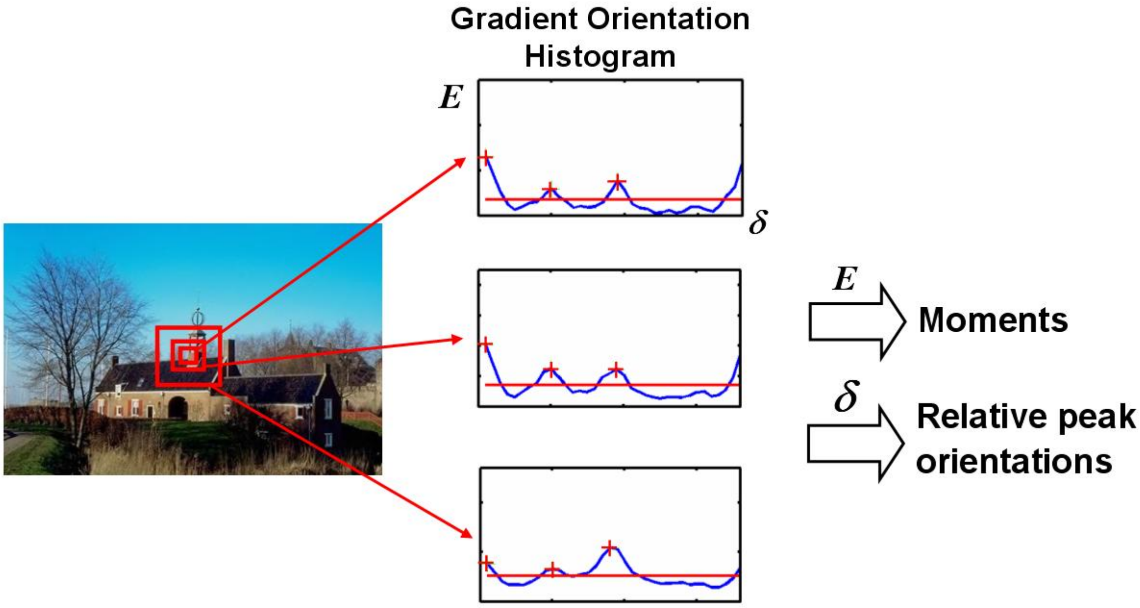


Figure A.2: Multiscale feature extraction at each block in the input image. At each block, image gradients are used to obtain gradient orientation histograms at multiple scales. Moments based features are computed using gradient magnitudes and orientation based features are computed using the peak gradient orientations.

In Figure A.2 we have shown the overall feature extraction procedure. Here at each block we first build orientation histograms of image gradients at multiple scales (three in this case). The vertical axis represents the gradient magnitude (E), and the horizontal axis represents the gradient orientation (δ). The histogram is smoothed using a Gaussian kernel. The solid line in a histogram shows the mean magnitude of the smoothed histogram, and the '+' signs indicate the orientation peaks that are above the mean magnitude. Using the magnitudes we compute the histogram moments while the peaks are used to compute the orientation based intra- as well as inter-scale features.

A.3 Experimental Setup

To test the performance of different models proposed in this thesis, we trained and tested these models on two different datasets drawn randomly from the Corel Photo Stock. The training set consisted of 108 images while the testing set contained 129 images, each of size 256×384 pixels. Most of the images in both the datasets contained both natural objects and man-made structures captured at medium to long distances from a ground-level camera. A subset of the training images is shown in Figure A.3. It is clear from the figure that this is a very challenging dataset. There are significant variations in the scale of the structures varying from very small to large ones. Also, the illumination in images is poor because of shadows and weather conditions. The pose of the structures was not restricted and there are significant perspective distortions. The dataset had several structures which had curved contours. So, simple line detection is not enough for them. The training set also contained a few negative examples such as images with horizon or tree trunks, which could give rise to features similar to that of man-made structures.

The ground truth was generated by hand-labeling each nonoverlapping 16×16 pixels block in each image as a *structured* or *nonstructured* block. This kind of coarse labeling was sufficient for our purpose as we were interested in finding the location of the *structured* blocks without explicitly delineating the object boundary. However, the block quantization introduces noise in the labels of the blocks lying on the object boundary, since a block containing a small part of the structure could be given either of the labels. This makes the quantitative evaluation of the results hard and there is no formal solution to this problem. To circumvent this, we do not count as false positive a misclassification that is adjacent to a block with ground truth label *structured*. In practice, small classification variations at the object boundary do not affect future processing such as grouping blocks into connected regions or extracting bounding boxes. The whole training set contained $36,269$ blocks from the *nonstructured* class, and $3,004$ blocks from the *structured* class.

