Data-Dependent Spatial Context for Computer Vision with Conditional Markov Fields

Jerod J. Weinman
weinman@cs.umass.edu
Department of Computer Science
University of Massachusetts-Amherst

Technical Report UM-CS-2005-052

September 2005

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Computer Science

Directed by: Allen R. Hanson and Andrew McCallum

Abstract

The central task of computer vision is to infer properties of the real world from image data. To solve the simplified problem of assigning to image pixels labels that correspond to some real world property, we use a conditional Markov field, a recent model for computer vision. Traditional Markov fields model the likelihood of seeing a particular image that is *given*, whereas the sole task is to determine the plausible corresponding labels. Conditional Markov fields directly model the posterior probability of a labeling for a given image. The resulting model captures dependencies between neighboring image region labels in a data-dependent way and bypasses the difficult problem of modeling image formation. Focusing only on the labeling task allows us to relax unwarranted assumptions made in the traditional model, which results in improved performance. We demonstrate the model with an application detecting signs in natural images as an aid to the visually impaired.

Contents

List of Figures

List of Tables

Figure 1: Example of a natural image containing signs, with regions of interest identified.

1 Introduction

A digital image is a projection of high dimensional, high resolution reality into a lower dimensional and lower resolution space. The basic purpose of computer vision, manifested in various ways, is to recover some unknown aspects of that reality from the comparatively small amount of data in images. In a few cases, geometry and laws of the physical world guide this inference process in a deductive fashion. In most cases however, images contain insufficient information to uniquely determine truths about reality. Therefore, we are forced to reason from incomplete information about propositions concerning the real world. For this reason, computer vision and probability theory are inextricably linked.

One of the oldest problems in computer vision is to identify regions of interest, usually by assigning labels to image pixels or groups of pixels. As an example in this work, we seek to identify signs in natural images (see Figure 1). Our goal is to integrate with a wearable system that will recognize any detected signs as a navigational aid to the visually impaired. Generic sign detection is a difficult problem. Signs may be located anywhere in an image, exhibit a wide range of sizes, and contain an extraordinarily broad set of fonts, colors, shapes, arrangements, etc. It is therefore important for any sign detection method to robustly handle this diversity. That everything else in a natural image varies quite widely only compounds the problem. We will treat signs as a general texture class and seek to detect such a class in the presence of many others found in natural images.

The value of contextual information in computer vision tasks has been studied in various ways for many years (e.g., [60, 55, 29, 41, 31, 57, 8]). Two types of context are important for this problem: label context and data context. By data context, we mean the image data surrounding any region

in question. Data context can be of almost any scale, from the immediate neighbors of some region to the entire image or an image sequence. Similarly, label context consists of any labels surrounding a region in question. In the absence of label context, local regions are independently considered. Disregard for the (perhaps unknown) configuration of labels often leads to isolated false alarms and missed detections upon classification. Likewise, the absence of data context means ignoring potentially helpful image information from around the region being classified. Both contexts are simultaneously important.

For example, since neighboring regions often have the same label, we could encourage smoothness by penalizing label discontinuities. Such regularity is typically imposed without regard for the actual data in the regions. The downside is that when the local evidence for a label is weak, the continuity constraints typically override the local data. On the other hand, if the neighboring data is considered, local evidence for a region to be labeled "sign" might be weak, but witnessing a strong edge in the *neighboring* region could bolster belief in the presence of a sign at the site because the edge indicates a transition. Thus, it makes sense to consider the labels *and* data of neighboring regions when making classification decisions. This is exactly what the conditional random field (CRF) model will allow us to do [32].

2 Markov Fields

As already mentioned, probability and vision seem to be inextricably linked. In this section, we review a standard probabilistic model used for many contextual computer vision tasks and illustrate how a recently introduced version differs from the historical approach. Markov fields are a befitting framework for dealing with images because they facilitate explicit modeling of conditional independencies and naturally codify the "compatibility" between data and hypotheses.

2.1 Fundamentals

Markov random fields (MRFs) have a long history in computer analysis of images [36, 63]. Here we formally define and describe these probability distributions, elaborating on how they have been used in the past to model images.

Markov fields are a special case of the Gibbs probability distribution [24]. A Gibbs distribution for an unknown quantity $\mathbf{X} \in \Omega$ has the form[1]

$$p(\mathbf{x}) = \frac{1}{Z} e^{-U(\mathbf{x})}, \tag{1}$$

where

$$Z = \sum_{\mathbf{x} \in \Omega} e^{-U(\mathbf{x})}$$

is a normalizing constant called the partition function and $U : \Omega \to \mathbb{R}$ is the energy function that characterizes the distribution.

Markov fields, a type of undirected graphical model, impose some structure on $\mathbf{X}$ by relating it to a graph (V, E), where V is the set of graph vertices and $E \subset V \times V$ is the set of graph edges. The quantity $\mathbf{X} = (\mathbf{X}_v)_{v \in V}$ is constrained to be a vector indexed by nodes in the graph. Each dimension of the vector has a range of values $\mathbf{X}_v \in \Omega_v$ so that the total configuration space is the Cartesian product of each node's range[2]

$$\Omega \triangleq \bigotimes_{v \in V} \Omega_v.$$

Rather than an explicitly defining an energy function, Markov fields are often specified by a set of compatibility functions, or factors, which implicitly yield the energy function. As their name

[1] Strictly speaking we should write $p(\mathbf{X} = \mathbf{x})$ to represent the probability of the proposition that the unknown fixed value $\mathbf{X}$ equals a postulated value $\mathbf{x}$. To alleviate burdensome notation we simply use $p(\mathbf{x})$. The proposition being evaluated should be clear from context.

[2] The term "node" is often used to refer both to v and the quantity associated with it, $\mathbf{X}_v$. The notation $\mathbf{X}_v$ simply refers to the quantity from Ω_v; it could be (and often is) a vector itself.

implies, compatibility functions give a relative affinity for a particular labeling on the nodes. The compatibilities are indexed by a family of cliques

$$\mathcal{C} = \{C \subset V : (u, v) \in E \, \forall u, v \in C\} .$$

The configuration space of a clique C is the Cartesian product of the constituent nodes' ranges,

$$\Omega_C \triangleq \bigotimes_{v \in C} \Omega_v .$$

We denote only those values of $\mathbf{X}$ that concern the clique by the map

$$\mathbf{X}_C : \Omega \to \Omega_C, \tag{2}$$

which is simply the projection of the entire configuration space Ω onto Ω_C, the configuration space of the clique. The compatibility functions

$$\psi_C : \Omega_C \to \mathbb{R}_+ \cup \{0\} , \, C \in \mathcal{C}$$

therefore depend solely on the configuration of a particular clique C and have strictly positive ranges. Given the set of compatibilities $\{\psi_C\}_{C \in \mathcal{C}}$, the probability distribution for the Markov graph may be written[3]

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C (\mathbf{x}_C) , \tag{3}$$

where Z is the normalizing constant

$$Z = \sum_{\mathbf{x} \in \Omega} \prod_{C \in \mathcal{C}} \psi_C (\mathbf{x}_C) .$$

Logical independencies among the dimensions of $\mathbf{X}$ are determined by adjacencies in the graph. Specifically, the distribution (3) is Markovian with respect to the graph; i.e., given its graph neighbors, a node is logically independent of all other nodes:

$$p\left(\mathbf{x}_v \mid \mathbf{x}_{V \setminus v}\right) = p\left(\mathbf{x}_v \mid \mathbf{x}_{\mathcal{N}(v)}\right) .$$

Here v's neighbors in the graph are indicated by

$$\mathcal{N}(v) \triangleq \{w : (w, v) \in E\} .$$

Let the augmented neighborhood be denoted

$$\mathcal{N}_+(v) \triangleq \mathcal{N}(v) \cup \{v\} .$$

Thus, the connectivity of the graph codifies this logical premise: if given the values at all neighbors of a node, the values of any other nodes tell us nothing further.

It is useful to see some key relationships between the Gibbs and graphical forms, (1) and (3). For instance, we may define the energy function

$$U(\mathbf{x}) = \sum_{C \in \mathcal{C}} U_C (\mathbf{x}_C) \tag{4}$$

with potentials $U_C (\mathbf{x}_C) = \ln \psi_C (\mathbf{x}_C)$ to recover Gibbs from the field. Alternately, we could define the compatibilities $\psi_C (\mathbf{x}_C) = e^{-U_C(\mathbf{x}_C)}$ to recover the Markov field from an energy function, *so long as the potentials have the appropriate clique domains.* For many tasks, including computer vision, it is convenient to think of the probability distribution in terms of the underlying graph. Since the models (1) and (3) are equivalent, we gain the advantages of both.

[3]In the same way that $\mathbf{X}$ represents a fixed, unknown value and $\mathbf{x}$ represents a postulated value, $\mathbf{X}_v$ and $\mathbf{X}_C$ index the unknown $\mathbf{X}$, while $\mathbf{x}_v$ and $\mathbf{x}_C$ index the postulated $\mathbf{x}$.

2.2 Image Modeling with Markov Fields

The most basic Markov field models used for computer vision directly capture relationships between raw pixel values, emphasizing the graphical formulation (3). These include auto-models [2] and multi-level logistic (MLL) models [15], where potentials are defined over small neighborhoods of pixel values. Often used for noise removal, such models are limiting because in order to capture larger spatial dependencies, more pixels must be linked, creating a dramatic increase in computational complexity.

More recently, models that use image filters to capture complex patterns over larger neighborhoods have been used. For instance, to represent global texture appearance, the FRAME model [66, 67] employs potentials calculated from histograms (spatial marginals) of filter responses, rather than values among pixel cliques. To avoid aliasing when different textures have the same marginals, the multi-resolution MRF (MRMRF) model [37] also uses multi-resolution filters, but localizes the responses to pixel clique potentials. MRMRF incorporates meaningful filter information like FRAME, but uses the filter responses as input to pixel clique potentials that handle compatibility in a soft, gradual way, unlike all-or-nothing MLL models. Both FRAME and MRMRF models provide advantages over the MLL model because the underlying filters capture more meaningful information about the image than any tractably-sized cliques are able to. In these models, the image data interacts with the potentials in complex ways that make some convenient forms of approximate inference—a necessity as we show later—infeasible. A primary advantage of the newer conditional Markov field eliminates the issue.

A particular Markov field (specified by the energy, potentials, or compatibility functions) typically represents uncertainty about a single class of images (i.e., one texture), so when our concern broadens to a set of classes $\mathcal{Y}$ (i.e., multiple textures), we are compelled to indicate that a model represents a class-conditional probability distribution $p(\mathbf{x} \mid y)$, where $y \in \mathcal{Y}$. When the goal is to discriminate among data from different classes, there must not only be a model for every class we wish to distinguish, but there must also be an accurate model even for "background" classes of no interest. This is a non-trivial task for images because the real world contains a myriad of image "classes" (region types, textures, objects, etc.). To identify a particular class, separated from all others, one must either have an accurate model of the "all others" class, or an accurate model of the class boundary. The former is rarely accomplished well by a single class-conditional model. The large number of natural image classes therefore makes such texture models prohibitive for many tasks in real-world settings. The latter boundary model is more easily accomplished, and it is the approach taken by conditional Markov fields.

The MLL, FRAME, and MRMRF models all represent the probability of image data. Often we desire to model additional information in the Markov field, such as pixel labels or restored (as from a noisy degradation) pixel values. Nodes representing these quantities are easily added to a graph (thus growing V), and usually a separate vector $\mathbf{Y} = (Y_v)_{v \in S}$, indexed by the set of new nodes S, is introduced to store the quantities; the data vector is thus indexed by the remaining nodes $\mathbf{X} = (\mathbf{X}_v)_{v \in V \setminus S}$. This provides the foundation for modeling relationships between image data $\mathbf{X}$ and (in our case) region labels $\mathbf{Y}$ in any number of ways, depending solely on the clique structures for which we assign compatibility functions.

Typical Markov fields for assigning class labels $\mathbf{y}$ to data $\mathbf{x}$ model the interaction among labels independently of the interaction between local data and its label. That is to say, a joint model over data and labels can be factored into a prior on label assignments and the likelihood of observed data, conditioned on single site labels,

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}) &= p(\mathbf{x} \mid \mathbf{y})\, p(\mathbf{y}) \\ &\triangleq \frac{1}{Z} \prod_{v \in S} \psi_v \left(\mathbf{x}_{\rho(v)}, y_v \right) \prod_{C \in \mathcal{C}'} \psi_C \left(\mathbf{y}_C \right), \end{aligned} \tag{5}$$

where $\rho : S \to V \setminus S$ is a bijection pairing label nodes and local image data nodes. Thus ψ_v refers to a compatibility function on the edge between a label node and a data node, and

$$\mathcal{C}' = \{ C \in \mathcal{C} : (u, v) \in E \wedge u, v \in S \; \forall u, v \in C \}$$

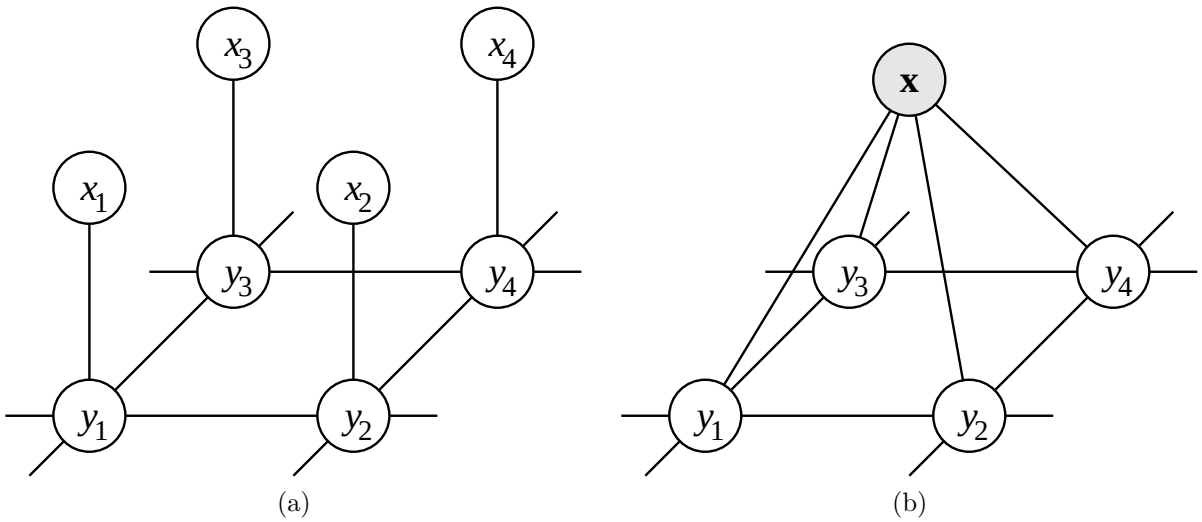


Figure 2: Markov fields: (a) Traditional joint Markov field over data $\mathbf{x}$ and labels $\mathbf{y}$ (cf. Eq. 5); (b) New conditional Markov field where data $\mathbf{x}$ is observed (cf. Eq. 9).

is a family of label node cliques. The neighborhood system of this graph (see Figure 2) indicates that when given a label node y_v, knowing any other label nodes would give us no further information about the corresponding (unknown) image data $\mathbf{x}_{\rho(v)}$.

The advantage of this model is that not only is the data likelihood $p(\mathbf{x} \mid \mathbf{y})$ factorized from the label prior $p(\mathbf{y})$, but the likelihood may be written as a product of independent probabilities

$$p(\mathbf{x} \mid \mathbf{y}) = \prod_{v \in S} p\left(\mathbf{x}_{\rho(v)} \mid y_v\right) \tag{6}$$

$$= \prod_{v \in S} \frac{1}{Z(y_v)} \psi\left(\mathbf{x}_{\rho(v)}, y_v\right). \tag{7}$$

This likelihood "field" is conditioned on $\mathbf{y}$, which allows the distribution to be factorized into an individual probability for each site v. Each of these probabilities may also be thought of as a "field" containing two nodes that correspond to a label that is given and a chunk of image data. Thus, we write $Z(y_v)$ because the normalization constant for this two node field depends on the given label value. A common choice for the class-conditional $p\left(\mathbf{x}_{\rho(v)} \mid y_v\right)$ is the Gaussian (a form of Gibbs distribution with a quadratic energy function) so that $Z(y_v)$ becomes the usual Gaussian normalization constant with a covariance matrix for a particular class.

One requirement of this formulation often overlooked is that the compatibility functions of the factored likelihood (7) should only depend on image data from the region v being labeled. Note that we have indicated only one compatibility function in (7), meaning the function's domain does not and cannot change. Therefore each $\mathbf{X}_v$ must have the same domain $\mathcal{X} = \Omega_v$ for all the "data" nodes $v \in V \setminus S$ and each Y_v must also have the same domain $\mathcal{Y} = \Omega_v \ \forall v \in S$ so that the compatibility of (7) has the signature $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+ \cup \{0\}$. This property that compatibility is invariant to location is called **homogeneity**. The consequence of this requirement is the compatibility function cannot depend at all on any part of $\mathbf{X}$ and $\mathbf{Y}$ save for $\mathbf{x}_v$ and y_v. Practically, this means once the image is partitioned into regions to be labeled, no features from image data outside those regions may be used while still truthfully retaining the Markov properties of the graph that gave rise to the factored likelihood (6) in the first place.

Due to these stringent locality requirements, factored likelihood (6) is a rather simplistic model of the data. When $\mathbf{y}$ denotes a denoised, ideal image and $\mathbf{x}$ the (given) noise-corrupted image, then the model is potentially quite suitable. However, as alluded in the introduction and elaborated above, assuming that neighboring image data is independent given the labels is too restrictive for many

computer vision tasks. For instance, in our detection task, while appearances may vary quite widely from sign to sign, the data comprising neighboring regions of a particular sign is highly dependent: similarity between data of neighboring regions is evidence they share the same label. Allowing more data (i.e., neighboring image region similarity) to influence site labels should increase the accuracy of the model.

When the goal is to discriminate among classes, the mode of the posterior $p(\mathbf{y} \mid \mathbf{x})$ is often used to label an image. By the probability product rule, this can be written in terms of the joint and a prior

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{p(\mathbf{y}, \mathbf{x})}{p(\mathbf{x})}. \tag{8}$$

Thus, choosing the most likely labeling for a particular image $\mathbf{x}$ means finding the $\mathbf{y}$ that maximizes $p(\mathbf{y}, \mathbf{X} = \mathbf{x})$ because $p(\mathbf{X} = \mathbf{x})$ behaves as constant in the posterior. This is the approach taken, for instance, by Freeman et al. [18]. As evidenced by this procedure, the probability of the data being observed is often irrelevant in computer vision tasks. Images happen. We are primarily interested in what may be inferred when we are *given* the observations. For such classification and labeling tasks, reasoning about the probability of seeing the data is unnecessary. Most importantly, the most commonly used joint model (5) makes inaccurate independence assumptions. To relax these assumptions would require adding edges to the graph G, thereby eliminating the convenient factored form (5) and introducing dramatically greater computational complexity.

Rather than calculate the posterior from the joint, Lafferty, McCallum, and Pereira [32] propose to model the posterior distribution *directly*, using a conditional random field of the form

$$p(\mathbf{y} \mid \mathbf{x}) \triangleq \frac{1}{Z(\mathbf{x})} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{y}_C, \mathbf{x}) \tag{9}$$

where Z is now an *observation-dependent* normalizer. The potentials are functions of label cliques, still facilitating neighboring label interaction, but they may also be functions of the *entire* observation. This differs markedly from the restrictions of (5) (see Figure 2). In the conditional Markov field framework, the probability of the observation no longer has to be evaluated, permitting the complex interactions between the data and labels that are prohibitively expensive in the joint model.

3 Conditional Markov Fields for Vision

In this section we discuss specifics of the probability distribution used to jointly classify image regions and then cover training procedures and prediction strategies. We conclude with a discussion of techniques used to prevent over-fitting model parameters to a small data sample.

3.1 Model

The graph topology commonly used for graphical modeling techniques applied to vision is the grid, as illustrated in Figure 2. Thus, the cliques may be identified by the observation connected to either a a single label node (1-cliques) or the edges between label nodes (2-cliques). We assume an *effectively* homogeneous, but anisotropic random field. Homogeneity means the probability distribution is translationally invariant, since cliques of the same class (such as label nodes) use the same features regardless of image location. With anisotropy, we intentionally spurn rotational invariance because it allows the model to learn any orientational bias of the labels that may be manifest in training data. Horizontal and vertical edges are considered different classes and thus have distinct potential functions. In the following, we formalize these notions.

Let $E_s = E_h \cup E_v$, where E_h and E_v are the sets of horizontal and vertical edges between label nodes S, and let $o \in V$ be the single node that represents the entire observation $\mathbf{x}$. Strictly speaking, homogeneity means the same compatibility functions are used throughout the graph, but this would require partitioning the image data before passing it to the potential, one of the strict limitations of earlier random fields, such as (5). Instead, since the CRF allows us to use potentials that are

functions of the entire observation, we compute some features of a scale *larger* than the region being labeled. As a result, the domains of our compatibility functions include that of the entire image Ω_o. We call our field *effectively* homogeneous because, while the unique compatibilities rely on features from a particular region of the image $\mathbf{x}$, all compatibilities of the same class use the those features in the same way. The anisotropy arises from the fact that we place compatibilities for horizontal and vertical edges in different classes, and thus they do not use the features identically.

The compatibilities describing the probability distribution are

$$\psi_v : \mathcal{Y} \times \Omega_o \to \mathbb{R}_+, \forall v \in S$$

for the single label node cliques and

$$\psi_{u,v} : \mathcal{Y} \times \mathcal{Y} \times \Omega_o \to \mathbb{R}_+, \forall (u,v) \in E_h \cup E_v$$

for the horizontal and vertical edges. The grid-structured random field takes the form

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) &= \frac{1}{Z(\mathbf{x})} \prod_{v \in S} \psi_v(y_v, \mathbf{x}) \prod_{(u,v) \in E_s} \psi_{u,v}(y_u, y_v, \mathbf{x}) && (10) \\ &= \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{v \in S} \boldsymbol{\lambda} \cdot F_v(y_v, \mathbf{x}) + \sum_{(u,v) \in E_s} \boldsymbol{\mu} \cdot G_{u,v}(y_u, y_v, \mathbf{x}) \right), && (11) \end{aligned}$$

where $F_v : \mathcal{Y} \times \Omega_o \to \mathbb{R}$ and $G_{u,v} : \mathcal{Y} \times \mathcal{Y} \times \Omega_o \to \mathbb{R}$ are scalar feature functions whose subscripts describe the location of the image $\mathbf{x}$ from which features are extracted; $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are real-valued feature weights. For brevity, rather than write the proper anisotropic energy

$$U(\mathbf{y}, \mathbf{x}) = \sum_{v \in S} \boldsymbol{\lambda} \cdot F_v(y_v, \mathbf{x}) + \sum_{(u,v) \in E_h} \boldsymbol{\mu}_h \cdot G_{u,v}(y_u, y_v, \mathbf{x}) + \sum_{(u,v) \in E_v} \boldsymbol{\mu}_v \cdot G_{u,v}(y_u, y_v, \mathbf{x}), \quad (12)$$

we unite the edge sums, with the implicit assumption that the appropriate parameter vector $\boldsymbol{\mu}_h$ or $\boldsymbol{\mu}_v$ is substituted for $\boldsymbol{\mu}$ depending on the membership of (u,v) in E_h or E_v. The vector $\boldsymbol{\theta} = \langle \boldsymbol{\lambda}, \boldsymbol{\mu}_h, \boldsymbol{\mu}_v \rangle$ specifies the linear weights on various features, indicating the parameterization of the distribution. The functions F_v couple features from an image region with a label for the region; likewise $G_{u,v}$ for pairs of regions and labels. Rather than having class-specific image features, we use one set of observation features, transforming them into the feature functions using the relationship

$$\begin{aligned} f^k_{y;v}(y_v, \mathbf{x}) &= \delta(y, y_v) f^k_v(\mathbf{x}) \\ g^j_{y,y';u,v}(y_u, y_v, \mathbf{x}) &= \delta(y, y_u) \delta(y', y_v) g^j_{u,v}(, \mathbf{x}) \end{aligned}$$

where $\mathbf{f}_v = \left(f^k_v\right)_{k=1\ldots K}$ is a vector of features for image region v (i.e., filter responses). Similarly, $\mathbf{g}_{u,v} = \left(g^j_{u,v}\right)_{j=1\ldots J}$ is a vector of image features from both regions u and v (i.e., differences between the responses of neighboring regions). The functions in the energy are composed as vectors

$$\begin{aligned} F_v &= \left(f^k_{y;v}\right)_{y \in \mathcal{Y}, k=1\ldots K} \\ G_{u,v} &= \left(g^j_{y,y';u,v}\right)_{y,y' \in \mathcal{Y} \times \mathcal{Y}, j=1\ldots J}. \end{aligned}$$

Thus $\boldsymbol{\lambda} \in \mathbb{R}^{K|\mathcal{Y}|}$ and $\boldsymbol{\mu}_h, \boldsymbol{\mu}_v \in \mathbb{R}^{J|\mathcal{Y}|^2}$. We discuss how these parameters are chosen in the next two sections.

Note that if $E_h = E_v = \emptyset$ so that there are no edges between the label nodes, the label probabilities are independent of each other. Such a label context-free model is a baseline for demonstrating the improvement due to contextual modeling with a Markov field, shown later by experimental results in section 5.

We now present some useful additional notation. The expected value of a function $f : \Omega \to \mathbb{R}$ with respect to a distribution p over Ω is abbreviated

$$p\left[f\right] \triangleq \sum_{\mathbf{x} \in \Omega} f\left(\mathbf{x}\right) p\left(\mathbf{x}\right).$$

When the expectation is conditional, as will often be the case with CRFs, we write

$$p\left[f \mid \mathbf{x}\right] \triangleq \sum_{\mathbf{y} \in \Omega_S} f\left(\mathbf{y}, \mathbf{x}\right) p\left(\mathbf{y} \mid \mathbf{x}\right). \tag{13}$$

When the function of interest $f : \Omega_C \times \Omega_o \to \mathbb{R}$ (as in our feature functions) only depends on some subset of the variables $\mathbf{y}_C$, the expectation is reduced to a marginal expectation:

$$\begin{aligned} p_C\left[f \mid \mathbf{x}\right] &\triangleq \sum_{\mathbf{y} \in \Omega_S} f\left(\mathbf{y}_C, \mathbf{x}\right) p\left(\mathbf{y} \mid \mathbf{x}\right) \\ &= \sum_{\mathbf{y} \in \Omega_C} f\left(\mathbf{y}_C, \mathbf{x}\right) \sum_{\mathbf{y} \in \Omega_{S \setminus C}} p\left(\mathbf{y} \mid \mathbf{x}\right) \\ &= \sum_{\mathbf{y} \in \Omega_C} f\left(\mathbf{y}_C, \mathbf{x}\right) p_C\left(\mathbf{y}_C \mid \mathbf{x}\right) \end{aligned}$$

with the marginal probability given by

$$p_C\left(\mathbf{y}_C \mid \mathbf{x}\right) \triangleq \sum_{\mathbf{y} \in \Omega_{S \setminus C}} p\left(\mathbf{y} \mid \mathbf{x}\right).$$

3.2 Bayesian Prediction

Our ultimate goal is to make predictions about the labels corresponding to images yet unseen. With little prior knowledge of how labels relate to images it would be difficult to say anything substantial on the matter. Therefore, we go to the task of collecting a sample of labeled images. Let our data be

$$\mathcal{D} = \left(\left\langle \mathbf{y}^{(i)}, \mathbf{x}^{(i)} \right\rangle\right)_{i=1\ldots|\mathcal{D}|}, \tag{14}$$

where each $\mathbf{x}^{(i)}$ is an image and $\mathbf{y}^{(i)}$ are the corresponding labels. Without further information, we use the probability $p\left(\mathbf{y} \mid \mathbf{x}, \mathcal{D}\right)$ to reason about labelings. Our model, based on some prior knowledge of the problem, asserts that there are underlying features (e.g., F and G) relevant to the labeling. However, these features have parameters that must somehow be dealt with before we can make predictions. Fortunately, probability theory provides a well-principled method for handling such parameters.

The Bayesian paradigm calls for parameters to be treated as unknowns. Our predictive distribution would result from the integral

$$p\left(\mathbf{y} \mid \mathbf{x}, \mathcal{D}\right) = \int p\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta} | \mathcal{D}\right) \mathrm{d}\boldsymbol{\theta}. \tag{15}$$

Note in (15) that (i) $\mathbf{y}$ is logically independent of $\mathcal{D}$ given $\boldsymbol{\theta}$, and (ii) $\boldsymbol{\theta}$ is logically independent of $\mathbf{x}$ given $\mathcal{D}$. Here, the first term $p\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right)$ would be of the form specified generically by (9) and specifically by (10), (11). The other term we may understand by using Bayes' Theorem.

If we have prior knowledge about the parameters encoded by the probability $p\left(\boldsymbol{\theta}\right)$, we may write

$$p\left(\boldsymbol{\theta} | \mathcal{D}\right) \propto p\left(\mathcal{D} | \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right). \tag{16}$$

Next we assume that upon being given the parameters, the data forms an exchangeable sequence of independent pairs, so that

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{i=1}^{|\mathcal{D}|} p\left(\mathbf{y}^{(i)}, \mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right) \\ &= \prod_{i=1}^{|\mathcal{D}|} p\left(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta}\right) p\left(\mathbf{x}^{(i)}\right). \end{aligned} \tag{17}$$

Factoring the likelihood in this fashion has two purposes. First, we explicitly indicate that the parameters are hypotheses concerning only the relationship between labels and a given image, rather than any intrinsic properties of the image. Second, since our objective is to make predictions about labels given images, it allows us to focus on the conditional term $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ and essentially ignore the image "prior." In fact, the prior probabilities of the sample images are irrelevant for prediction because the $p\left(\mathbf{x}^{(i)}\right)$ terms cancel *completely* from (15).

Using $p(\mathbf{y} \mid \mathbf{x}, \mathcal{D})$ to make predictions would fully utilize any prior knowledge codified by $p(\boldsymbol{\theta})$. The integral (15) is an average of $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ weighted by the posterior probability of different parameters $\boldsymbol{\theta}$ given the sample of labeled images. Unfortunately, the integral is rather unwieldy, so we will be forced to make an approximation. If the posterior (16) is strongly peaked at its mode $\widehat{\boldsymbol{\theta}}$, then there is little other support and we have

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) \approx p\left(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\theta}}\right). \tag{18}$$

This is a typical approximation, and one we shall subsequently appeal to. However, a strategy to approximate the full integral by approximating the parameter posterior (16) was recently introduced by Qi, Szummer, and Minka [47]. The advantage is most pronounced on a synthetic problem when the labeled data is scant, thus rendering the likelihood (17) less peaked and requiring more reliance on prior information from $p(\boldsymbol{\theta})$. Error rates on a real problem (FAQ segmentation), though statistically significant, improved from just under 1.5% to just over 0.5%. This could be because the prior is not particularly helpful or the likelihood is strongly peaked.

Clearly, more important than the outright accuracy of the approximation (18) will be its effect on prediction: if $\mathbf{Y}$ is the true labeling of an image $\mathbf{x}$, the important matter is how much worse (if any) will the prediction $\hat{\mathbf{y}}$ resulting from $p\left(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\theta}}\right)$ be than that resulting from $p(\mathbf{y} \mid \mathbf{x}, \mathcal{D})$. Although we do not have such comparisons for our applications, experiments in other real domains indicate only a slight absolute improvement of 1-2% [47, 48]. We discuss prediction strategies in section 3.5 and parameter priors in section 3.6. Next, we discuss the issue of finding the mode $\widehat{\boldsymbol{\theta}}$.

3.3 Likelihood and Pseudo-Likelihood

To approximate the true predictive distribution $p(\mathbf{y} \mid \mathbf{x}, \mathcal{D})$ we must find the most likely hypothesis for the data: the parameters that maximize the posterior probability

$$\begin{aligned} \widehat{\boldsymbol{\theta}} &= \arg\max_{\theta \in \Theta} p(\boldsymbol{\theta}|\mathcal{D}) \\ &= \arg\max_{\theta \in \Theta} p(\boldsymbol{\theta}) \prod_{i=1}^{|\mathcal{D}|} p\left(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta}\right) \\ &= \arg\max_{\theta \in \Theta} \log p(\boldsymbol{\theta}) + \sum_{i=1}^{|\mathcal{D}|} \log p\left(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta}\right) \end{aligned} \tag{19}$$

We shall ignore the first prior term for now and focus on the likelihood. The specific form of the so-called log-likelihood of (11) is given by

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) \triangleq \sum_{i=1}^{|\mathcal{D}|} \log p\left(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}; \boldsymbol{\theta}\right)$$

$$= \sum_i \left(\sum_{v \in S^{(i)}} \boldsymbol{\lambda} \cdot F_v \left(y_v^{(i)}, \mathbf{x}^{(i)} \right) + \sum_{(u,v) \in E_s^{(i)}} \boldsymbol{\mu} \cdot G_{u,v} \left(y_u^{(i)}, y_v^{(i)}, \mathbf{x} \right) - \log Z \left(\mathbf{x}^{(i)} \right) \right) \tag{20}$$

The observation-dependent normalizers have the form

$$Z \left(\mathbf{x}^{(i)} \right) = \sum_{\mathbf{y} \in \mathcal{Y}^{\left| S^{(i)} \right|}} \exp \left(\sum_{v \in S^{(i)}} \boldsymbol{\lambda} \cdot F_v \left(y_v, \mathbf{x}^{(i)} \right) + \sum_{(u,v) \in E_s} \boldsymbol{\mu} \cdot G_{u,v} \left(y_u, y_v, \mathbf{x} \right) \right) \tag{21}$$

The Hessian of (20) is positive semi-definite, so if the Hessian of the prior term is also positive semi-definite, the optimization problem is convex and the optimal parameters $\widehat{\boldsymbol{\theta}}$ may be found via quasi-Newton gradient ascent methods[4], such as L-BFGS [7].

The gradient is given by

$$\begin{aligned} \nabla_{\boldsymbol{\lambda}} \mathcal{L} \left(\boldsymbol{\theta}, \mathcal{D} \right) &= \sum_i \sum_{v \in S^{(i)}} \left(F_v \left(y_v^{(i)}, \mathbf{x}^{(i)} \right) - p_v \left[F_v \mid \mathbf{x}^{(i)}, \boldsymbol{\theta} \right] \right) \\ \nabla_{\boldsymbol{\mu}} \mathcal{L} \left(\boldsymbol{\theta}, \mathcal{D} \right) &= \sum_i \sum_{(u,v) \in E_a} \left(G_{u,v} \left(y_u^{(i)}, y_v^{(i)}, \mathbf{x} \right) - p_{u,v} \left[G_{u,v} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta} \right] \right) \end{aligned} \tag{22}$$

where the expectations are over node and edge marginals, and E_a is the appropriate horizontal or vertical edge set, depending on whether the parameter vector $\boldsymbol{\mu}$ corresponds to $\boldsymbol{\mu}_h$ or $\boldsymbol{\mu}_v$.

Unfortunately, the summation over $\mathcal{Y}^{|S|}$ required by (21) is exponential in the number of vertices, making exact calculations of both the likelihood and the gradient problematic for any but the smallest graphs. This is a well-known limitation of the model, and as a result, many approximations are well-studied. We discuss these in more detail in Section (3.4).

A simple alternative is to use the so-called pseudo-likelihood (PL) estimator [3, 4]. This approximation takes advantage of the conditional independencies of the model by giving it more information than would will actually be available for prediction in order to eliminate the intractable sums. Specifically, from the grid structure of (11) and its Markovian properties we have that

$$p \left(y_v \mid \mathbf{y}_{\mathcal{N}(v)}, \mathbf{x} \right) = \frac{1}{Z \left(\mathbf{y}_{\mathcal{N}(v)}, \mathbf{x} \right)} \exp \left(\boldsymbol{\lambda} \cdot F_v \left(y_v, \mathbf{x} \right) + \sum_{u \in \mathcal{N}(v)} \boldsymbol{\mu} \cdot G_{u,v} \left(y_u, y_v, \mathbf{x} \right) \right), \tag{23}$$

with the corresponding partition function

$$Z \left(\mathbf{y}_{\mathcal{N}(v)}, \mathbf{x} \right) = \sum_{y_v \in \mathcal{Y}} \exp \left(\boldsymbol{\lambda} \cdot F_v \left(y_v, \mathbf{x} \right) + \sum_{u \in \mathcal{N}(v)} \boldsymbol{\mu} \cdot G_{u,v} \left(y_u, y_v, \mathbf{x} \right) \right).$$

This partition function is tractable since the summation is only over the label for a single node.