To train different models, a multiscale feature vector was computed for each nonoverlapping 16×16 pixels block in the training images. The details of the feature vector were given in Section A.2. One of the reasons for choosing this block size is related to the fundamental ambiguity in the structure detection task. If the *structure* is too far, it will become like 'texture', and if it is too near, only a small portion (e.g., a long edge or a smooth patch from a wall) will occupy almost the whole image. The lowest and the highest scales for the feature extraction were chosen to constrain this ambiguity. We are interested in the *structures* which are not smaller than the lowest scale, and are not totally smooth or contain only unidirectional edges at the highest scale. For multiscale feature computation, the number of scales was chosen to be 3, with the scales changing in regular octaves. The lowest scale was fixed at 16×16

Figure A.3: Some example images from the training set for the task of man-made structure detection in natural scenes. This task is difficult as there are significant variations in the scale of the objects (row 1), illumination conditions (row 2), perspective distortions (row 3), and non-linear structures (row 4). Row 5 shows some of the negative samples that were also used in the training set.

pixels, and the highest scale at 64×64 pixels. The largest scale implicitly defines the neighborhood ω_m defined in Eq. (2.1) over which the data dependencies are captured.

For each image block, a Gaussian smoothing kernel was used to smooth the weighted orientation histogram at each scale. The bandwidth of the kernel was chosen to be 0.7 to restrict the smoothing to two neighboring bins on each side. The moment features for orders $p \geq 1$ were found to be correlated at all the scales. Thus, we chose only two moment features, ν_0 and ν_2 at each scale. This yielded twelve intrascale features from the three scales including two orientation based features for each scale. For the interscale features, we used only the highest peaks of the histograms at each scale, yielding two features. Hence, for each image block m, a fourteen component multiscale feature vector f_m was obtained. We used only a limited number of features due to the lack of sufficient training data to reliably estimate the model parameters. Each feature was normalized linearly over the training set between zero and one for numerical reasons.

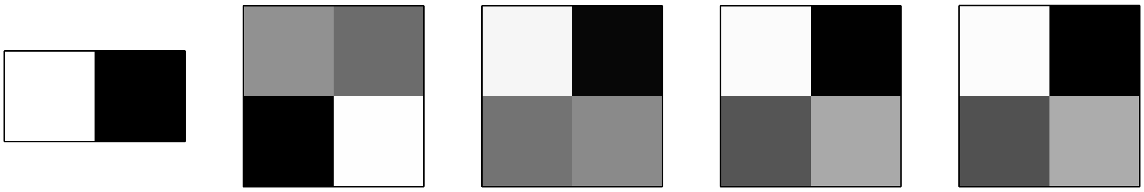
Appendix B

Performance of The Causal Models

Here we discuss the qualitative as well as the quantitative performance of the causal Multi-Scale Random Field (MSRF) model described in Chapter 2. We applied the model to the problem of detecting man-made structures in natural outdoor scenes. The proposed detection scheme was trained and tested on two different datasets drawn randomly from the Corel Photo Stock. The training set consisted of 108 images while the testing set contained 129 images, each of size 256×384 pixels. Further details of the dataset and the features are given in Appendix A.

To learn the parameters of the MSRF model (Θ_p), a quad-tree was constructed considering each 16×16 pixels nonoverlapping block in the image to be a node at the leaf level N. This arrangement resulted in 16×24 nodes at the leaf level and five levels ($N = 5$) in the tree. To take into account the $2:3$ aspect ratio of the images, we modified the quad-tree as suggested in [34] such that the root node had six children. Since we had assumed the conditional transition probability to be the same for each link within a level, we needed to estimate four transition probability matrices, θ_{nkl}, and the prior probability distribution over the root node. For the ML learning described in Section 2.3, the parameter values were initialized by building the empirical trees over the image labels in the training images using the max-voting over the nodes. The training took 8 iterations to converge in 773 s in Matlab 6.5 on a 1.5 GHz Pentium class machine.

The learned parameters are shown in Figure B.1. The brighter intensity indicates a higher probability. It can be noted that for finer levels, the diagonal probabilities are dominant indicating high probabilities of transition to the same class. The transition matrix between level 1 and level 2 shows a more random transition due to the mixing



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

Figure B.1: The learned parameters for the 2-class, 5-level MSRF model. The brighter intensity indicates a higher probability. (a) Prior probabilities at the root node (right block indicates the *structured* class), (b) through (e) transition probability matrices for the links between adjacent levels starting from level 1 to level 5 (top left block indicates the transition from *structured* to *structured* class).

of blocks at coarser levels. Finally, the prior probability distribution at the root node highly favors the root node to be from the *nonstructured* class. This is reasonable since most of the images have much lesser *structured* blocks compared to the *nonstructured* blocks. For the GMM based observation model, the number of Gaussians in the mixture model was selected to be 8 using cross-validation. The mean vectors, full covariance matrices and the mixing parameters were learned using the standard EM technique.

B.1 Performance Evaluation

In this section we present a qualitative as well as a quantitative evaluation of the detection scheme using the MSRF model. First we compare the detection results on the test images using two different methods: only GMM (i.e., no prior model over the labels) with maximum likelihood inference, and GMM in addition to MSRF prior with MPM inference. For convenience, the former will be referred to as the GMM and the latter as the MSRF model in the rest of the paper. The same set of learned parameters was used in GMM for both the methods. For an input test image, the *structure* detection results from the two methods are given in Figure B.2. The blocks identified as *structured* have been shown enclosed within an artificial boundary. It can be noted that for the same detection rate, the number of false positives have significantly reduced for the MSRF based detection. The MSRF model tends to smooth the labels in the image and removes most of the isolated false positives. The bottom image in Figure B.2 shows the MSRF posterior map over the input image for the *structured* class, displaying the posterior marginals for each image block. The