Rather than optimizing the full joint log-likelihood, we then maximize the product of these conditional probabilities for all nodes in the data. The PL estimate is thus

$$\begin{aligned} \mathcal{PL} \left(\boldsymbol{\theta}, \mathcal{D} \right) &\triangleq \log \prod_{i=1}^{|\mathcal{D}|} \prod_{v \in S^{(i)}} p \left(y_v^{(i)} \mid \mathbf{y}_{\mathcal{N}(v)}^{(i)} \mathbf{x}^{(i)} \right) \\ &= \sum_i \left(\sum_{v \in S^{(i)}} \boldsymbol{\lambda} \cdot F_v \left(y_v^{(i)}, \mathbf{x}^{(i)} \right) + \right. \end{aligned}$$

[4]Generally the number of features is quite large, which precludes the use of actual second order methods for optimization since they require space quadratic in the number of features.

$$2 \sum_{(u,v)\in E_s^{(i)}} \boldsymbol{\mu} \cdot G_{u,v}\left(y_u^{(i)}, y_v^{(i)}, \mathbf{x}^{(i)}\right) - \sum_{v\in S^{(i)}} \log Z\left(\mathbf{y}_{\mathcal{N}(v)}^{(i)} \mathbf{x}^{(i)}\right)\Bigg). \quad (24)$$

Note that the edge energies are now counted twice, since each edge acts as a bridge to a left (top) or right (bottom) neighbor, depending on which is given. The gradient is given by

$$\begin{aligned}
\nabla_{\boldsymbol{\lambda}} \mathcal{PL}(\boldsymbol{\theta}, \mathcal{D}) &= \sum_i \sum_{v\in S^{(i)}} \left(F_{u,v}\left(y_v^{(i)}, \mathbf{x}^{(i)}\right) - p_v\left[F_v \mid \mathbf{y}_{\mathcal{N}(v)}^{(i)}, \mathbf{x}^{(i)}, \boldsymbol{\theta}\right]\right) \\
\nabla_{\boldsymbol{\mu}} \mathcal{PL}(\boldsymbol{\theta}, \mathcal{D}) &= \sum_i \sum_{(u,v)\in E_a} \Big(2G_{u,v}\left(y_u, y_v, \mathbf{x}^{(i)}\right) - \\
&\qquad \left(p_u\left[G_{u,v} \mid \mathbf{y}_{\mathcal{N}(u)}^{(i)}, \mathbf{x}^{(i)}, \boldsymbol{\theta}\right] + p_v\left[G_{u,v} \mid \mathbf{y}_{\mathcal{N}(v)}^{(i)}, \mathbf{x}^{(i)}, \boldsymbol{\theta}\right]\right)\Big)
\end{aligned}$$

Due to the Markov properties of the graph, a function over the augmented neighborhood $f : \Omega_{\mathcal{N}_+(v)} \times \Omega_o \to \mathbb{R}$ has the tractable conditional expectation

$$p_v\left[f \mid \mathbf{y}_{\mathcal{N}(v)}, \mathbf{x}, \boldsymbol{\theta}\right] = \sum_{y_v\in\mathcal{Y}} f\left(y_{\mathcal{N}_+(v)}, \mathbf{x}\right) p\left(y_v \mid \mathbf{y}_{\mathcal{N}(v)} \mathbf{x}, \boldsymbol{\theta}\right).$$

As our experiments will show, pseudo-likelihood can potentially be an extremely poor approximation. In the next section, we review an alternate method for approximating the true likelihood that gives a better estimate of $\boldsymbol{\theta}$ since it yields predictions superior to those resulting from PL estimation.

3.4 Approximate Inference

The problem of inference for Markov fields typically involves calculating the partition function and the expectations required for the gradient, as described in the previous section.[5] In this section we briefly describe two traditional methods of inference (MCMC and mean field) before covering TRP, which we ultimately use for our experiments.

For the more typical generative models like (3) and (5), Markov chain Monte Carlo (MCMC) [63] has often been used to garner a sample from the distribution. The expectations and partition function may then be approximated with the sample. Since this is a discriminative model, the distribution is conditioned on the image data $\mathbf{x}$, and thus a sample must be drawn *for every observation*. Convergence of a Markov chain is required before sampling, so the approximation time is potentially much longer than for generative joint models.

An entirely different framework for approximate inference is based on a variational characterization of the Gibbs distribution p on Ω with energy U and partition function Z. For any other probability distribution q on Ω, the relative entropy of q and p is

$$\begin{aligned}
D(q \parallel p) &= q\left[\log \frac{q}{p}\right] \\
&= q[\log q] - q\left[\log \frac{1}{Z} e^{-U}\right] \\
&= q[U] - H(q) + \log Z,
\end{aligned}$$

where $H(q) = -q[\log q]$ is the entropy of q. Of course,

$$D(q \parallel p) = 0 \quad \Leftrightarrow \quad p \equiv q,$$

so minimizing the relative entropy between an arbitrary distribution q and the Gibbs distribution p that we are interested in is an intuitive way for finding reasonable approximations. Instead of sampling (as in MCMC) to garner the likelihood and expectations necessary for optimizing $\boldsymbol{\theta}$, one

[5] This is true when potentials U_C are linear in the parameters, as in our case.

could choose a distribution q that is easier than p to work with, minimize the relative entropy between them, and then instead use the likelihood and expectations resulting from q. The quantity $q[U] - H(q)$ is the expected energy with respect to q minus the entropy of q, commonly called the Gibbs **free energy**. Since Z does not depend on q, it suffices to minimize the free energy.

There are many techniques for performing approximate inference using the free energy. One approach, called mean field, is to constrain q to a class of tractable distributions $\mathcal{Q}$, which allows the minimal free energy to be calculated in a straightforward manner (see e.g., [43])

$$\min_{q \in \mathcal{Q}} q[U] - H(q).$$

This technique is called an *inner approximation*, since the optimization region $\mathcal{Q}$ does not include all possible distributions over Ω (e.g., it excludes, of course, the intractable Gibbs distribution p). We discuss next an alternate *outer approximation* that enlarges, rather than restricts, the constraint set.

Rather than minimize the exact free energy for a restricted set of distributions (the inner approximation), an alternate technique approximates the overall free energy as the sum of free energies for small regions of the graph

$$\mathcal{R} = \left\{ R = \bigcup_{C \in \mathcal{C}'} C : \mathcal{C}' \subset \mathcal{C} \right\},$$

comprised of unions of the cliques $\mathcal{C}$ that parameterize the distribution [65]. Recall that the Gibbs energy for a Markov field can be decomposed into clique potentials U_C. The energy for a region $U_R : \Omega_R \to \mathbb{R}$ is defined as the sum of the potentials of cliques comprising the region.[6] The arbitrary probability distribution q for Ω is then replaced by a set of "beliefs" $b = (b_R)_{R \in \mathcal{R}}$ over the regions, with $b_R : \Omega_R \to \mathbb{R}$. The region-based expected energy and entropy terms now sum over the domain Ω_R of each region $R \in \mathcal{R}$, rather than over all of Ω:

$$\begin{aligned} b[U] &\triangleq \sum_{R \in \mathcal{R}} c_R \sum_{\mathbf{x}_R \in \Omega_R} b_R(\mathbf{x}_R) U_R(\mathbf{x}_R) \\ H(b) &\triangleq -\sum_{R \in \mathcal{R}} c_R \sum_{\mathbf{x}_R \in \Omega_R} b_R(\mathbf{x}_R) \log b_R(\mathbf{x}_R), \end{aligned}$$

where c_R is a "counting number" that ensures each node in the graph is counted only once (since regions may overlap).[7] Constraining the beliefs to the set

$$\mathcal{B} = \left\{ b : \sum_{\mathbf{x}_R \in \Omega_R} b_R(\mathbf{x}_R) = 1 \ \forall R \in \mathcal{R} \wedge b_R(\mathbf{x}_R) \in [0,1] \ \ \forall R \in \mathcal{R}, \mathbf{x}_R \in \Omega_R \right\}$$

ensures they behave like probabilities (non-negative and normalized). Thus, solving the region-based free energy problem

$$\min_{b \in \mathcal{B}} b[U] - H(b)$$

yields b_R that represent approximations of the marginals p_R. In fact, when the pseudo-marginals b_R are exact $b_R \equiv p_R$, then the expected energy is exact $p[U] = b[U]$. This technique is an outer approximation because it admits values for the pseudo-marginals that may not be the marginal probabilities of any distribution on Ω.

The well-known Bethe free energy is the simplest example of region-based free energy approximation, with all single nodes and pairs of adjacent nodes as regions. Loopy belief propagation (BP) [42] is an iterative message-passing algorithm for finding stationary points of the constrained Bethe free energy minimization problem. However, it is not guaranteed to converge. A relatively new algorithm for finding stationary points is called tree reparameterization (TRP) [58], which tends to

[6] Note that since regions may overlap, the Gibbs energy is not the sum of the region energies.

[7] The use of $b[U]$ and $H(b)$ is a slight abuse of notation since b does not represent a true probability distribution over Ω.

converge faster and more often than loopy BP. We provide here some high-level interpretations of the algorithm and refer the reader to [58] for the great notational details required to fully describe what amounts to a succinct algorithm.

TRP is simply a tree-based schedule for belief propagation that relies on two key facts. First, exact inference in graphical models without loops (unlike our lattice model) is very efficient because of the junction tree algorithm (see e.g., [33]). Second, since a Markov field is a normalized product of compatibility functions, the compatibilities may be changed without altering the distribution, so long as the product remains the same. Indeed, the reparameterization in the name TRP stems from altering compatibilities in this invariant fashion.

TRP operates iteratively by first computing the exact marginals on a spanning tree embedded in the original loopy graph. Then the values of these marginals are used to reparameterize the compatibilities from the spanning tree, and the process repeats with a different embedded spanning tree until the parameterization (marginal distributions) converges.

We use TRP in our experiments, comparing the results of approximating the full likelihood with TRP to the results of estimating parameters with pseudo-likelihood.

3.5 Predicting Labels

Given the image data, the model simply yields a posterior distribution on labels. When we need to pick a hard and fast label for each region of the image, the question becomes what to do with that distribution; what estimator do we use? A simple, oft-used answer is to find the most likely labeling. That is, use *maximum a posteriori* (MAP) estimation:

$$\begin{aligned} \hat{\mathbf{y}} &= \arg\max_{\mathbf{y}\in\mathcal{Y}^{|S|}} p\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \\ &= \arg\max_{\mathbf{y}\in\mathcal{Y}^{|S|}} U\left(\mathbf{y}, \mathbf{x}\right). \end{aligned}$$

The second line follows from the normalizer $Z\left(\mathbf{x}\right)$'s independence from $\mathbf{y}$ and taking the natural logarithm (a strictly monotonic function that does not alter the ordering of values) to yield the energy (12). Unfortunately, once again we have the issue of an intractable search space $\mathcal{Y}^{|S|}$. For a binary labeling problem, an exact maximum can be found by a reduction to a max-cut/min-flow problem (see e.g., [63, pp. 132–134]). However, the reduction is valid only when the node interactions (e.g., $\boldsymbol{\mu} \cdot G$) are positive, a restriction that is likely too strong for our purposes. Fortunately, a slight variation of TRP allows approximate MAP estimates to be calculated. The MAP estimator has an important caveat: poor predictions can result when the maximum of the posterior is not representative of most of the other likely labelings [22, 17]. In other words, the highest peak is not at the center of most of the posterior volume.

An alternative method for prediction is called maximum posterior marginal (MPM) estimation:

$$\hat{y}_v = \arg\max_{y_v\in\mathcal{Y}} p\left(y_v \mid \mathbf{x}\right), \forall v \in S.$$

It corresponds to choosing the label at each node that maximizes the probability with all other node labelings marginalized. This can often be a more effective way of considering the probabilities of all the labelings, rather than simply the maximum (joint) labeling, as in MAP. Marginalization, however, suffers from the same computational complexity problems as likelihood and MAP. Once again, TRP comes through because its very nature is to reveal (approximate) marginals on the nodes, which may be used for the MPM estimator.

An extremely simple alternative to MAP and MPM is called iterated conditional modes (ICM). Given some initial labeling $\mathbf{y}^0$, subsequent labels are given by

$$y_v^{k+1} = \arg\max_{y_v\in\mathcal{Y}} p\left(y_v \mid y_{\mathcal{N}(v)}^k, \mathbf{x}\right), \forall v \in S$$

until the convergence criterion $\mathbf{y}^{k+1} = \mathbf{y}^k$ is reached or an iteration limit $k > k_{\max}$ is exceeded. Of course, the conditional probability is quite tractable, as shown by (23). Often, the initial labeling

comes from the local compatibility maximum

$$y_v^0 = \arg\max_{y_v \in \mathcal{Y}} \psi_v\left(y_v, \mathbf{x}\right), \forall v \in S.$$

This method is quite fast, but in detection problems with highly skewed class distributions and weak local evidence it can lead to lower detection rates.

In practice, the MAP estimate tends to be conservative, trying to give the most correct labels, while MPM tends to give higher detection rates. These three methods are compared in our experiments.

3.6 Parameter Priors

Our earlier discussion of Bayesian prediction in section (3.2) acknowledged a role for prior information about our model parameters but said nothing of what that prior knowledge might be. In this section we review some common prior probabilities for the parameters and discuss some of their qualitative interpretations. First, we point out a useful property of the Gibbs distribution (1).

Whereas Laplace's "Principle of Insufficient Reason" assigns equal probabilities to propositions when there is no reason to do otherwise, the so-called "Maximum Entropy Method" (hence, MaxEnt), put forth by Jaynes [27], is a technique for assigning prior probabilities from partial information. In particular, when given an expectation of some function $\bar{f} \triangleq p\left[f\right]$ but not p itself, for any other inference task MaxEnt says to choose a p such that it has the given expectation but otherwise has maximum entropy. This sets up a constrained optimization problem for a distribution that agrees with the partial information but is otherwise "maximally noncommittal with regard to missing information," as Jaynes says. The result is precisely a Gibbs distribution of the form

$$p\left(\mathbf{x} \mid \bar{F}\right) \propto \exp\left(-\boldsymbol{\lambda} \cdot F\left(\mathbf{x}\right)\right) \tag{25}$$

where the functions comprising vector $F = (f)$ are those whose expectations $\bar{F}$ are given, and the parameters $\boldsymbol{\lambda}$ are the Lagrangian multipliers from the optimization, as determined by $\bar{F}$. Thus, some information is given, and further inference is based on a distribution that agrees with that information but reflects our otherwise maximal uncertainty. Not coincidentally, this maximum entropy distribution corresponds to the functional form of the Markov field specified by (11).

The relationship comes about via the following. If we fix the distribution on images $p\left(\mathbf{x}\right)$ to be the empirical relative frequency of $\mathbf{x}$

$$p\left(\mathbf{x}\right) \triangleq \frac{\left|\left\{i : \mathbf{x} = \mathbf{x}^{(i)}\right\}\right|}{|\mathcal{D}|} \tag{26}$$

and take the expectation of interest to be

$$\bar{F} \triangleq p\left[F\right] = \sum_{\mathbf{y}} \sum_{\mathbf{x}} p\left(\mathbf{y} \mid \mathbf{x}\right) p\left(\mathbf{x}\right) F\left(\mathbf{y}, \mathbf{x}\right),$$

then the Markov field (11) arises from choosing the conditional distribution $p\left(\mathbf{y} \mid \mathbf{x}\right)$ that matches averages from the training sample (14)

$$\tilde{F} \triangleq \frac{1}{|\mathcal{D}|} \sum_{i} F\left(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}\right), \tag{27}$$

so that

$$\bar{F} = \tilde{F} \tag{28}$$

but otherwise has maximum conditional entropy

$$\sum_{\mathbf{y}} \sum_{\mathbf{x}} p\left(\mathbf{y} \mid \mathbf{x}\right) p\left(\mathbf{x}\right) \log p\left(\mathbf{y} \mid \mathbf{x}\right)$$

($p(\mathbf{x})$ still fixed). See Berger, Della Pietra, and Della Pietra [1] for further details. It should be clear that at the mode of the log-likelihood (20) where the gradient (20) is zero, the constraints (28) are met.

In its original formulation, MaxEnt would not be considered to "overfit" to data. Rather, the information $\tilde{F}$ is taken as a reliable given, and p is then chosen to assume that and nothing more. For this reason, MaxEnt has proven to be a useful method for assigning prior probabilities. As initially utilized in statistical physics, $p[F]$ represented a macroscopic quantity of some matter, and the resulting distribution would be used to predict other macroscopic quantities $p[G]$ of the *same matter*. An important, perhaps subtle, approximation has been made in the recent applications of MaxEnt.

If $\tilde{F}$ was calculated from all possible image/label pairs, or at least only the pairs we are interested in reasoning about, no further mechanisms would be needed; we could make plausible inferences about the expected value of any function $g(\mathbf{y}, \mathbf{x})$ on the image/label pairs in the data because no bias *beyond the data* would be introduced by MaxEnt. However, our data is now of a different sort than that of the original maximum entropy formulation. Rather than being given a reliable macroscopic quantity with which we want to predict other macroscopic quantities, we are instead given a handful of molecules and asked to make predictions about *other individual* molecules from the same cloud of gas. One must then wonder how similar the molecules in this handful are to the others in the cloud, both individually and collectively. Qualitatively, we would like to say "somewhat similar." But how much is somewhat?

We must acknowledge our data, or $\tilde{F}$, is an *estimate,* rather than a true, reliable average. In our physics example, the true macroscopic quantity would thus equal our estimate of it plus some error:

$$\bar{F} = \tilde{F} + \boldsymbol{\epsilon}. \tag{29}$$

Of course, we do not know what our error is, or else we would know the true quantity, returning us to the original MaxEnt method. It seems that we are left with two options: (i) stay with the probability *assignment* framework, of which MaxEnt and Insufficient Reason are primary examples, or (ii) shift to the Bayesian probability *updating* framework. One perhaps frustrating property of probability theory is that it does not dictate how to assign probabilities, only how to update them in light of new information. For progress to be made, the infinite regress of postulating hypothesis and constructing priors for the hypothesis must be stopped somewhere. In this case, we could shift our problem to one of assigning prior probabilities for the errors $\boldsymbol{\epsilon}$ and use Bayesian reasoning

$$p\left(\mathbf{x} \mid \tilde{F}\right) = \int p\left(\mathbf{x} \mid \tilde{F} + \boldsymbol{\epsilon}\right) p(\boldsymbol{\epsilon}) \, \mathrm{d}\boldsymbol{\epsilon} \tag{30}$$

for our predictions, with the MaxEnt distribution as the presumed model $p\left(\mathbf{x} \mid \tilde{F} + \boldsymbol{\epsilon}\right)$. But the Bayesian integral (30) alone does not suffice to solve our problem of assigning a probability to $\mathbf{x}$, for we must still specify the prior probability for $\boldsymbol{\epsilon}$. Without additional information, assigning $p(\boldsymbol{\epsilon})$ is of the same fundamental difficulty as assigning $p(\mathbf{x})$. Worse yet, adding the level of uncertainty introduces a computationally challenging integral. This is only exacerbated by the fact that while $p\left(\mathbf{x} \mid \bar{F}\right)$ is theoretically specified by the expectations $\bar{F}$, it is actually parameterized by $\boldsymbol{\lambda}$, which in general cannot be solved for in closed form from $\bar{F}$. Although not a theoretical argument against pushing further down the hypothesis chain by introducing $p(\boldsymbol{\epsilon})$, these difficulties are aggravating factors.

So what is the alternative? As mentioned above, it is to stay with the probability assignment framework. We are trying to translate both quantitative and qualitative prior knowledge into a probability distribution. Our quantitative knowledge is the estimate $\tilde{F}$, and our qualitative knowledge is that the estimate could be wrong. Just how wrong is the question.

Once again, the original MaxEnt framework is to maximize the entropy $H(p)$ subject to the exact equality constraint (28). This is a good start; we definitely would not want a distribution with any *less* entropy, for that would be assuming *more* than our data (or estimate $\tilde{F}$) warrants. Due to our qualitative acknowledgment of $\tilde{F}$ as an estimate, we perhaps should allow *even more* entropy, but not so much that it would imply untenable error in our estimate.

We must finally introduce quantitative measures so that we may proceed with some computation, but not without an important note on interpretation. The $\boldsymbol{\epsilon}$ in (29) is always an error, so long as there is a true average value of some population. We may also wish to refer to it as a deviation, but it is important to keep in mind that it is the deviation *of* the estimate *from* the true value.

Instead of introducing (and assigning) a probability for the errors, we introduce a convex function $M(\boldsymbol{\epsilon})$ that reflects our judgment of the "potential" for a given error or deviation. With this function in hand, our slightly altered maximum entropy method is

$$\begin{aligned} &\text{Maximize } H(p) - M(\boldsymbol{\epsilon}) \\ &\text{subject to } \bar{F} = \tilde{F} + \boldsymbol{\epsilon}. \end{aligned} \tag{31}$$

If we believe large errors are unlikely, then the potential (somewhat a misnomer) $M(\boldsymbol{\epsilon})$ should grow as the errors increase. This relaxed MaxEnt problem allows a solution with more entropy than the original optimization because it allows the expectations of the assigned probability to deviate from the estimates. These deviations are permitted to increase entropy unless the error potential $M(\boldsymbol{\epsilon})$ grows too large.

As hinted at in the start of this section, the original entropy optimization problem with constraints (28) has an equivalent dual optimization problem. The objective function $H(p)$ is concave, therefore it has a dual objective function that may be found by the Legendre transform [5]. Namely, the dual is maximum likelihood estimate of $\boldsymbol{\lambda}$ from (25) for the sample $\mathcal{D}$ from which the constraints $\tilde{F}$ were derived:

$$\widehat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}, \mathcal{D}).$$

(See Della Pietra, Della Pietra, and Lafferty [14] for a proof of this duality.) Since $-M(\boldsymbol{\epsilon})$ is concave, the new relaxed optimization problem (31) also has a dual:

$$\arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}, \mathcal{D}) - M^*(\boldsymbol{\lambda}),$$

where $M^*(\boldsymbol{\lambda})$ is a dual of $M(\boldsymbol{\epsilon})$. (See e.g., Lebanon and Lafferty [34] for proof.) This optimization corresponds to the MAP estimate of parameters $\boldsymbol{\lambda}$ with a prior

$$p(\boldsymbol{\lambda}) \propto \exp\left(-M^*(\boldsymbol{\lambda})\right).$$

By now the connection to the Bayesian formulation in Section 3.2 should be becoming clear: returning to the derivation for the Gibbs distribution from a maximum entropy framework gives us some ground on which to stand for establishing parameter priors. To the author, the theoretical justification for extrapolating a full prior $p(\boldsymbol{\lambda})$ from a dual of the relaxed maximum entropy problem that can be construed as a MAP estimate seems wanting. Nonetheless, it appears widely accepted. Furthermore, Qi, Szummer, and Minka [47] have demonstrated that such an extrapolation to a full (though approximate) Bayesian calculation does indeed improve results. That being the case, we proceed to review parameter priors.

The most widely used prior on the parameters $\boldsymbol{\lambda}$ is the Gaussian. The convex dual is thus

$$M^*(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^\top \Sigma^{-1} \boldsymbol{\lambda},$$

where Σ is semi-positive definite, the familiar covariance matrix. This primal potential of this M^* is the very similar looking

$$M(\boldsymbol{\epsilon}) = \frac{1}{2}\boldsymbol{\epsilon}^\top \Sigma \boldsymbol{\epsilon}, \tag{32}$$

where the only difference is that the matrix Σ is not inverted (proof of this duality may be found in Lebanon and Lafferty [34]). This potential is quadratic in the errors. When $\Sigma = I\sigma^2$, it corresponds to an ℓ_2 regularizer for the parameters with weight $1/\sigma^2$:

$$M^*(\boldsymbol{\lambda}) = \frac{1}{2\sigma^2}\|\boldsymbol{\lambda}\|_2^2. \tag{33}$$

This is commonly used in logistic regression (called ridge regression) and neural networks (called weight decay). This Gaussian prior has been shown to improve predictions made by relaxed maximum entropy distributions, especially in statistical language modeling and document classification (where it has been called fuzzy maximum entropy) [9].

An alternative regularizer has also been proposed in the logistic regression and neural networks community which is gaining traction in maximum entropy methods: the ℓ_1 regularizer. With a convex dual of

$$M^*(\boldsymbol{\lambda}) = \frac{\alpha}{2} \|\boldsymbol{\lambda}\|_1, \tag{34}$$

this corresponds to a zero-mean Laplacian or double exponential prior on the parameters $\boldsymbol{\lambda}$. The primal potential of (34) has the interesting form

$$M(\boldsymbol{\epsilon}) = \begin{cases} 0 & \|\boldsymbol{\epsilon}\|_\infty \leq \alpha \\ \infty & \text{otherwise} \end{cases}, \tag{35}$$

where $\|\boldsymbol{\epsilon}\|_\infty$ is the maximum absolute value entry of $\boldsymbol{\epsilon}$ (proof of this duality may be found in Riezler and Vasserman [51]). The potential (35) may be interpreted as placing a hard cut-off on the absolute deviation, but with no other penalty. The optimization problem (31) with potential (35) may therefore be equivalently written as

$$\begin{array}{l} \text{Maximize } H(p) \\ \text{subject to } \left\| \bar{F} - \tilde{F} \right\|_\infty \leq \alpha. \end{array} \tag{36}$$

It should be clear that the ℓ_1 and ℓ_2 regularizers incorporate qualitatively different types of prior knowledge. The former allows deviation up to some given amount, while the latter continuously and increasingly penalizes (squared) deviation on some given scale. If one's prior knowledge of the problem is best described by the ℓ_1 penalty, it certainly should be used. However, it is often not clear how much the deviation limit should be. If it is too low, a model will not be sufficiently general. Too high, and constraints will no effect. The discontinuity strikes us as challenging.

That being the case, this discontinuity gives the ℓ_1 regularizer a principle advantage over ℓ_2: feature selection may be incorporated directly into the parameter estimation. This is because if a constraint $\left| \bar{f} - \tilde{f} \right| \leq \alpha$ is met with the corresponding parameter $\lambda = 0$, the feature can be discarded. Eliminating features in this fashion can immensely aid the computational burden in applications where hundreds of thousands of features are used (e.g., text classification) [16].

An issue with both the Gaussian prior (ℓ_2 regularizer) and Laplacian prior (ℓ_1 regularizer) is that they have "nuisance" parameters. That is, we still have not completely escaped the regress of hypotheses and priors. Usually, the σ or α are unknown. One way to handle this is to determine them empirically, by choosing values that seem to give the desired results in some type of validation scheme. An alternative, is to return to the Bayesian framework, treat them as unknowns, and give them prior probabilities.

Our Gaussian prior for the parameters $\boldsymbol{\lambda}$ ought to reflect the fact it is conditioned on a known scale factor: $p(\boldsymbol{\lambda} \mid \sigma)$. A common way to handle this unknown scale factor is to assign it an (improper) non-informative prior density: let $\log \sigma$ be uniformly distributed on the positive real axis.[8] We may therefore recover the prior $p(\boldsymbol{\lambda})$ by marginalizing σ from the joint

$$\begin{aligned} p(\boldsymbol{\lambda}) &= \int p(\boldsymbol{\lambda}, \sigma) \, \mathrm{d}\sigma \\ &= \int p(\boldsymbol{\lambda} \mid \sigma) \, p(\sigma) \, \mathrm{d}\sigma \\ &\propto \|\boldsymbol{\lambda}\|_2^{-K} \end{aligned}$$

[8] The logarithm is to have uniform density because σ represents a scale. This is akin to having "insufficient reason" to prefer inches, feet, yards, or miles, as an acceptable scale of error rather than one inch, or two inches, or three inches, etc.

where $p(\boldsymbol{\lambda} \mid \sigma)$ is Gaussian, $p(\sigma) = 1/\sigma$, and K is the dimensionality of $\boldsymbol{\lambda}$. This result is attributed to Buntine and Weigend [6], who applied it to neural networks. With this "scale-free" prior the term in the log-posterior used for parameter optimization is then

$$\log p(\boldsymbol{\lambda}) = -K \log \|\boldsymbol{\lambda}\|_2 + C \tag{37}$$

where C is an irrelevant constant term corresponding to the normalizing factor. The same strategy may be employed to handle the α of the Laplacian prior $p(\boldsymbol{\lambda} \mid \alpha)$, which similarly yields

$$\log p(\boldsymbol{\lambda}) = -K \log \|\boldsymbol{\lambda}\|_1 + C. \tag{38}$$

This regularizer was used by Williams [62], who argued for the propriety of the Laplacian regularizer on neural networks based on the particular form of prior knowledge it reflects. Unfortunately, the functions (37) and (38) are not convex, so optimization methods such as gradient ascent only find local optima. Practical issues connected with this are discussed with our experimental setup and analysis.

Rather than marginalizing nuisance parameters and then finding a MAP estimate, MacKay instead proposes finding the scale σ or α that maximizes the "evidence" for the data, e.g., $p(\mathcal{D} \mid \sigma)$. [38]. It is important to remember that the MAP estimate of parameters is an approximate method for prediction from the Bayesian standpoint of (15). However, the evidence framework seems to introduce an approximation where the identical Bayesian calculation could be computed exactly. MacKay argues that the evidence framework is a better representation of the true posterior volume than the MAP estimate with σ or α marginalized—a sentiment similar to those we expressed for the labeling strategy in Section 3.5. Unfortunately, the evidence framework requires the Hessian of the model likelihood. Since extensive approximations are already required for inference and more than ten thousand (in other applications, hundreds of thousands) features are used, calculating or even storing the Hessian makes the evidence framework non-ideal for CRF application.

In the experiments section, the Gaussian regularizer (33) and its scale-free counterpart (37) version are used.

Heretofore in this section, we have been describing the parameter vector of the Gibbs maximum entropy distribution (25) as a single unit $\boldsymbol{\lambda}$. However, in our grid-shaped CRF model we have a partition of the parameters $\boldsymbol{\theta} = \langle \boldsymbol{\lambda}, \boldsymbol{\mu}_h, \boldsymbol{\mu}_v \rangle$. The edge parameters $\boldsymbol{\mu}$ are somewhat more prone to overfitting than the node parameters $\boldsymbol{\lambda}$ because the possible volume of patch pairs relative to the actual amount of training data is much larger than that of the patches alone. In his work on neural networks, Williams discusses different classes of weights that call for different forms of regularization [62]. Following this lead, we place the parameter vectors $\boldsymbol{\lambda}$, $\boldsymbol{\mu}_h$, and $\boldsymbol{\mu}_v$ in different "classes." Thus, each may have their own regularization scale. Under a Gaussian prior, for instance, this translates to independence among the parameters

$$p(\boldsymbol{\lambda}, \boldsymbol{\mu}_h, \boldsymbol{\mu}_v \mid \sigma, \sigma_h, \sigma_v) = p(\boldsymbol{\lambda} \mid \sigma)\, p(\boldsymbol{\mu}_h \mid \sigma_h)\, p(\boldsymbol{\mu}_v \mid \sigma_v).$$

More importantly, for the scale-free prior (37), we allow each vector to have its own adaptive scale

$$\log p(\boldsymbol{\lambda}, \boldsymbol{\mu}_h, \boldsymbol{\mu}_v) = -K\,|\mathcal{Y}| \log \|\boldsymbol{\lambda}\|_2 - K\,|\mathcal{Y}|^2 \log \|\boldsymbol{\mu}_h\|_2 - K\,|\mathcal{Y}|^2 \log \|\boldsymbol{\mu}_v\|_2 + C.$$

The details of regularization as they concern our experiments are described in 5.2.3.

4 Image Features for Sign Detection

In this section we describe the general purpose texture features used to detect signs from among other areas of natural images. The features are statistics of outputs from image filters motivated by biological vision systems.

4.1 Scale and Orientation Selective Simple and Complex Cells

Studies of mammalian vision systems indicate the presence of so-called simple cells, which respond selectively to stimuli at a particular phase, scale, and orientation in a receptive field area [12]. Although other models exist, two-dimensional Gabor filters are often used as a simple computational model for this biological phenomenon. Information from simple cell quadrature phase pairs is captured by a unit called the complex cell. The output from image filters designed to model these and other cells mimic the neurological responses of the cells and act as input to basic vision systems, where higher level processing maps the responses to interpretations, such as texture recognition.

A standard framework for texture *synthesis* is to measure some moments of features (filter responses) from a texture example and then sample from a distribution that matches those moments (see e.g., Zhu [66]). However, it has been shown that statistics of outputs from individual filters are insufficient for generating complex textures [46]. This is because responses of different filters (i.e., of different scales and/or orientations) may exhibit correlations on some texture. Therefore, the problem arises from the fact that distinct textures with the same marginal statistics (e.g., auto-correlation) can have different joint statistics (e.g., cross-correlation). Capturing these joint relationships is important for accurately representing, generating, and discriminating textures. For that reason, we use the inter-filter statistics of Portilla and Simoncelli [46], which are based on scale-selective, directionally oriented filters (loosely inspired by simple cells) and include:

Marginal Statistics: Range (minimum, maximum), normalized sample moments (variance, skewness, kurtosis) of gray values in the low-pass image at each scale.

Auto-Correlation: Center auto-correlation values of both the low-pass image at each scale and the complex magnitudes of wavelet coefficients (filter responses) at each scale and orientation.

Cross-Correlation: Center cross-correlation values of wavelet coefficient complex magnitudes for each scale and orientation with all orientations of the neighboring coarser parent scale and all orientations of the same scale.

Cross-Scale Phase: Center cross-correlation values of wavelet coefficient real parts for each scale and orientation with both real and imaginary parts of phase-doubled coefficients of all orientations at the neighboring coarser parent scale.

The total number of statistics grows quadratically in both the number of scales and orientations used. We employ just four scales and four orientations to limit the number of model parameters that eventually need to be learned; Portilla and Simoncelli [46] generate impressive textures using the same number of scales and orientations. These filter statistics are a good basis for describing the myriad of textures our system will be expected to encounter in natural images.

It is worth briefly noting a difference between the oft-used Gabor filterbank design of Manjunath [39] and the steerable pyramid basis filters [54], both shown in Figure 3. While both are complete but non-orthogonal (also called over-complete), the design of the Gabor filterbank leaves relatively large areas of weak support in the frequency domain. Thus, although biologically inspired, this Gabor filterbank design strategy is unsuitable for small numbers of orientations and/or scales. Like the Gabor filter, the steerable pyramid basis functions are polar separable in the frequency domain (a requirement for simple cell models [12]). Since we wish to keep our parameter count low but also adequately cover frequencies, we use the steerable pyramid for these simple and complex cell-based statistics.

4.2 Grating Cells

The existence of periodic and aperiodic pattern selective cells in the visual cortex of monkeys has been discovered relatively recently. These so-called grating cells use the simple cells described above as input and act in a highly non-linear fashion to discriminate between a grating of several bars and a single bar. The important difference between the simple or complex cells and these newly discovered

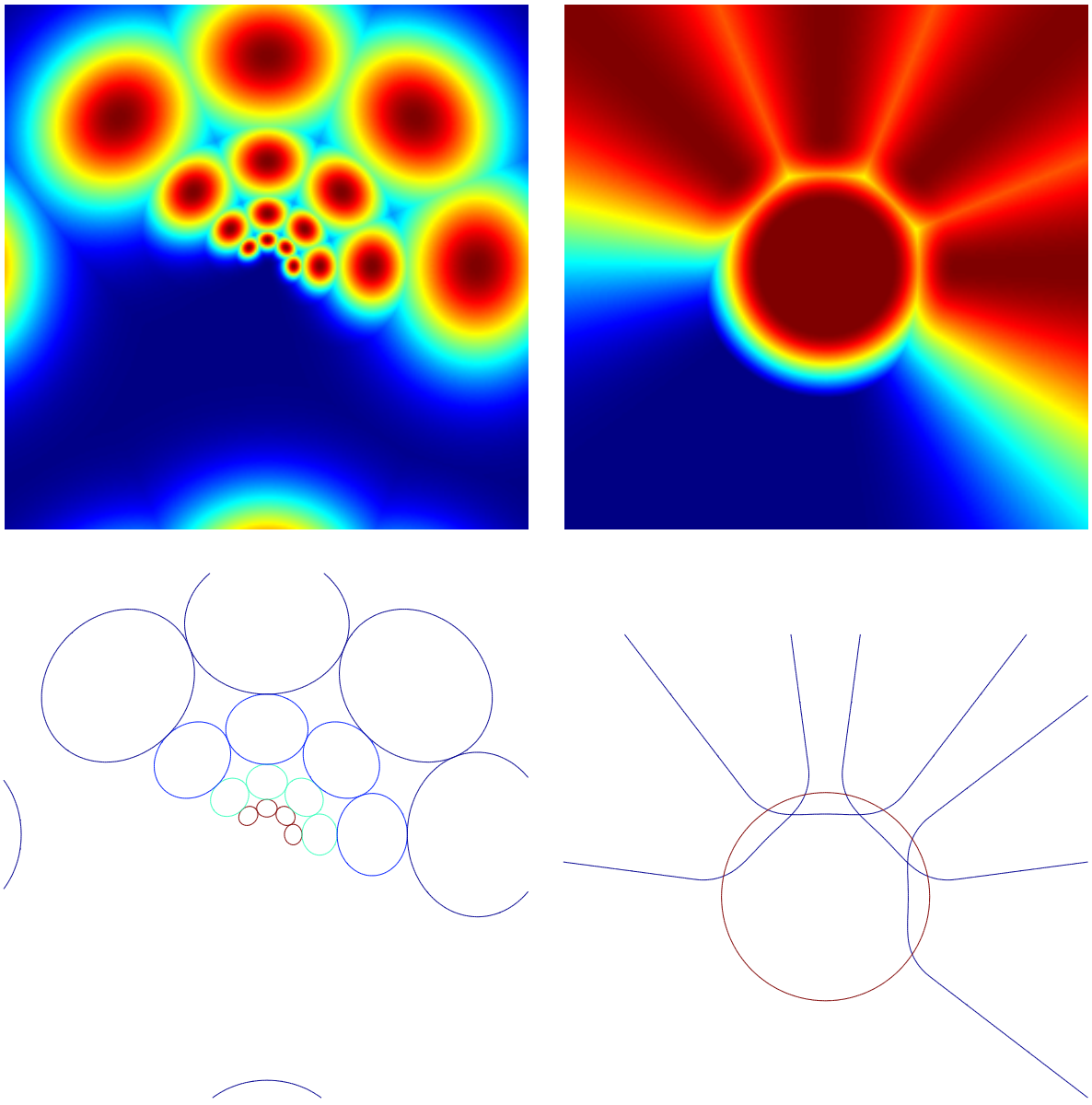


Figure 3: Filter envelope comparison. LEFT: Gabor filterbank constructed so that neighboring filters touch at half-peak magnitude. RIGHT: Steerable pyramid wavelet basis filters. TOP: Superposed filter maximum Fourier energies. BOTTOM: Half-peak magnitude contours.

cells, is that the former respond to both gratings and bars, while the latter respond selectively to gratings and not to bars.