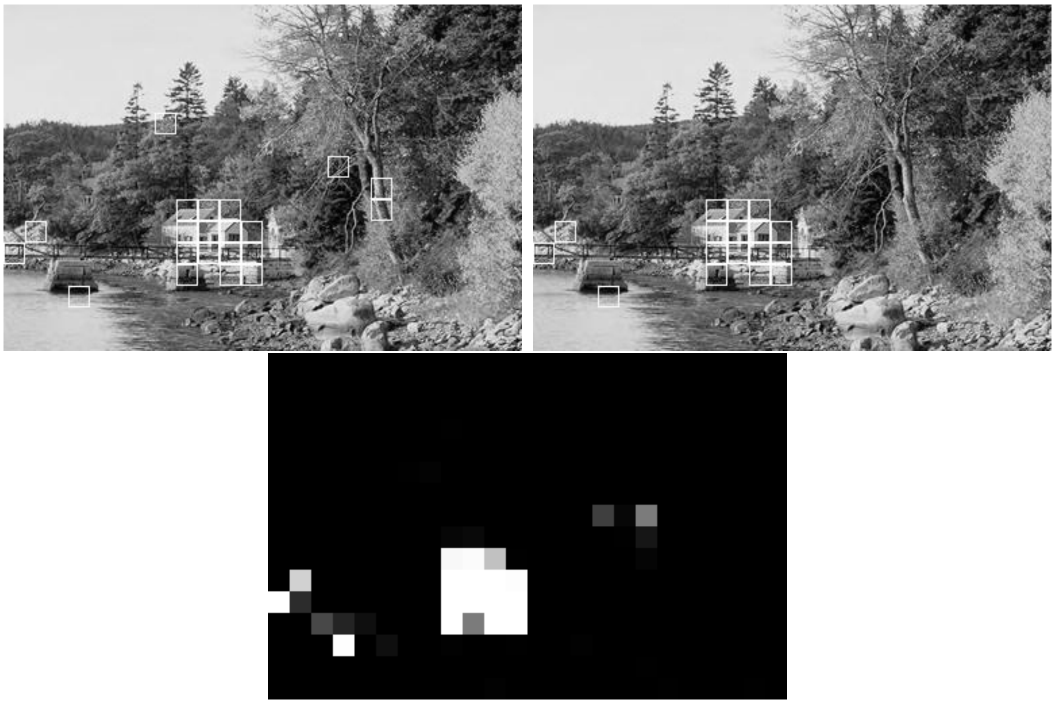
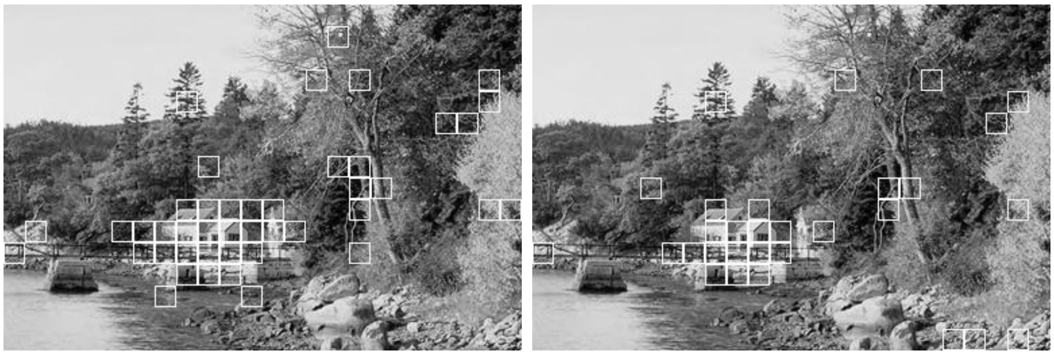


Figure B.2: The structure detection results for the input image given in Figure A.1 (a). Top left: Maximum likelihood results using only GMM. Top right: MPM results using MSRF model. Bottom: The MSRF posterior map displaying the posterior marginals over the image blocks for the *structured* class. The brighter intensity indicates a higher probability.

posterior map exhibits high probability for the *structured* blocks, and the number of *nonstructured* blocks with significant probability is very low. This shows that the MSRF based technique is making fairly confident predictions.

We compare the above results with the results from two other popular classification techniques: Support Vector Machine (SVM) and Sparse Classifier (SC). A Bayesian learning of sparse classifiers was proposed recently by Figueiredo and Jain [38], who have shown good results on the standard machine learning databases. Both classifiers used the multiscale feature vectors defined earlier as the data associated with the image blocks. We implemented a kernel classifier using a symmetric Gaussian kernel of bandwidth 0.1 for both SVM and SC. The cost parameter for SVM was set to be 1000 from cross-validation. The number of support vectors in SVM were found to be 2305, while the number of sparse relevance vectors in SC were 66. The detection results for these two techniques are shown in Figure B.3. The results from SC were based on the MAP inference. It can be seen that the detection rate in the image is



(a) (b)

Figure B.3: The structure detection results using (a) SC, (b) SVM. Both techniques have higher number of false positives in comparison to the MSRF result for a similar detection rate.

	NS	S
NS	42976	188
S	1776	4596

(a) MSRF

	NS	S
NS	42791	373
S	1776	4596

(b) GMM

	NS	S
NS	42587	577
S	1777	4595

(c) SC

	NS	S
NS	42534	630
S	2004	4368

(d) SVM

Figure B.4: Confusion matrices for different techniques. S - *structured*, and NS - *nonstructured*. The detection rate was kept nearly the same for all the techniques. The rows contain the ground truth while the columns contain the detection results.

fairly good for both the techniques. This demonstrates that the multiscale features capture relevant data dependencies for the structure detection. However, the number of false positives for both techniques is significantly higher than that from the MSRF model. Similar to GMM, SVM and SC do not enforce the smoothness in the labels, which led to increased false positives. The average time taken in processing an image of size 256×384 pixels in Matlab 6.5 on a 1.5 GHz Pentium class machine was 2.8 s for MSRF, 2.3 s for GMM, 2.3 s for SC, and 2.8 s for SVM.

To carry out the quantitative evaluation of our work, we first computed the block wise classification accuracy over all the test images. We obtained 94.6% classification accuracy for the 49,536 blocks contained in 129 test images. However, the classification accuracy is not a very informative criterion here as the number of *nonstructured*

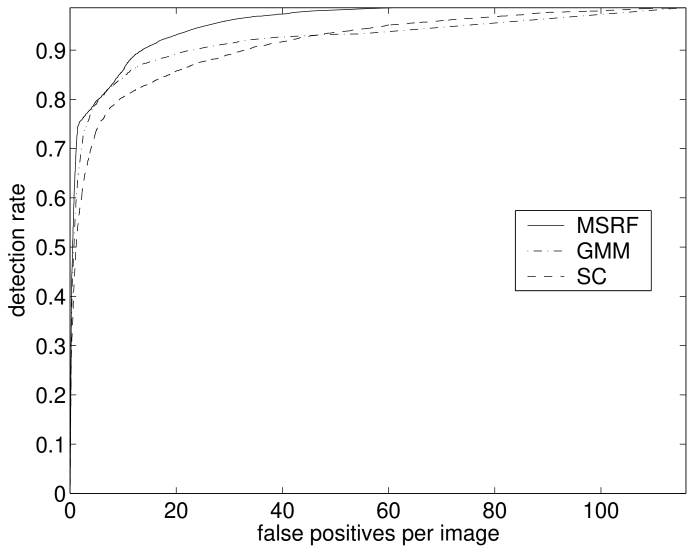


Figure B.5: ROC curves for MSRF, GMM, and SC techniques

blocks $(43, 164)$ is much higher than the number of *structured* blocks $(6, 372)$, and a high classification accuracy can be obtained even by classifying every block to the *nonstructured* class. Hence, we computed two-class confusion matrices for each technique. The confusion matrix for the MSRF model is given in Figure B.4 (a). For an overall detection rate of 72.13%, the false positive rate was 0.43% or 1.46 false positives per image. The main reason for a relatively low detection rate is that the algorithm fails to detect the *structured* blocks that are part of the smooth roofs or walls that have no significant gradients even at larger scales. In fact, it is almost impossible to differentiate these blocks from the smooth blocks contained in natural regions (e.g., sky, land) using any technique without exploiting other auxiliary information such as color. Similarly, too small structures and bad illumination contrast in natural images also make the detection hard. However, it should be noted that this is a significant detection rate at the block level given a low false positive rate. In general we do not require all the blocks of an structured object to be detected since one could use other postprocessing techniques such as color based region-growing to detect the missing blocks of an object.

Keeping the same detection rate as from the MSRF model, we obtain confusion matrices for the GMM and SC. Since SVM does not output probabilities, we varied the cost parameter to obtain the closest possible detection rate. The confusion matrices are given in Figure B.4. The average false positives per image for the GMM, SC and SVM are 2.89, 4.47, and 4.88 respectively. The best among these three gives almost twice false positives per image in comparison to the MSRF model. The results from SVM and SC are quite similar with SC having a slight advantage, since the SVM detection rate is 68.55% in comparison to 72.13% of SC for comparable false positives. For a more complete comparison of the detection performance of the MSRF, GMM, and SC techniques, the corresponding ROC curves are shown in Figure B.5. The MSRF model performs better than the other two techniques. The GMM performs better than the SC most of the times for our test set. For the regions of low false positive per image (less than 2), the performance of MSRF model is significantly better than the other two techniques.