Periodic grating cells respond strongly to a series of at least three bars at a particular orientation and periodicity (or scale). The computational model of grating cells proposed by Petkov and Kruizinga [44, 30] is used as the basis for our text detecting operator.

While real grating cells respond at very low (1%) contrast, and the model of [44] is normalized for contrast, we expect text in signs to have relatively high contrast, so we forego such normalization. Let $I_{\theta,\omega,\phi}$ be the response of an input image to a simple cell with preferred orientation angle θ, spatial frequency ω, and phase ϕ.[9] Since we will only be interested in finding areas of strong "center-on" (a bright peak, $\phi = \pi$) or "center-off" (a dark valley, $\phi = 0$) responses, we clip the lower limit of the range of $I_{\theta,\omega,\phi}$ to zero and normalize so that the maximal response is one.

For our text-detection application, we are not interested in arbitrarily long gratings. Rather, letters have a limited aspect ratio, thus the "bars" in text have bounded height. In accordance with this observation, we alter slightly the model of [44] by sending the original simple filter response $I_{\theta,\omega,\phi}$ through a second round of simple filtering with output $T_{\theta,\omega,\phi}$, where θ, ω, ϕ still indicates the parameters of the primary simple filter. However, this secondary simple filter has an orthogonal orientation $\theta + \frac{\pi}{2}$, a center-on phase of π, and a frequency of no more than $\omega/2$. These combine to elicit stronger responses from bars of limited height. The range of $T_{\theta,\omega,\phi}$ is similarly changed to $[0, 1]$ by clipping the bottom and normalizing the top. Finally, the original simple filter response is weighted by the perpendicular response,

$$F_{\theta,\omega,\phi}(x, y) \triangleq I_{\theta,\omega,\phi}(x, y)\, T_{\theta,\omega,\phi}(x, y).$$

Figure 6 shows the primary , orthogonal, and weighted simple filter responses of an example image.

Once the weighted simple units are calculated, a binary grating cell subunit $Q_{\theta,\omega}$ indicates the presence of a grating at each image location. To make such a determination, we consider a line with orientation θ centered at each location (x, y). The line is divided into intervals of length $1/2\omega$, given by

$$\mathcal{R}_{\theta,\omega,n}(x, y) \triangleq \left\{ (x', y') : \frac{n}{2\omega}\cos\theta \le (x - x') < \frac{n+1}{2\omega}\cos\theta \wedge \frac{n}{2\omega}\sin\theta \le (y - y') < \frac{n+1}{2\omega}\sin\theta \right\}.$$

The minimal criterion for a grating to be present is three bars, so we consider only six intervals, $n = -3, \ldots, 2$, along the line. The presence of such a grating would be indicated by alternating strong center-on ($\phi = \pi$) and center-off ($\phi = 0$) responses along the line. First we find the maximum such responses within in these regions

$$\begin{aligned} M_{\theta,\omega,n}(x, y) &\triangleq \max\left\{ F_{\theta,\omega,\phi_n}(x', y') : (x', y') \in \mathcal{R}_{\theta,\omega,n}(x, y) \right\}, \\ \phi_n &\triangleq \begin{cases} 0 & n \bmod 2 = 1 \\ \pi & n \bmod 2 = 0, \end{cases} \end{aligned}$$

as well as the overall maximum

$$\hat{M}_{\theta,\omega}(x, y) \triangleq \max_{n=-3,\ldots,2} M_{\theta,\omega,n}(x, y).$$

Finally, the binary grating subunit indicator is given by

$$Q_{\theta,\omega}(x, y) \triangleq \begin{cases} 1 & \text{if } M_{\theta,\omega,n}(x, y) \ge \rho \hat{M}_{\theta,\omega}(x, y),\ n = -3, \ldots, 2 \\ 0 & \text{otherwise,} \end{cases}$$

where $0 < \rho < 1$ is a threshold that indicates the minimum relative strength of simple cell maximal responses in the line-segment receptive fields. For example, if $\rho = 0.9$, then all the receptive field maxima must be least 90% of the strongest response. These receptive field regional maximums as well as the AND-type behavior of the grating subunit make this operator very non-linear.

[9] Note that this θ is an angle, and is not the same as the parameter vector $\boldsymbol{\theta}$. The two should be distinguishable from context.

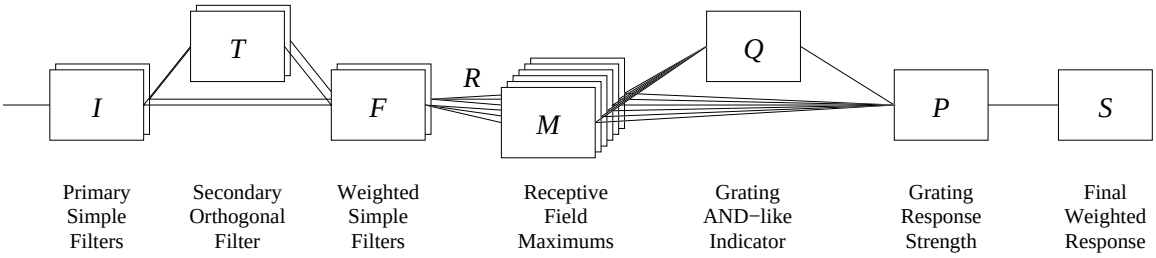


Figure 4: Grating cell data flow for a single scale and orientation. Two boxes at I and T are for the center-on and center-off filters, while the boxes at M are for the six receptive fields.

Now that we have an indicator for the presence of a grating, it remains to quantify the "strength" of the grating. We use the max of the receptive field maximums wherever a grating is present and zero elsewhere:

$$P_{\theta,\omega}(x,y) \triangleq \begin{cases} \hat{M}_{\theta,\omega}(x,y) & \text{if } Q_{\theta,\omega}(x,y) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Alternately, the mean of the receptive field maximums

$$\frac{1}{6}\sum_{n=-3}^{2} M_{\theta,\omega,n}(x,y)$$

could be used in place of the max $\hat{M}_{\theta,\omega}(x,y)$ if the overall strength is more important. Since letters are rarely perfectly aligned bars, we use the maximum response (rather than the mean) when a grating is indicated. The final response of the grating cell is a local weighted average of the grating strengths $P_{\theta,\omega}$. As Kruizinga and Petkov explain, this "is a provision made to model the spatial summation properties of [biological] grating cells with respect to the number of bars and their length as well as their unmodulated responses with respect to the exact position (phase) of a grating [30, p. 1399]." We accomplish this by convolving the output with a symmetric Gaussian kernel

$$S_{\theta,\omega} = P_{\theta,\omega} * G_{\omega}$$

where the standard deviation of the Gaussian G_{ω} is five times that of the Gaussian that attenuates the simple filter with frequency ω.

The flow of data from image to grating response is represented in Figure 4. Differences between our model and the original of [44] are (1) the elimination of contrast normalization, (2) the addition of an orthogonal secondary filtering stage, and (3) the dependency of the final value on the receptive field maximum rather than the indicator subunit exclusively. An example is shown in Figure 7. Note how the nonlinearity is required to remove remaining spurious non-grating responses (compare the bottom image of Figure 6).

4.3 Color

We have also added some simple color features to our model. While color may not be intrinsically useful for detecting signs, due to their wide variety, it has two other advantages. First, color can be a good indicator for non-sign regions. If the training data contains significant amounts of blue sky, red bricks, green foliage, etc., the presence of such colors can reduce the evidence for the presence of a sign at a site. The second advantage stems from the fact that in the CRF model, we can use relationships between data at neighboring sites. Therefore, the use of color continuity might be expected to increase accuracy.

The image is converted into the HSV (hue, saturation, and value) color space, where each channel has a range of $[0, 1]$, hue being circular. The distribution of saturation in a region is measured by a normalized histogram. Saturation can vary wildly under different lighting conditions, so it is

Figure 5: Grating operator on text. Upper Left: Input image. Upper Right: Center-on and center-off simple filter responses ($\theta = 0$). Bottom: Slice of simple filter responses (center-on in solid and center-off in dashed) with receptive regions for a marked point.

advisable to use only a few bins. As saturation decreases, the corresponding hue becomes less stable. Therefore, rather than measuring the distribution of hue with a simple histogram, the contribution of a pixel's hue is linearly weighted by its saturation prior to normalization [52]. We do not bin the value elements because brightness is captured by the features described in Section 4.1.

5 Experiments

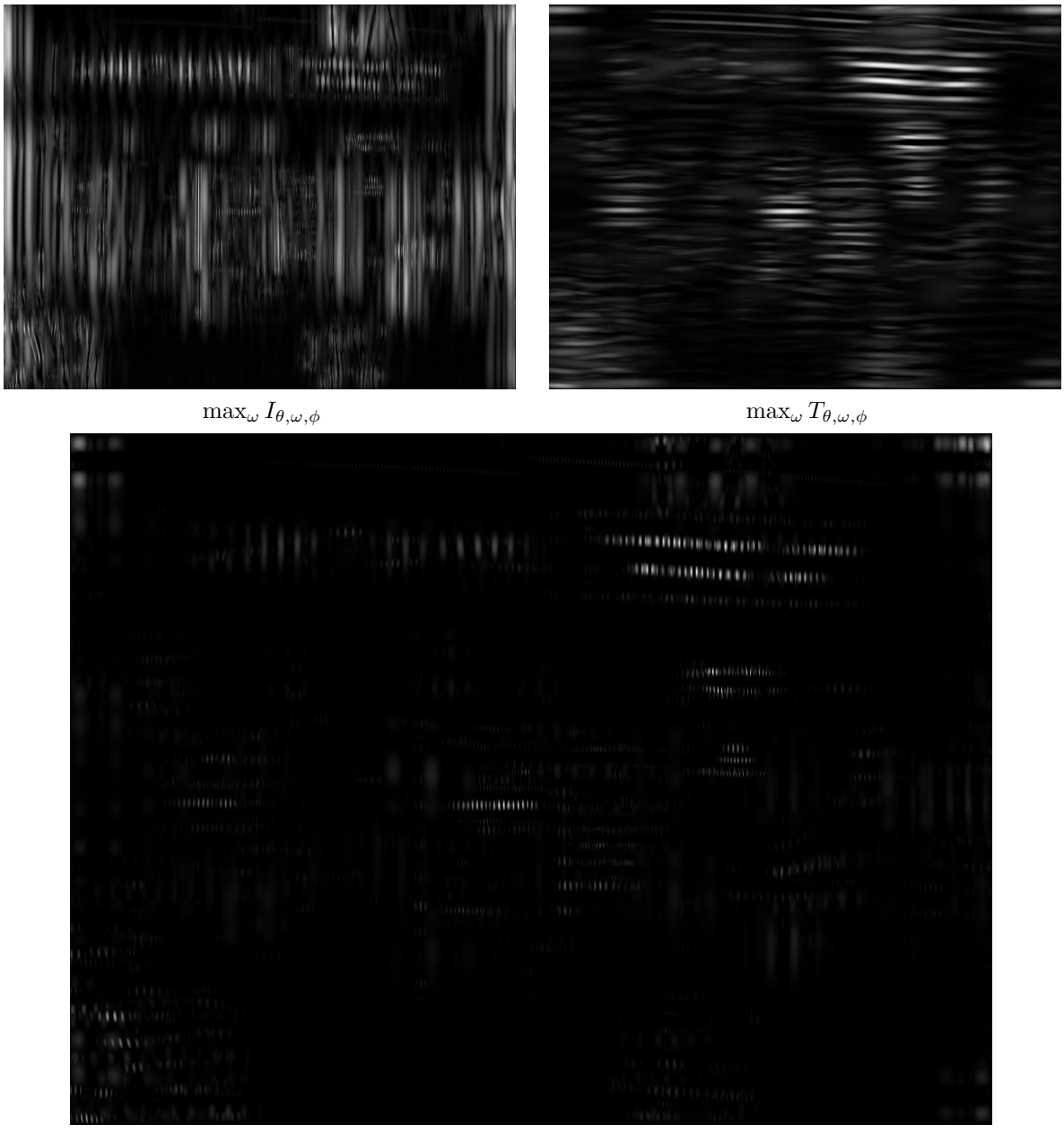
E.T. Jaynes [28, p. 371]

In this section we have two primary aims. First, to evaluate the usefulness of the image features outlined in Section 4 on the problem of sign detection. Second, to demonstrate improvement in accuracy by using label and data context that CRFs provide.

5.1 Data

Images of downtown Amherst, MA were collected with a Nikon Coolpix 995 during September 2003. A total of 309 24-bit color images of size 2048x1536 were taken during various times of day throughout the month.[10] Signs in the image were then labeled (masked) by hand. The general criteria for marking a sign were as follows:

[10] Available from `<http://vis-www.cs.umass.edu/projects/vidi>`.



$\max_\omega I_{\theta,\omega,\phi}$ $\max_\omega T_{\theta,\omega,\phi}$

$\max_\omega F_{\theta,\omega,\phi}$

Figure 6: Filter responses on a natural image ($\theta = 0$, $\phi = \pi$). Top Left: Primary simple filter responses over several scales. Top Right: Secondary orthogonal responses over several scales. Bottom: Primary and secondary orthogonal filter response products over several scales.

Figure 7: Text detection with grating cells (cf. Figure 1). Top: Grating cell responses over several scales. Bottom: Contours of responses overlaid on the image.

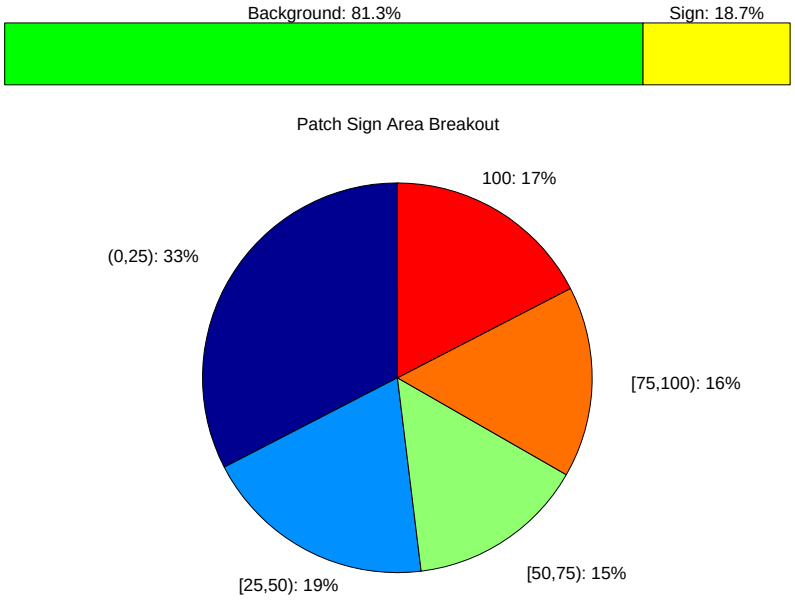


Figure 8: Breakdown of data patches by class. Patches labeled Sign contain varying amounts of true sign because they straddle a mask boundary (see text).

- it should be readable (if text) or recognizable (if a logo) at 25% magnification (512x384)
- if a sign's natural border is reasonably close to the text or logo, extend the mask to the border, otherwise the mask should extend to roughly 128-256 pixels in any direction beyond the text or logo.

It should be noted that these were not hard and fast rules, but heuristic guidelines. There could be small signs present in an image that are not labeled as such because they can not be read legibly, but could still have their text detected. The second guideline allows color continuity and presumed edge evidence for signs to be used to a reasonable extent. As will be demonstrated in the results, the hand-masked regions should not be considered absolute ground truth, because there is flexibility in the location of boundaries. Figure 1 shows an example of hand-masked sign regions.

For our experiments, we scaled the images by half to 1024×768 and measured the features described in Section 4 over disjoint 64×64 regions. This yields a 16×12 grid CRF. Whereas the masks described above have roughly pixel precision, these regions, which we call image patches, might be composed of all sign, no sign, or something in between. How the regions are labeled for training and testing impacts the results. Throughout the section, we cluster patches by the amount of area covered by sign into the percentage ranges $(0, 25)$, $[25, 50)$, $[50, 75)$, $[75, 100)$, and 100. As shown in the top of Figure 8, the number of uninteresting background patches exceeds the patches of interest by a ratio of roughly 4:1. We will be most interested in detecting the patches that are mostly sign, and these are outnumbered by background patches more than 8:1.

In preliminary work, we treated the task as a binary classification problem [59]. Specifically, if a patch contained area at least 50% hand-masked as sign, the training and evaluation label given was Sign, otherwise the label was Background. The mixed sign/non-sign regions are highly troublesome for a classifier. For instance, a patch containing 49% sign area is labeled non-sign while another patch, perhaps even a neighbor, containing 51% sign area is labeled sign. The features for these regions are extremely similar and if the classifier tries to discriminate between them it may come at the cost of more effectively discriminating pure sign from background patches. Even though a full one-third of the Sign patches contain less than 25% sign, we attempt to detect them. We prefer

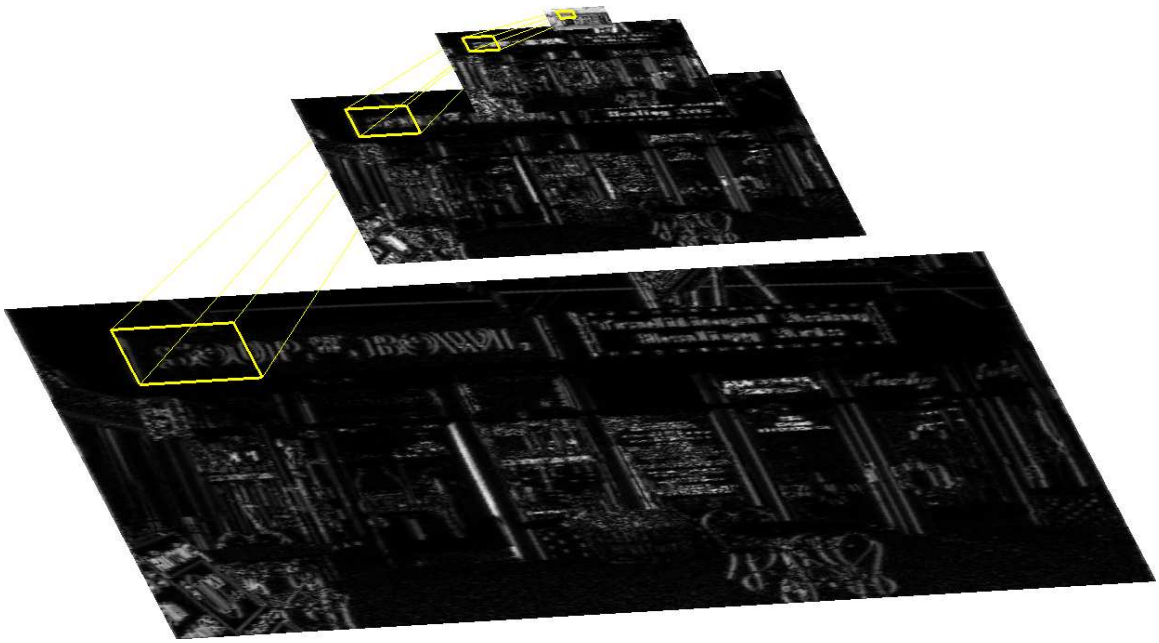


Figure 9: Relevant areas of an image pyramid for calculating features. The complex magnitudes of three scales at one orientation plus the low-pass image are shown. Statistics of the same area at different scales are calculated . The example area shown is enlarged for illustrative purposes.

there to be confusion where it matters less: between patches containing no sign and those with just a little, rather than between patches that are slightly less and slightly more than half sign.

We randomly split the images into train and test sets (155 and 154 images, respectively) for our evaluation and used same splits throughout all experiments.

5.2 Features and Parameters

5.2.1 Images

As mentioned in §4.1, we use a pyramid of four scales and four orientations. Taking only the center five correlation values and accounting for symmetry, these statistics give 539 features; let these features be $\mathbf{f}_p \in \mathbb{R}^{539}$.[11] Figure 9 illustrates the regions used for calculating features of the image corresponding to the grid regions.

Our color histograms contain 8 hue bins and 10 saturation bins, for a feature vector $\mathbf{f}_c = \langle \mathbf{f}_h, \mathbf{f}_s \rangle \in \mathbb{R}^{18}$.

For the grating cells, we use an 8 scale, 8 orientation Gabor filterbank as described in [39], with centers spaced at roughly 1.5 octaves between $\omega = 0.25$ and $\omega = 0.4$ cycles per pixel. To enhance the detection rate, we use $\rho = 0.5$. This weak requirement for grating presence is necessary since text is not a perfect grating (for example, note the relatively weak left-most response of Figure 5). Text in our dataset nearly always appears with vertical strokes, so we use only the orientation $\theta = 0$. We take the min, max, mean, and variance, of grating responses $S_{\theta,\omega}$ at each scale in each region, giving a feature vector $\mathbf{f}_g \in \mathbb{R}^{32}$.

5.2.2 CRF

The observation features f^k for nodes are simply formed from the entries of the concatenated vector of features described in the previous section, $\mathbf{f} = \langle \mathbf{f}_p, \mathbf{f}_c, \mathbf{f}_g \rangle$. In addition, a class bias term $f_y^K = 1$ is

[11] MATLAB code for calculating the features can be obtained from <http://www.cns.nyu.edu/~lcv/texture>.

used when the full feature function vector F is formed.

We use a few types of observation features for edges. Whereas node feature functions f_y^k are specific to a single label, edge feature functions $g_{y,y'}^j$ are specific to a pair of labels. Neighboring class labels add helpful contextual information to even features of a single site, so we use $\mathbf{g}_n = \langle \mathbf{f}, \mathbf{f}' \rangle$ where $\mathbf{f}$ and $\mathbf{f}'$ are the feature vectors of the two nodes attached to the edge. To capture any difference or similarity between individual features of neighboring patches, we measure the squared difference between each component of the feature vectors, denoted by $\mathbf{g}_d = \text{diag}\left((\mathbf{f} - \mathbf{f}')(\mathbf{f} - \mathbf{f}')^\top\right)$. To summarize overall (dis)continuity of the filter statistics, hue distribution, and saturation distribution, we use the l norms of their differences, $\mathbf{g}_l = \left\langle \left\|\mathbf{f}_p - \mathbf{f}'_p\right\|_2, \left\|\mathbf{f}_c - \mathbf{f}'_c\right\|_2, \left\|\mathbf{f}_h - \mathbf{f}'_h\right\|_2, \left\|\mathbf{f}_s - \mathbf{f}'_s\right\|_2 \right\rangle$. The observation features g^j for edges are simply formed from the entries of the concatenated vector of features $\mathbf{g} = \langle \mathbf{g}_n, \mathbf{g}_d, \mathbf{g}_l \rangle$. In addition, a class-pair bias term $g_{y,y'}^J = 1$ is used when the full feature function vector G is formed.

With $\mathcal{Y} = \{\mathsf{Sign}, \mathsf{Background}\}$, these features result in a total parameter vector $\boldsymbol{\theta}$ of size 15,356. To evaluate the utility of the contextual information, we also test the "independent" model where $E_s = \emptyset$.

5.2.3 Regularization

In initial experiments we found no dramatic difference between the Gaussian and Laplacian priors, or their respective scale-free versions. Therefore, we only present results for unregularized models, those trained under a Gaussian prior with the scale determined by validation, and the scale-free Gaussian.

The validation method was as follows. The training data (50% overall) was evenly split, with half (25% overall) for training and the remainder for evaluation with different values of the parameter. The parameter value that yielded the largest area under the ROC curve for the MPM estimator was chosen and training then proceeded on the full training set. We chose this criterion because the area under the ROC curve is a good single-valued measure of the detection/false alarm trade-off; MPM is the only estimator with an obvious ranking function (marginal posterior probability), which the ROC curve requires.

The scale-free Gaussian is infinite at $\boldsymbol{\theta} = \mathbf{0}$. Aside from the fact that it is the obvious global maximum of the log posterior (19), the gradient there is undefined. We therefore initialize the parameters at their maximum-likelihood (unregularized) values before proceeding with gradient ascent. Although the maximum is indeed at $\boldsymbol{\theta} = \mathbf{0}$, the ascent algorithm terminates when either the change in value or largest gradient become sufficiently small: the gradient relative to an absolute threshold, and the value relative to previous values before a step. As we describe in our analysis, a subset of the parameters tends to move quickly to the zero-vector, pushing the log-posterior toward infinity. When the value gets high, the relative change is small enough to cause the gradient ascent to terminate before *all* the parameter vectors become $\mathbf{0}$. Since the scale-free priors are non-convex, gradient ascent finds local optima. If the algorithm terminates without a vector norm being driven to zero, it will be in a region of parameter space where the likelihood is more strongly peaked than the prior.

5.3 Results

Here we present the results of our experiments testing combinations of three factors:

Training: Independent, Pseudo-Likelihood approximation (PL), and Tree Reparameterization (TRP) approximation (cf. Sections 3.1, 3.3, and 3.4).

Regularization: No prior, the scale-free Gaussian prior, and a Gaussian prior with the variance determined by validation (cf. Section 3.6).

Prediction: Iterated Conditional Modes (ICM), Maximum a Posteriori (MAP), and Maximum Posterior Marginal (MPM) (cf. Section 3.5).

In practice, we found that ICM only converged about 30% of the time, but it typically did so in about 2-7 iterations. TRP almost always converged during training and prediction, only failing occasionally (with a 400 iteration limit) when the parameters were estimated with no prior or the scale-free Gaussian.

Figures 10, 11, and 12 show the ROC curves for the posterior marginals of patches with the three priors (none, scale-free Gaussian, and Gaussian, respectively) and all training methods. These show the relative power of the classifiers to discriminate between Background patches and others containing varying amounts of sign. See the Appendix for areas under the curves.

Figures 13, 14, and 15 give the detection and false alarm rates for all training and prediction methods with each prior. The overall detection rate of all Sign patches is given as well as the detection rates for patches containing various amounts of sign. There is only one false alarm rate for a given classification since there is but one negative class, Background.

Whereas the previous results report patch level statistics, Figure 16 gives sign level stats for the TRP-trained CRF with a Gaussian prior using ICM prediction. This provides the rate of detection for entire signs, and, when part of a sign is found, what fraction of the patches constituting the sign are detected.

Figures 17, 18, and 19 show example detections using the CRF trained by TRP with a Gaussian prior using ICM for prediction. The green patches are correct detections, red patches are false positives, and cyan contours indicate the hand-drawn masks. Figure 20 shows some conspicuous signs gone undetected by our model, and Figure 21 illustrates some marginal posterior probabilities of one image. Finally, Figure 22 highlights some qualitative differences between predictions made by ICM and MPM.

6 Discussion

Here we present our analysis of the results from Section 5.3 and discuss previous related work.

6.1 Analysis

We first explain the most peculiar aspect of the results, which has to do with pseudo-likelihood training. Recall that PL maximizes the conditional likelihood of all patches *given* the neighboring labels. Due to the extreme value of label context for the problem—being surrounded on all sides by Sign virtually guarantees a patch is Sign—this is nearly all the information that is needed for classification. Unfortunately, come prediction time the resulting classifier will have put too much weight on features that are not as reliable as they appeared during training. Simply put, PL maximizes the wrong (though easy to work with) objective function. The fact that unreliable edge features are given drastically great sway in decision making means virtually all posterior marginals are zero or one. This is why the ROC curves in Figures 10 and 11 are so abnormal; many of the patches with $p\left(y = \mathsf{Sign} \mid \mathbf{x}\right) = 1$ are actually Background. Thus, before the parameter $t = p\left(y = \mathsf{Sign} \mid \mathbf{x}\right)$ can even be moved, a radically large false alarm rate is incurred. By way of comparison, after TRP training with no prior, the horizontal edge parameters have an ℓ_2 norm of $\left\|\boldsymbol{\mu}_h\right\|_2 = O\left(1e-1\right)$, while after PL training with no prior the same norm is $\left\|\boldsymbol{\mu}_h\right\|_2 = O\left(1e3\right)$. This explains the irregularity in the corresponding ROC curves and the need for strong regularization to counter-balance the overconfidence of pseudo-likelihood. We postpone discussion of properly regularized PL results until later.

The edgeless independent classifier has many fewer parameters than the full grid-equipped CRF. Although TRP is a better approximation to the full-likelihood than pseudo-likelihood, we expect the greater complexity of the CRF to require some regularization for it to achieve optimal performance. Comparing the results of the two classifiers trained with no prior only reveals relative amounts of overfitting. The TRP-trained CRF mildly overfits with lower ROC areas overall and on patches containing mostly sign, but even with the overfitting it does do slightly better than the independent classifier on those cases where context is most helpful, namely patches containing very little sign but neighboring other more obvious sign patches. However, with ICM prediction, the TRP-CRF

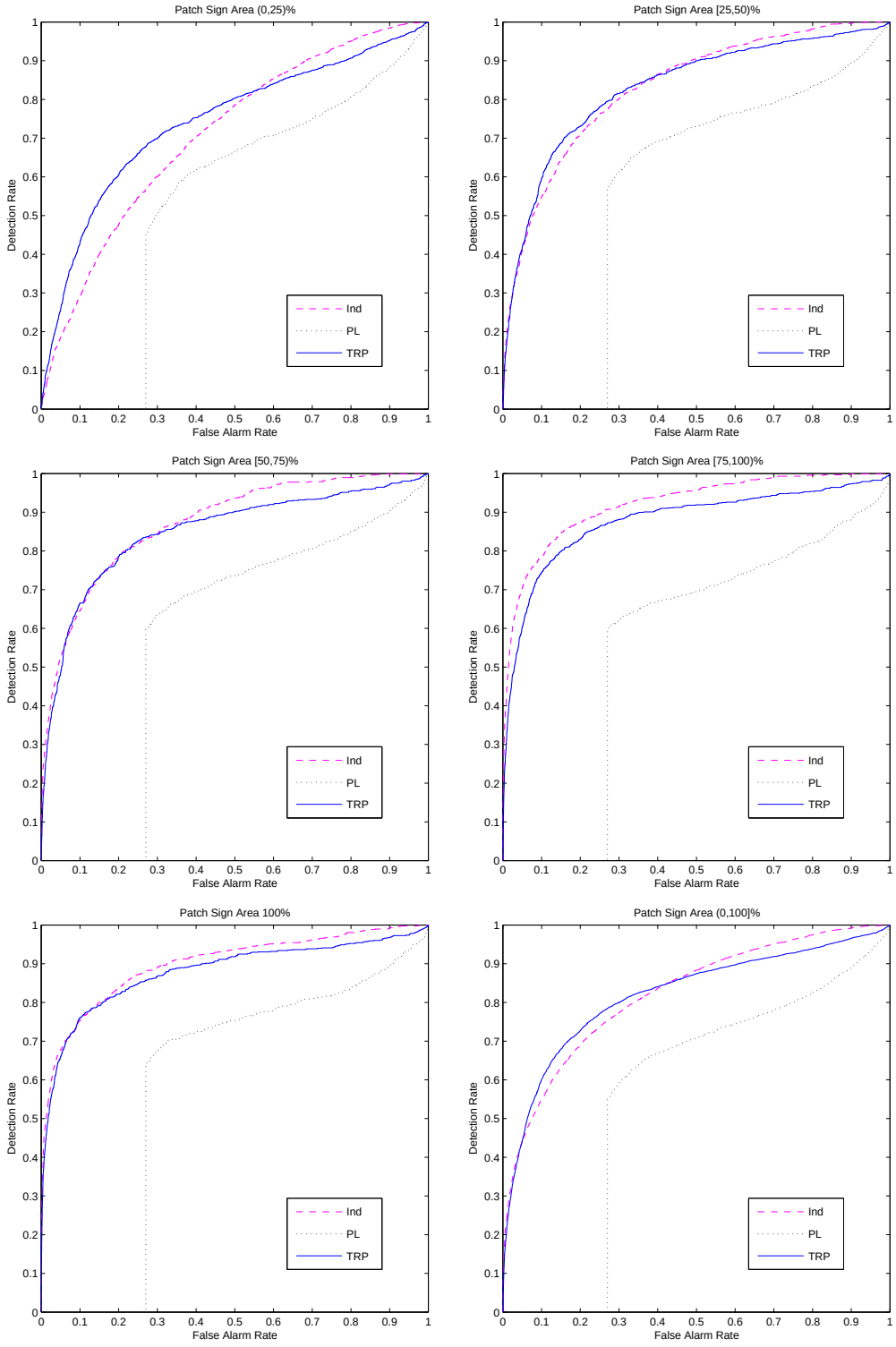


Figure 10: ROC curves for the posterior marginals with no prior. Results are broken down by training method and the amount of sign (percentage of total patch area) contained in the positive patches.

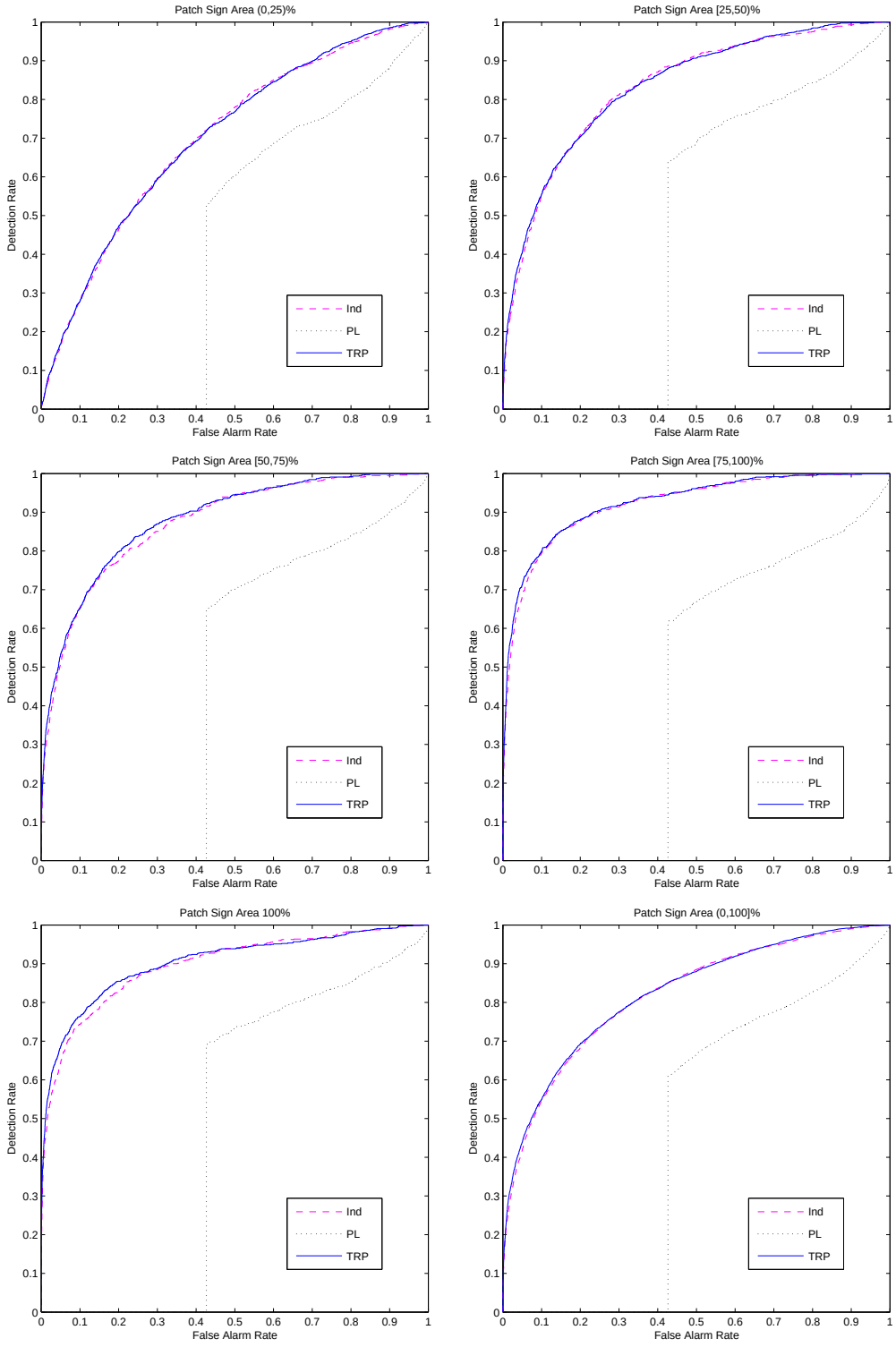


Figure 11: ROC curves for the posterior marginals with a scale-free Gaussian prior. Results are broken down by training method and the amount of sign (percentage of total patch area) contained in the positive patches.