Appendix C

Optical Illusion

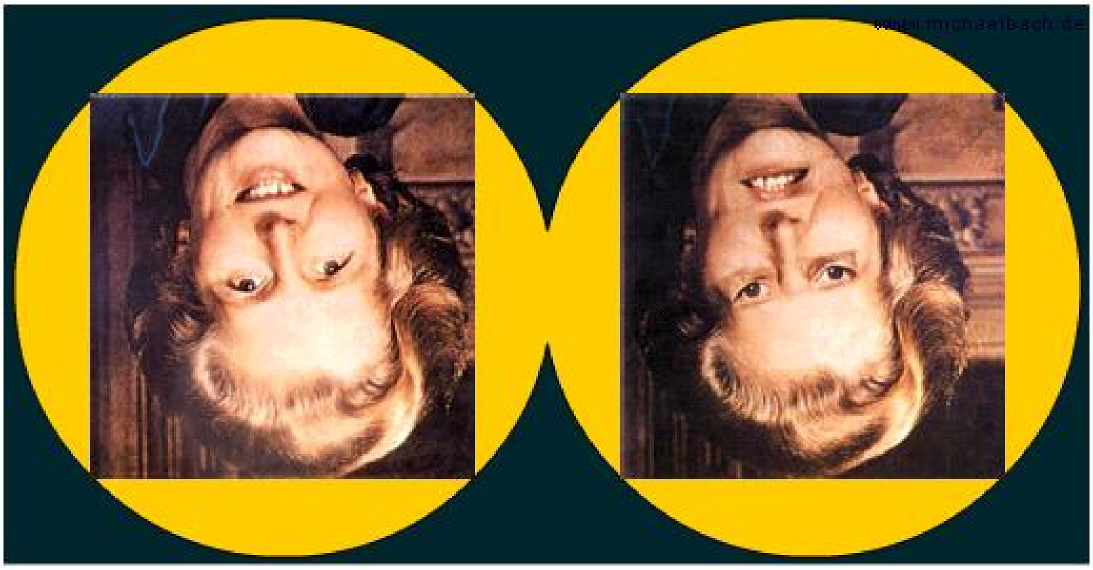


Figure C.1: Are there any differences in the two images shown above? See the next page for more.

Figure C.2: Correct orientation is important even for human visual understanding! This example is from Bach [6].

Bibliography

[1] *ICML'03 Workshop on Learning from Imbalanced Data Sets*. 2003.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[3] Y. Altun, T. Hofmann, and M. Johnson. Discriminative learning for label sequences via boosting. *Advances in Neural Information Processing Systems (NIPS 15)*, 2003.

[4] Y. Altun, T. Hofmann, and A. Smola. Gaussian process classification for segmenting and annotating sequences. *21th International Conference on Machine Learning (ICML)*, 2004.

[5] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden markov support vector machines. *In Proc. 20th International Conference on Machine Learning (ICML)*, 2003.

[6] M. Bach. Optical illusions and visual phenomena. *http://www.michaelbach.de/ot/fcsthompson-thatcher/index.html*.

[7] W. A. Barrett and K. D. Petersen. Houghing the hough: Peak collection for detection of corners, junctions and line intersections. *In Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition*, 2:302–309, 2001.

[8] J. Batlle, A. Casals, J. Freixenet, and J. Marti. A review on strategies for recognizing natural objects in color images of outdoor scenes. *Image and Vision Computing*, 18:515–530, 2000.

[9] A. L. Berger. The improved iterative scaling algorithm: A gentle introduction. 1997.

[10] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Society B*, 36:192–236, 1974.

[11] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Soc.*, B-48:259–302, 1986.

[12] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford Press, Oxford, 1995.

[13] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. *In Proc. European Conf. on Computer Vision (ECCV)*, 2004.

[14] J. S. D. Bonet and P. Viola. A non-parametric multi-scale statistical model for natural images. *In Advances in Neural Information Processing*, 10, 1997.

[15] L. Bottou. *Une Approache theorique de l'Apprentissage Connexionniste Applications a la Reconnaissance de la Parole*. PhD thesis, University de Paris, France, 1991.

[16] C. A. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Trans. on Image Processing*, 3(2):162–177, 1994.

[17] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *In Proc. International Conference on Computer Vision (ICCV)*, I:105–112, 2001.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001.

[19] B. Bradshow, B. Scholkopf, and J. C. Platt. *Kernel Methods for Extracting Local Image Semantics*. Tech. Report MSR-TR-2001-99, Microsoft Research, 2001.

[20] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. *In Proc. European Conf. Computer Vision*, 2004.

[21] H. Cheng and C. A. Bouman. Multiscale bayesian segmentation using a trainable context model. *IEEE Trans. on Image Processing*, 10(4):511–525, 2001.

[22] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(8):749–764, 1995.

[23] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *In Proc. EMNLP*, 2002.

[24] J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. *In Proc. European Conf on Computer Vision (ECCV 02)*, 2002.

[25] D. Crevier and R. Lepage. Knowledge-based image understanding systems: a survey. *Comput. Vis. Image Underst.*, 67(2):160–185, 1997.

[26] G. C. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans Pattern Anal. Machine Intell.*, 5:25–39, 1983.

[27] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480, 1972.

[28] C. D'Elia, G. Poggi, and G. Scarpa. A tree-structured markov random field model for bayesian image segmentation. *IEEE Trans. on Image Processing*, 12(10):1259–1273, October 2003.

[29] A. D. Dempseter, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Jour. Royal Statistical Soc. Ser B*, 39:1–37, 1977.

[30] T. G. Dietterich, A. Ashenfelter, and Y. Bulatov. Training conditional random fields via gradient tree boosting. *In Proc. Int. Conf. Machine Learning*, 2004.

[31] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001.

[32] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. *In Proc. IEEE International Conference on Computer Vision (ICCV 03)*, 2:1134–1141, 2003.

[33] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 00).*, 2000.

[34] X. Feng, C. K. I. Williams, and S. N. Felderhof. Combining belief networks and neural networks for scene segmentation. *IEEE Trans. Pattern Anal. Machine Intelligence*, 24(4):467–483, 2002.

[35] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *In Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 03)*, 2:264–271, 2003.

[36] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *In Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[37] M. A. T. Figueiredo. Adaptive sparseness using jeffreys prior. *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[38] M. A. T. Figueiredo and A. K. Jain. Bayesian learning of sparse classifiers. *In Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition*, 1:35–41, 2001.

[39] M. Fink and P. Perona. Mutual boosting for contextual inference. *Neural Information Processing Systems*, 2004.

[40] M. A. Fischler. The representation and matching of pictorial structures. *IEEE Trans. Computers*, C-22(4):67–92, 1973.

[41] L. R. Ford and D. R. Fulkerson. *Flow in Networks.* Princeton University Press, Princeton, 1962.

[42] C. Fox and G. Nicholls. Exact map states and expectations from perfect sampling: Greig, porteous and seheult revisited. *In Proc. Twentieth Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Sci. and Eng.*, 2000.

[43] Y. Freund and R. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

[44] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. *In Proc. Int. Conf. on Machine Learning*, 1996.

[45] B. Frey and D. J. C. Mackay. A revolution: Belief proapagation in graphs with cycles. *In Proc. Advances in Neural Information Processing Systems*, 10, 1997.

[46] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998.

[47] T. D. Garvey. *Perceptual Strategies for Purposive Vision*. Ph.D. Dissertation, Department of Electrical Engineering, Stanford, California, 1975.

[48] D. Geiger and F. Girosi. Parallel and deterministic algorithms from mrf's: Surface reconstruction. *IEEE Trans PAMI*, 5(5):401–412, May 1991.

[49] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Trans. on Patt. Anal. Mach. Intelli.*, 6:721–741, 1984.

[50] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, San Diego, 1981.

[51] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of Royal Statis. Soc.*, 51(2):271–279, 1989.

[52] C. E. Guo, S. Zhu, and Y. N. Wu. Modeling visual patterns by integrating descriptive and generative models. *International Journal of Computer Vision*, 53(1):5–29, 2003.

[53] J. M. Hammersley and P. Clifford. Markov field on finite graph and lattices. *Unpublished*.

[54] A. R. Hanson and E. M. Riseman. Visions: A computer vision system for interpreting scenes. *Computer Vision Systems*, 1978.

[55] R. L. Harr. Sketching, estimating object positions from relational descriptions. *Computer Graphics and Image Processing*, 19:227–247, 1982.

[56] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labelling. *IEEE Int. Conf. CVPR*, 2004.

[57] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. *Advances in Neural Information Processing Systems 14*, 2:1239–1245, 2002.

[58] G. E. Hinton. Training product of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

[59] G. E. Hinton, S. Osindero, and K. Bao. Learning causally linked markov random fields. *AI & Statistics*, 2005.

[60] G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. pages 282–317, 1986.

[61] C. hsin Wu and P. C. Doerschuk. Tree approximations to markov random fields. *IEEE Trans. Patt. Anal. Machine Intell.*, 17(4):391–402, April 1995.

[62] K. Ikeuchi and T. Kanade. Automatic generation of object recognition programs. *Proc. IEEE*, 76(8):1016–1035, 1988.

[63] Q. Iqbal and J. K. Aggarwal. Applying perceptual grouping to content-based image retrieval: Building images. *In Proc. IEEE Int. Conf. on CVPR*, 1:42–48, 1999.

[64] E. Ising. Beitrag zur theorie der ferromagnetismus. *Zeitschrift Fur Physik*, 31:253–258, 1925.

[65] C. Itzykson and J. M. Drouffle. *Statistical Field Theory*. Cambridge University Press, Cambridge, 1989.

[66] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, November 2001.

[67] T. Kanade. Survey-region segmentation: Signal vs semantics. *Computer Graphics and Image Processing*, 13:279–297, 1980.

[68] Z. Kato, M. Berthod, and J. Zerubia. A hierarchical markov random field model and multi-temperature annealing for parallel image classification. *CVGIP*, 4(9):18–37, 1996.

[69] J. Kittler. Probabilistic relaxation: Potential, relationships and open problems. *In Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 393–408, 1997.

[70] J. Kittler and E. R. Hancock. Combining evidence in probabilistic relaxation. *Int. Jour. Pattern Recog. Artificial Intelli.*, 3(1):29–51, 1989.

[71] J. Kittler and J. Illingworth. Relaxation labeling algorithms - a review. *Image and Vision Computing*, 3(4):206–216, 1985.

[72] J. Kittler and D. Pairman. Contextual pattern recognition applied to cloud detection and identification. *IEEE Trans.on Geo. and Remote Sensing*, 23(6):855–863, 1985.

[73] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *In Proc. European Conf. on Computer Vision*, 3:65–81, 2002.

[74] V. N. Kolmogorov and M. J. Wainwright. On optimality of tree-reweighted max-product message-passing. *In Proc. Uncertainty in Artificial Intelligence (UAI)*, July 2005.

[75] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. *In Proc. IEEE Int. Conf. CVPR*, pages 125–132, 2000.