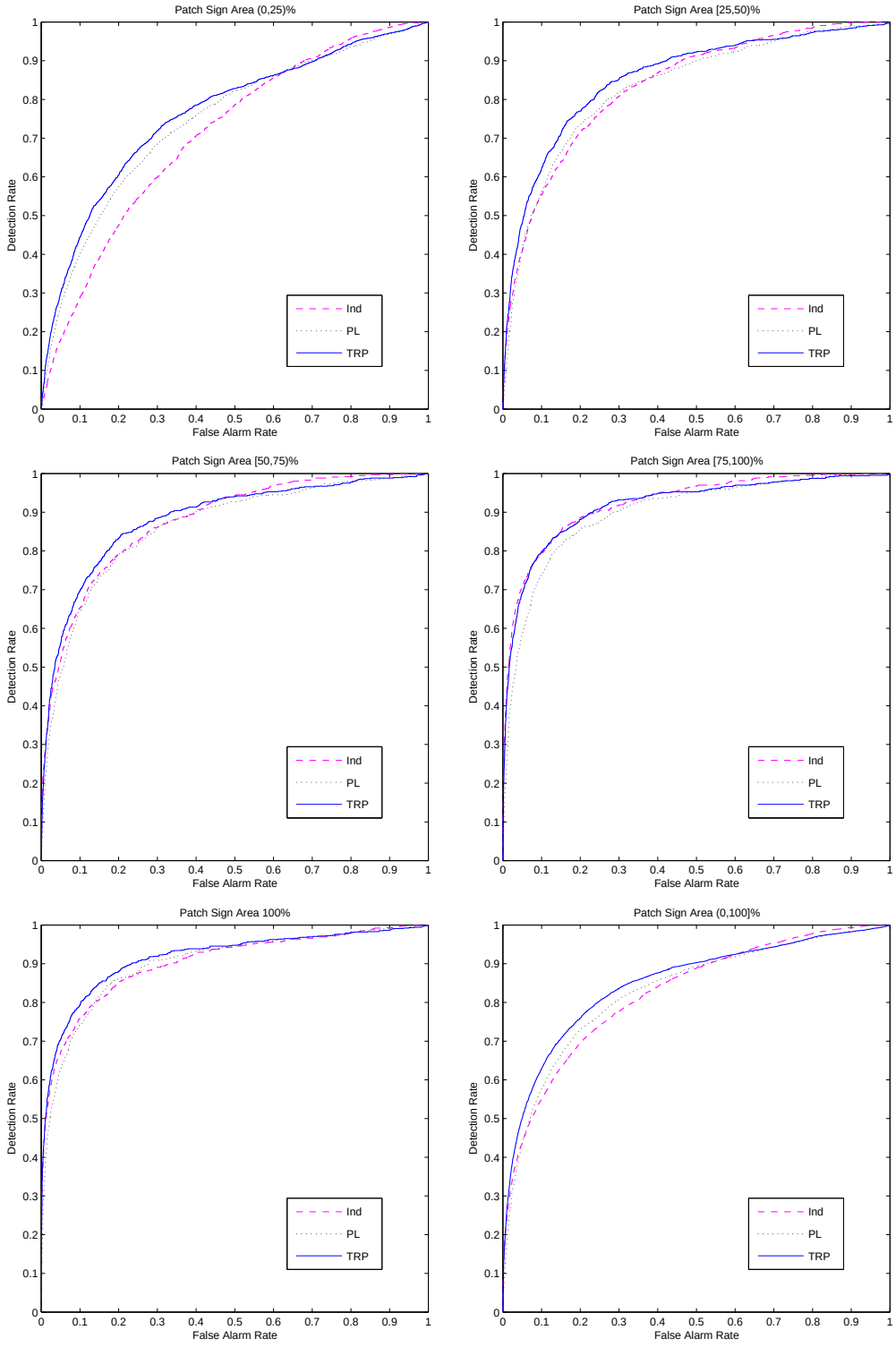


Figure 12: ROC curves for the posterior marginals with a Gaussian prior. Results are broken down by training method and the amount of sign (percentage of total patch area) contained in the positive patches.

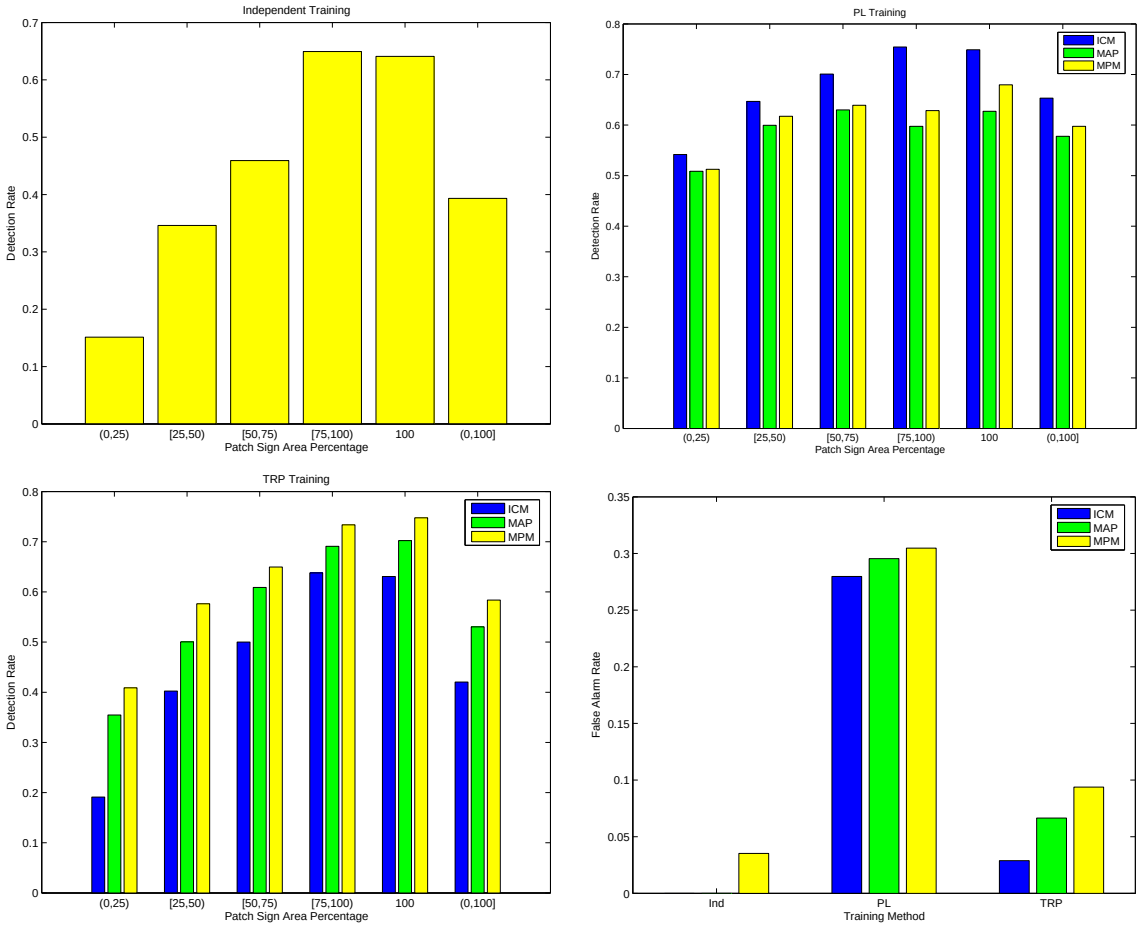


Figure 13: Detection and false alarm rates for various training and prediction methods with no prior.

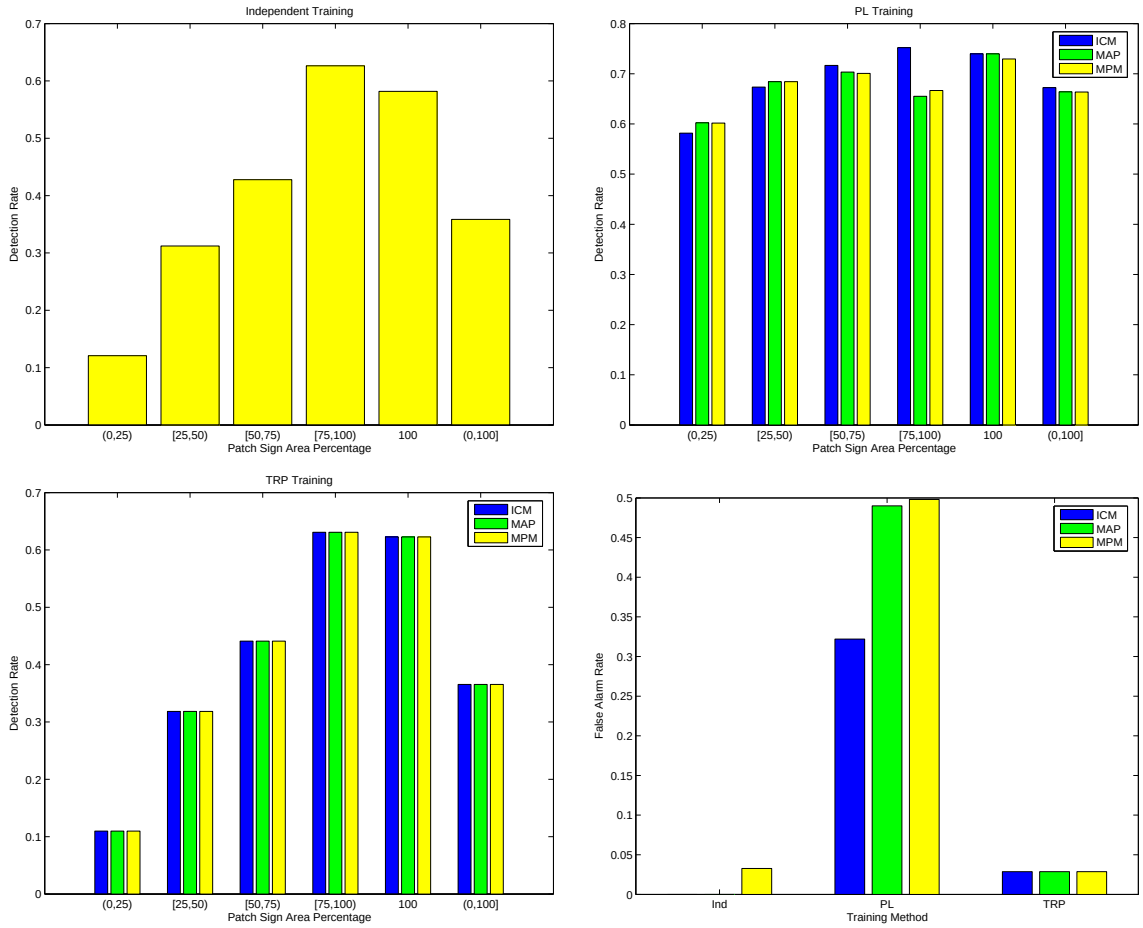


Figure 14: Detection and false alarm rates for various training and prediction methods with a scale-free Gaussian prior.

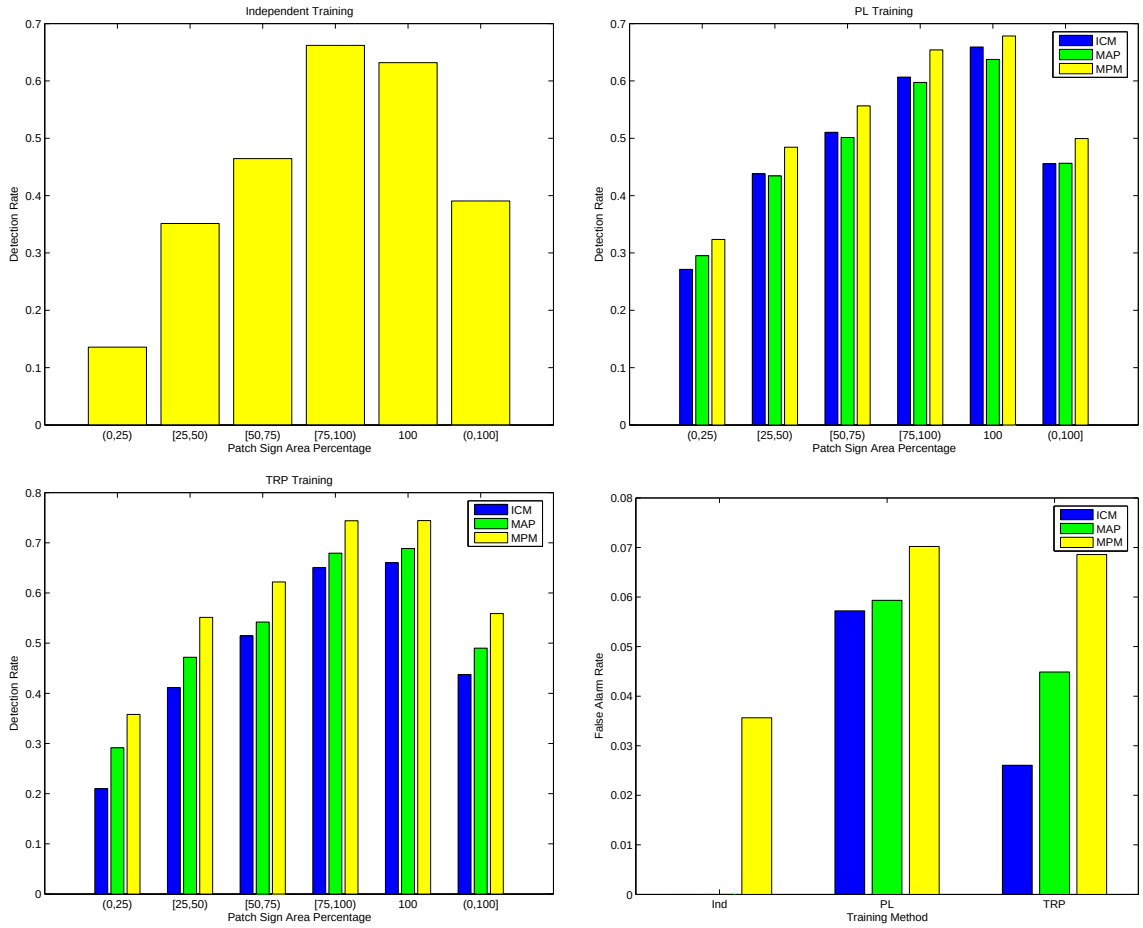
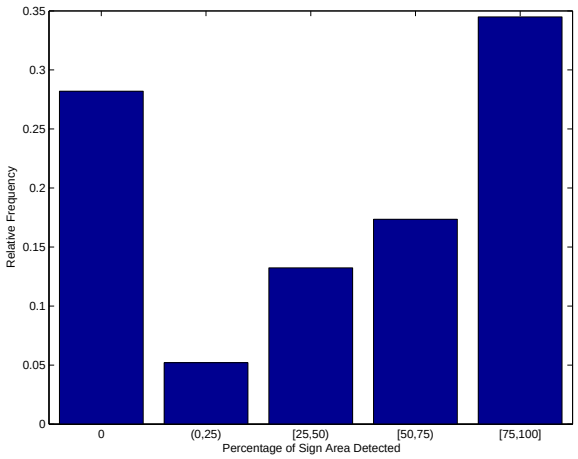


Figure 15: Detection and false alarm rates for various training and prediction methods with a Gaussian prior.



Sign Detection Rate	0.7180
Detection Coverage	
Average	0.6949 ± 0.2909
Median	0.7500

Figure 16: Sign-level results. The histogram on the left shows the relative frequency of the percentage of patches detected, while the right-hand table provides the overall sign-level detection rate and coverage statistics (fraction of detected patches) for detected signs.

has higher detection rates and fewer false alarms than the independent classifier even when both are trained no prior. Thus, even without regularization, the contextual power of the CRF largely outweighs its complexity and potential for over-fitting.

The scale-free Gaussian prior has some interesting effects. PL training with no prior leaves the node parameters with norm $\|\boldsymbol{\lambda}\|_2 = O(1e2)$. However, the prior effectively drives the node parameters to zero, leaving them with $\|\boldsymbol{\lambda}\|_2 = O(1e-13)$ after training, while the edge parameters still abnormally large at $\|\boldsymbol{\mu}_h\|_2 = O(1e2)$. Thus, while the node features are arguably more truthfully reliable, this regularization scheme eliminates them in favor of the edge parameters. These are much more useful in the pseudo-likelihood framework where the neighboring labels are given and practically determine the label in question. Thus the edge parameters contribute most to $\mathcal{PL}(\boldsymbol{\theta}, \mathcal{D})$.

The exact opposite behavior occurs when using TRP to train the CRF with the scale-free prior. While the edge parameters start with reasonable magnitudes $\|\boldsymbol{\mu}_h\|_2 \approx \|\boldsymbol{\mu}_v\|_2 = O(1e-1)$, regularization drives these to nearly zero, leaving them at $\|\boldsymbol{\mu}_h\|_2 = O(1e-8)$ and $\|\boldsymbol{\mu}_v\|_2 = O(1e-13)$. The node parameters, however, are virtually unchanged at $\|\boldsymbol{\lambda}\|_2 = O(1e2)$. Since the natural logarithm drops rapidly to negative infinity as its argument approaches zero, the log-norm form of the scale-free regularization term fiercely prefers small parameter norms. After likelihood maximization (training with no prior), the gradient of likelihood is nearly zero. However, once the prior is introduced, the gradient of the weights becomes slightly negative. Since the edge parameter norms are already relatively small, shrinking the weights yields a gain in prior probability that quickly outweighs the small loss in likelihood. After two steps of gradient ascent, the edge parameter norms have decreased by an order of magnitude and are quickly on their way to becoming insignificantly small. (This is why all prediction strategies show the same results in Figure 12: with virtually no interactions between the nodes they are effectively equivalent.) The gradient of this regularizer is inversely proportional to the norm; if the norm begins large, the regularization will not drive parameters to zero.

At initialization, the edge parameter norms of the TRP trained model are three orders of magnitude smaller than the node parameter norms, which explains why they are driven to zero first. However, due to the gradient ascent algorithm described in §5.2.3, the node parameters are not completely eliminated. The node parameter norms are driven to zero for the same reason in the PL trained model; at initialization they are one order of magnitude smaller than the node parameters. It is not clear whether this behavior can be useful in general, as it seems to be highly dependent on

Figure 17: Signs detected by a CRF trained using TRP and a Gaussian prior with ICM for prediction.