[76] S. Krishnamachari and R. Chellappa. Delineating buildings by grouping lines with MRFs. *IEEE Trans. on Pat. Anal. Mach. Intell.*, 5(1):164–168, 1996.

[77] H. Kruppa. *Object Detection Using Scale-Specific Boosted Parts and a Bayesian Combiner*. PhD thesis, Department of Computer Science, ETH, Zurich, 2004.

[78] S. Kumar, J. August, and M. Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. *Fourth Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2005.

[79] S. Kumar and M. Hebert. Discriminative Random Fields: A discriminative framework for contextual interaction in classification. *in proc. IEEE International Conference on Computer Vision (ICCV)*, 2:1150–1157, October 2003.

[80] S. Kumar and M. Hebert. Man-made structure detection in natural images using a causal multiscale random field. *In Proc. IEEE Int. Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 1:119–126, 2003.

[81] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. *in adv. in Neural Information Processing Systems (NIPS)*, 2004.

[82] S. Kumar and M. Hebert. Multiclass discriminative fields for parts-based object detection. *Snowbird Learning Workshop, Utah*, 2004.

[83] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. *IEEE Int. Conf. on Computer Vision (ICCV)*, 2005.

[84] S. Kumar, A. C. Loui, and M. Hebert. Probabilistic classification of image regions using an observation-constrained generative approach. *In Proc. ECCV Workshop on Generative Models based Vision (GMBV)*, June 2002.

[85] S. Kumar, A. C. loui, and M. Hebert. An observation-constrained generative approach for probabilistic classification of image regions. *Image and Vision Computing, Special Issue on Generative Models Based Vision*, 21:87–97, 2003.

[86] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. Int. Conf. on Machine Learning*, 2001.

[87] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: Representation and clique selection. *In Proc. Twenty-First International Conference on Machine Learning (ICML)*, 2004.

[88] M. S. Langer. Large-scale failures of $f^{-\alpha}$ scaling in natural image spectra. *Journal of Optical Society of America*, 17(1):28–33, 2000.

[89] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Black. Face recognition: A convolutional neural-network approach. *IEEE Trans. Neural Networks*, 8(1):98–113, 1997.

[90] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. pages 255–258, 1998.

[91] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[92] Y. LeCun and F. J. Huang. Loss functions for discriminative training of energy-based models. *AI-Stats*, 2005.

[93] M. D. Levine and A. M. Nazif. An experimental rule-based image segmentation: A dynamic control strategy approach. *In Proc. Computer Vision, Graphics and Image Processing*, 32, 1985.

[94] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, Tokyo, 2001.

[95] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*, 72:101–121, 1998.

[96] D. G. Lowe. Object recognition from local scale-invariant features. *In Proc. International Conference on Computer Vision (ICCV 99)*, pages 1150–1157, 1999.

[97] D. Mackay. Bayesian non-linear modelling for the 1993 energy prediction competition. *In Maximum Entropy and Bayesian Methods*, pages 221–234, 1996.

[98] S. Mahamud and M. Hebert. Minimum risk distance measure for object recognition. *In Proc IEEE International Conference on Computer Vision (ICCV 03)*, 2003.

[99] H. Mayer. Automatic object extraction from aerial imagery- a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2):138–149, 1999.

[100] A. McCallum. Efficiently inducing features of conditional random fields. *In Proceedings of the 19th Conference in Uncertainty in Artifical Intelligence (UAI)*, 2003.

[101] A. McCallum, K. Rohanimanesh, and C. Sutton. Dynamic conditional random fields for jointly labeling multiple sequences. *NIPS'03 workshop on Syntax, Semantics and Statistic*, 2003.

[102] P. McCullagh and J. A. Nelder. *Generalised Linear Models.* Chapman and Hall, London, 1987.

[103] M. Mignotte, C. Collet, P. Prez, and P. Bouthemy. Sonar image segmentation using a hierarchical mrf model. *IEEE Trans. on Image Proc.*, 75(2):1216–1231, 2000.

[104] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *In European Conference on Computer Vision*, 1:128–142, 2002.

[105] T. P. Minka. *Algorithms for Maximum-Likelihood Logistic Regression.* Statistics Tech Report 758, Carnegie Mellon University, 2001.

[106] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD Thesis. Massachusetts Institute of Technology, Department of EE and CS, 2001.

[107] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[108] K. Murphy, A. Torralba, and W. T. Freeman. Using the forest to see the trees:a graphical model relating features, objects and scenes. *in Advances in Neural Information Processing Systems (NIPS 03)*, 2003.

[109] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[110] Y. Ohta. *A Region-Oriented Image-Analysis System by Computer*. Doctoral Dissertation, Information Science Department, Kyoto University, Kyoto, Japan, 1980.

[111] A. Oliva and A. Torralba. The shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[112] A. Olmos and E. Trucco. Detecting man-made objects in unconstrained subsea videos. *In Proc. British Machine Vision Conference*, pages 517–526, 2002.

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

[114] A. P. Pentland. *From Pixels to Predicates*. Ablex, Norwood, NJ, 1986.

[115] C. Peterson and J. Anderson. A mean-field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[116] W. Pieczynski and A. N. Tebbache. Pairwise markov random fields and its application in textured images segmentation. *In Proc. 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 106–110, 2000.

[117] S. D. Pietra, V. J. D. Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Machine Intelligence (PAMI)*, 19(4):380–393, 1997.

[118] Y. Qi, M. Szummer, and T. P. Minka. Bayesian conditional random fields. *AI & Statistics*, 2005.

[119] Y. Qi, M. Szummer, and T. P. Minka. Diagram structure recognition by bayesian conditional random fields. *In Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[120] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. *Neural Information Processing Systems (NIPS)*, December 2004.

[121] A. R. Rao and R. Jain. Knowledge representation and control in computer vision systems. *IEEE Expert*, 3(3):64–79, 1988.

[122] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans System, Man, Cybernatics*, SMC-6:420–433, 1976.

[123] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[124] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.

[125] Y. D. Rubinstein and T. Hastie. Discriminative vs informative learning. *In Proc. Third Int. Conf. on Knowledge Discovery and Data Mining*, pages 49–53, 1997.

[126] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. *In Advances in Neural Information Processing Systems 17,*, 2005.

[127] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Trans. on Pat. Anal. Mach. Intell.*, 22(5):504–525, 2000.

[128] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 00)*, 2000.

[129] F. Sha and F. Pereira. Shallow parsing with conditional random fields. *In Proc. Human Language Technology-NAACL*, 2003.

[130] A. Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhauser, Basel, 1993.

[131] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. *IEEE Int. Conf. on CVPR*, pages 235–241, 2003.

[132] T. M. Strat. *Natural Object Recognition*. Springer Verlag, New York, 1992.

[133] M. Szummer and Y. Qi. Contextual recognition of hand-drawn diagrams with conditional random fields. *Workshop on Frontiers in Handwriting Recognition*, 2004.

[134] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov network. *Neural Information Processing Systems Conference (NIPS03)*, 2003.

[135] M. Tipping. The relevance vector machine. *Advances in Neural Information Processing Systems-NIPS 12*, pages 652–658, 2000.

[136] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. *Adv. in Neural Information Processing Systems (NIPS)*, 2005.

[137] A. Torralba and P. Sinha. Statistical context priming for object detection. *In Proc. Int. Conf. on Computer Vision*, 2001.

[138] A. Vailaya, A. K. Jain, and H. J. Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31:1921–1936, 1998.

[139] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[140] P. Viola and M. Jones. Robust real-time object detection. *In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 01)*, 2001.

[141] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation and approximate ml estimation by pseudo-moment matching. *9th Workshop on AI Stat*, 2003.

[142] M. J. Wainwright, T. S. Jaakkola, and S. W. Alan. Tree-based reparameterization for approximate inference on loopy graphs. *In Advances NIPS*, 14, 2002.

[143] H. Wallach. *Efficient Training of Conditional Fields*. MS Thesis, Division of Informatics, University of Edinburgh, 2002.

[144] D. L. Waltz. *Understanding Line Drawing of Scenes with Shadows.* The Psychology of Computer Vision, P H Winston, ed. McGraw-Hill, New York, 1975.

[145] Y. Wang and Q. Ji. A dynamic conditional random field model for object segmentation in image sequences. *In Proc. IEEE Int. Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 1:264–270, 2005.

[146] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. *In Proc.IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 00)*, June 2000.

[147] J. Weinman, A. Hanson, and A. McCallum. Sign detection in natural images with conditional random fields. *In Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2004.

[148] M. Welling and Y. W. Teh. Belief optimization for binary networks: A stable alternative to loopy belief propagation. *In Proc. Uncertainty in Artificial Intelligence*, 2001.

[149] C. K. I. Williams and N. J. Adams. Dts: Dynamic trees. *Advances in Neural Information Processing Systems*, 11, 1999.

[150] C. K. I. Williams and F. V. Agakov. *An Analysis of Contrastive Divergence Learning in Gaussian Boltzmann Machines.* EDI-INF-RR-0120, Informatics Research Report, May 2002.

[151] P. Williams. Bayesian regularization and pruning using a laplacian prior. *Neural Computation*, 7:117–143, 1995.

[152] R. Wilson and C. T. Li. A class of discrete multiresolution random fields and its application to image segmentation. *IEEE Trans. on Pattern Anal. and Machine Intelli.*, 25(1):42–56, 2003.

[153] P. H. Winston. *Learning Structural Descriptions from Examples.* PhD Thesis, Project MAC, MIT, 1970.

[154] C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP*, 54:308–328, 1992.

[155] G. Xiao, M. Brady, J. A. Noble, and Y. Zhang. Segmentation of ultrasound b-mode images with intensity inhomogeneity correction. *IEEE Trans. on Medical Imaging*, 21(1):48–57, 2002.

[156] Y. Yakimovksy and J. A. Feldman. A semantics-based decision theory region analyzer. *In proc. Third Joint Conference on Artificial Intelligence*, pages 580–588, 1973.

[157] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. *In Advances Neural Information Processing Systems*, 13:689–695, 2001.

[158] A. L. Yuille. Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.

[159] S. C. Zhu, Y. N. Wu, and D. B. Mumford. Filters, random field and maximum entropy: Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):1–20, 1998.