Figure 18: More signs detected by a CRF trained using TRP and a Gaussian prior with ICM for prediction.

Figure 19: Still more signs detected by a CRF trained using TRP and a Gaussian prior with ICM for prediction.

Figure 20: Conspicuous signs gone undetected by a CRF trained using TRP and a Gaussian prior with ICM for prediction.

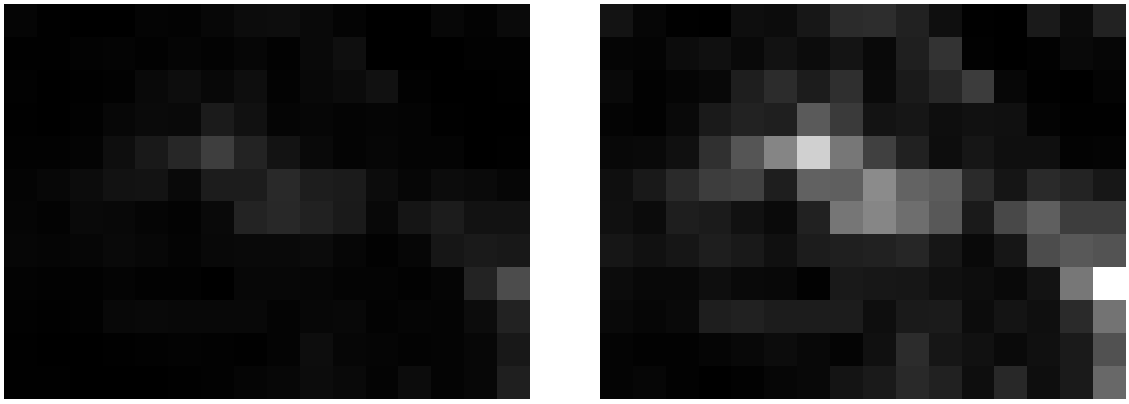


Figure 21: Marginal posterior probabilities for an image from Figure 20. Black represents $p\left(y = \mathsf{Sign} \mid \mathbf{x}\right) = 0$. For visualization, on the left white is $p\left(y = \mathsf{Sign} \mid \mathbf{x}\right) = 1$, while on the right white is $\max_{v \in S} p\left(y_v = \mathsf{Sign} \mid \mathbf{x}\right)$, the largest posterior probability in the image.

the partitioning of the parameter vector.

The behavior is not based strictly on parameter magnitude, but also how the prior interacts with parameter values and the likelihood. When all the model parameters are zero (yielding the uniform distribution), the log-likelihood is $\mathcal{L}\left(\mathbf{0}, \mathcal{D}\right) = -20,628$. When edge parameters are fixed at zero, the maximum log-likelihood is $\mathcal{L}\left(\left\langle \widehat{\boldsymbol{\lambda}}, \mathbf{0}, \mathbf{0} \right\rangle, \mathcal{D}\right) = -9,794$. When unfixed, the log-likelihood rises (as approximated by TRP) to $\mathcal{L}\left(\widehat{\boldsymbol{\theta}}, \mathcal{D}\right) = -3,395$. Thus, the edge features contribute to about one-third the total gain in likelihood. Similarly, in the pseudo-likelihood framework the node-parameters contribute practically nothing to the overall maximum pseudo-likelihood $\mathcal{PL}\left(\widehat{\boldsymbol{\theta}}, \mathcal{D}\right)$. In the independent model, there is only one set of parameters. The scale-free prior ends up halving the node parameter norm $\|\boldsymbol{\lambda}\|_2$, but decreases the likelihood to $\mathcal{L}\left(\left\langle \widehat{\boldsymbol{\lambda}}, \mathbf{0}, \mathbf{0} \right\rangle, \mathcal{D}\right) = -10,225$. The likelihood is thus strongly peaked in a region where the prior is still basically flat. All of this suggests a set of parameters that remain small or do not contribute strongly to the likelihood will be virtually eliminated by the scale-free prior. This is qualitatively positive behavior because it ensures only the strongest (set of) features are used. The drawback is that when features do have modest value (which, as we shall see, our edge features do) they are eliminated.

Next we examine the results of explicitly controlling the amount of regularization using our held-out validation set. The overall performance of the independent classifier changes insignificantly with the addition of the Gaussian prior. As expected, pseudo-likelihood training has a more dramatic change in performance. As the amount of regularization decreases, the parameters improve the value of the objective function $\mathcal{PL}\left(\boldsymbol{\theta}, \mathcal{D}\right)$. However, the result is eventually the worsened performance of honest prediction (no neighboring labels given) on the validation set. In a sense, choosing the scale yielding the best honest prediction give us something for nothing because it allows us to make the incorrect approximation as correct as possible. The ROC curves of Figure 12 show a clear improvement over the independent classification, mostly due to the patches ($(0, 25)\,\%$ sign) in drastic need of context.

The best overall performance is achieved by the TRP-trained CRF with a Gaussian prior. Its ROC curve for detecting all ($(0, 100]\,\%$) Sign patches is clearly the best of all, and with the ICM predictor it achieves both a higher overall detection rate and fewer false alarms than the best independent classifier. The PL-trained CRFs tend to have higher detection rates, but are accompanied by more false positives. Furthermore, the area under the ROC curves are decidedly higher for the TRP-trained CRF than for the PL-trained, indicating a superior detection/false-alarm trade-off. These experiments indicate that contextual processing improves the results when proper regularization is

Figure 22: Contrasting examples of ICM and MPM prediction by a CRF trained using TRP and a Gaussian prior.

used.

One interesting, though perhaps subtle, phenomenon is worth addressing. Almost universally, the ROC curve area and detection rate of Sign patches which contain $[75, 100)$ % true sign are higher than those of patches containing 100% sign, especially in the context-free independent classifier. We speculate this is because such a patch is largely filled with data indicative of a sign (e.g., text) but also contains a crucial bit of context: namely, some sort of edge or boundary. Therefore, this case could be easier to detect than a patch that is purely sign, which may have some feature-space overlap with Background patches.

Finally we address some of the visual results. Figure 17 contains signs both straightforward and not. The upper-left shows the example image traced throughout the paper. Although the grating cell-based filters responded well to much of the text present in the image, those signs with smaller text are not detected as sign. That smaller signs go undetected is a dominant trend in our results. Perhaps small-scale grating cells respond to too many things besides text (e.g., foliage); more discriminative features will need to be found before our system performs better with smaller signs. The center-left image of the figure demonstrates reasonable performance even when specular reflection would be expected to degrade results. The lower-left image illustrates detection of signs with logos as well. Figure 18 demonstrates detection on signs with fonts of many sizes and types and in varying amounts of rotation and foreshortening. The two images at the bottom of Figure 19 illustrate one difficulty with ground truth. For visual recognition purposes, the patches registered as false positives in the bottom-left image perhaps should be considered part of the sign, since they involve the logos and entire awning. Conversely, we are reluctant to call the remaining undetected patches inside the hand-drawn contour of the lower-right false negatives because arguably the most important part of the sign has been detected. We have not made any attempt to adjust the ground-truth after the fact, but we present these examples to demonstrate that it is still unclear whether the current performance might be fully satisfactory in a deployed system.

While MPM prediction boasts a much higher detection rate than ICM, it is accompanied by triple the false alarm rate. We therefore have presented some results due to ICM prediction because they are more visually appealing. Figure 22 highlights some typical differences between the two methods. The top and bottom show cases were ICM does end up missing crucial patches. However, the false positives of the center-right image are typical of the over-zealous detection resulting from the MPM decision rule. Perhaps the largest advantage of ICM is that it is *very* fast: once features are computed, just 10 iterations of MAX operations on an image yield about the best possible results. It boosts detection over independent classification by 4% and cuts the false positive rate by one-fourth, on average eliminating 1-2 false positive patches per image.

Due to the uncertainty in ground truth and a presumed robustness on the part of a sign recognizer to relatively small amounts of additional bordering image, we relax our evaluation slightly. If we eliminate from consideration the Background patches that are immediate neighbors to Sign patches, the false positive rate of 0.0261 reported for the TRP-CRF with ICM prediction (Figure 15) drops to 0.0175

As mentioned earlier, one common cause for non-detection is small size. We conjecture the cause of this to be prevalence of similar texture features at that scale in Background patches. There are more conspicuous undetected signs in Figure 20. Some rotation is present in signs that are successfully detected, but smaller text combined with rotation and some foreshortening would appear to be the cause of some failures. We especially note that there is little space between letters in the center-left image, lower-left image, and the leftmost sign of the upper-right image. We believe the primary cause of failure in the lower-right image to be the vertical orientation, rather than the foreign alphabet. Perhaps 4-5 signs in the entire dataset contain vertical text, which is an insufficient number to learn the properties of such regions. The center-right image contains definite peaks in the marginal sign probabilities where they would be expected, as can be seen from Figure 21. However, they only reach a probability of about 0.298 which is insufficient for detection by MPM.

6.2 Related Work

Text Detection Earlier approaches to text detection either use independent, local classifications or connected component analysis, and are based on edge detectors or more general texture features. Garcia and Apostolidis [21] use edge detectors and morphological operations to remove noise and fill in dense edgel areas, finally using local heuristics to label a region as text. Gao and Yang [19, 20] use a pipelined approach involving edge detection, adaptive search, color modeling, and layout analysis. Jain and Bhattacharjee [26] treat text more like a general texture, filtering images with a bank of Gabor filters and clustering to classify pixels. Li et al. [35] use a neural network trained on the mean and second- and third-order central moments of wavelet coefficients to independently classify blocks of pixels. Wu et al. [64] use a non-linear function of multiscale Gaussian derivatives to identify regions of high energy, which typically correspond to text. To clean and isolate regions of text, they employ heuristics for finding only horizontal, linear text strings. All of these approaches either rely on thresholded filter outputs and extensive use of heuristics or make implicit independence assumptions. Our approach calculates the most likely labeling of image regions as text *jointly*, rather than independently, and obviates layout heuristics by allowing a conditional random field to learn the characteristics of regions that contain text.

The most recent and relevant work involves text detection for signs in natural images. This differs from detection of artificial, superimposed text in images and video which tends to be an easier problem. The system of Gao and Yang [19, 20] detects Chinese characters in images. Their data appears to be mostly frontal with little slant or foreshortening. Our data contains signs with such text, but it remains a challenge for our detector, detracting from our results numerically. They report a seemingly large false alarm rate (.101), but do achieve and impressive detection rate of 0.992. More recently, they have introduced a system for rectifying text with affine transformations [11]. Visual results of detection are impressive, though no numbers are given.

Another system has been developed by Chen and Yuille [10]. They use a cascaded AdaBoost classifier to detect text of all sizes in images. By using features fixed to a particular window size, they are able to train a single classifier and run it at several window sizes to detect text at many scales. The features are a combination of intensity and gradient statistics, histograms motivated by the bimodal nature of text regions, and edge linking. Their results are impressive with detection rates of 0.982. Although only an absolute number of false positives is reported, rather than the rate, they do report an average of 4 false positives per image, which would roughly correspond to our false alarm rate of 0.0261. Both of these systems are geared very specifically toward text. Our purpose is to generalize signs to a class of textures to enable recognition of signs consisting of logos and graphics in addition to (or instead of) text, allowing for broader visual recognition techniques. Therefore we have started with some more general image features.

Texture Features The features used by any heuristic or learning algorithm are important. One of the most popular features for classifying textures is the orientation and scale selective Gabor filter [13, 26, 61, 39]. Typically Gabor filters are combined to create a non-orthogonal basis covering the frequency spectrum; however, Weldon [61] used optimal and approximately optimal techniques to intentionally center Gabor filters in order to maximize texture discrimination. Zhu et al. [66, 67] used histograms of outputs from a Gabor filter bank and Laplacian of Gaussian filters to generate textures from a maximum entropy (i.e. Gibbs) random field. Their model produced reasonable textures but was quite slow in both training and synthesis. More recently, Portilla and Simoncelli [45, 46] generated textures with a fast successive projection algorithm using features based on the steerable pyramid [54], an overcomplete wavelet decomposition. Rather than simply taking spatial marginals of filter responses, they capture moments of filter responses and introduce auto-correlation and inter-filter statistics, including correlation across scales and orientations. The necessity of such features for proper texture synthesis are demonstrated in [46]. A comprehensive study of several filters for texture classification by Randen and Husøy [50] showed that the Gabor filters, both general and optimized, were outperformed in almost all cases by another filter. However, the study merely used a local average of filter response energy as the feature for classifying individual pixels. This captures far less information about the distribution of responses than a full histogram (e.g., [66]),

to say nothing of explicit relationships between coefficients (e.g., [46]).

Motivated by recent discoveries about the mammalian visual system, Kruizinga and Petkov [30] proposed the grating cell, a nonlinear texture operator that responds to multiple oriented bars of particular scales. Unlike traditional operators (i.e., Gabor filters) that respond to any single edge, these operators are specifically geared toward repeated features—texture. In comparisons that were by no means exhaustive or conclusive, [30] demonstrated that the discriminative power of the grating cell is double that of the typical Gabor filter bank; another comparison illustrated superiority of the grating cell over the Gabor filters in several cases [23]. From these results, it is clear that the raw energy of Gabor filter output is insufficient to capture much texture information. However, it is not immediately clear whether the addition of correlation and phase statistics (as in [46]) will perform on par with the grating cell.

Markov Field Modeling Long studied in computer vision and image processing applications, Markov fields are being revived with the introduction of conditionally trained versions by Lafferty, McCallum, and Pereira [32]. The improved segmentation and detection performance of the conditional Markov field in computer vision applications was first noted by Kumar and Hebert [31], where a grid model was trained using pseudo-likelihood. We noted further improvements resulting from free-energy based likelihood approximations such as TRP in [59]. More global label context was incorporated to the grid CRF for segmentation by He, Zemel, and Carreira-Perpiñán [25]. Different techniques and graph structures are now being used for more general object recognition by Torralba, Murphy, and Freeman [56] and Quattoni, Collins, and Darrell [49].

7 Conclusions and Future Work

We demonstrated the utility of contextual information for an application in generic sign detection. The conditionally-trained grid-shaped Markov field model, which has been shown previously to give improved detection performance over generative models, allows us to use arbitrary image features. Incorporating large scale features and the relative similarity between neighboring patches while considering the joint labeling of an image demonstrably improves detection rates and reduces false alarms.

The addition of more specialized features geared specifically toward text detection should improve our results. Furthermore, making more refined decisions (i.e., smaller patches and larger grids) will boost the number of Sign patches that are unpolluted by background image data. By training on Sign patches with *only* sign image content, the results of a local independent classifier are greatly improved [53]. It remains to extend this to the CRF model, incorporating the useful spatial context. If ICM remains a useful prediction method, the increased grid size should only increase training time, not hampering the speed of test-time detection.

Adding context definitively improves accuracy in our application, but the power of our current features to strongly discriminate between signs and background remains a limitation. By calculating more accurate probabilities and parameter estimates, better approximate inference techniques might yield better results in such cases. Obvious candidates include better region-based free energy approximations like the Kikuchi free energy, which takes advantage of the grid topology by adding regions consisting of four node loops.

Another matter for consideration is the particular types of error that feature estimates may be prone to. For instance, in language modeling, the estimates are relative frequencies of certain features of a small dataset. It is more likely that some features are witnessed more rarely than the expected frequency, so we should expect positive errors. Bayesian analysis yields better results when the prior information is better, and the "dual" relaxed maximum entropy problem should not be different. Taking account of the tendency of errors such as these for the various applications where the altered maximum entropy method is used is a promising area for research.

Acknowledgments

This work was made possible by NSF grant IIS-0100851. Thanks to the developers of MALLET [40], especially Khashayar Rohanimanesh, Aron Culotta, and Charles Sutton for their extensive, assistive discussions. Gratitude to Robert Heller for systems support through all stages of this work. Thanks also go to Anne-Marie Strohman for helpful comments on early versions of this paper.

Appendix A

	Training Method		
Patch Sign Area	Ind.	PL	TRP
$(0, 25)$	0.7098	0.5327	0.7423
$[25, 50)$	0.8330	0.5686	0.8313
$[50, 75)$	0.8730	0.5762	0.8494
$[75, 100)$	0.9209	0.5550	0.8777
100	0.8990	0.5828	0.8792
$(0, 100]$	0.8224	0.5579	0.8195

Table 1: Area under ROC curves for posterior marginals with no prior (cf. Figure 10).

	Training Method		
Patch Sign Area	Ind.	PL	TRP
$(0, 25)$	0.7099	0.7456	0.7632
$[25, 50)$	0.8364	0.8322	0.8566
$[50, 75)$	0.8776	0.8624	0.8835
$[75, 100)$	0.9243	0.8993	0.9183
100	0.9025	0.9009	0.9162
$(0, 100]$	0.8249	0.8295	0.8488

Table 2: Area under ROC curves for posterior marginals with a Gaussian prior (cf. Figure 12).

	Training Method		
Patch Sign Area	Ind.	PL	TRP
$(0, 25)$	0.7025	0.4329	0.7031
$[25, 50)$	0.8331	0.4623	0.8342
$[50, 75)$	0.8729	0.4619	0.8796
$[75, 100)$	0.9206	0.4465	0.9244
100	0.8964	0.4731	0.9031
$(0, 100]$	0.8195	0.4517	0.8226

Table 3: Area under ROC curves for posterior marginals with a scale-free Gaussian prior (cf. Figure 11).

Patch Sign Area	MAP
$(0, 25)$	0.1512
$[25, 50)$	0.3461
$[50, 75)$	0.4593
$[75, 100)$	0.6494
100	0.6409
$(0, 100]$	0.3934
False Alarm Rate	0.0352

Table 4: Detection and false alarm rates using independent training with no prior (cf. Figure 13).

Patch Sign Area	MAP
$(0, 25)$	0.1208
$[25, 50)$	0.3122
$[50, 75)$	0.4278
$[75, 100)$	0.6263
100	0.5818
$(0, 100]$	0.3586
False Alarm Rate	0.0327

Table 5: Detection and false alarm rates using independent training with a scale-free Gaussian prior (cf. Figure 14).

Patch Sign Area	MAP
$(0, 25)$	0.1357
$[25, 50)$	0.3515
$[50, 75)$	0.4646
$[75, 100)$	0.6621
100	0.6318
$(0, 100]$	0.3906
False Alarm Rate	0.0356

Table 6: Detection and false alarm rates using independent training with a Gaussian prior (cf. Figure 15).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.5418	0.5086	0.5125
$[25, 50)$	0.6467	0.5995	0.6173
$[50, 75)$	0.7008	0.6299	0.6391
$[75, 100)$	0.7543	0.5975	0.6286
100	0.7489	0.6273	0.6795
$(0, 100]$	0.6532	0.5777	0.5974
False Alarm Rate	0.2797	0.2955	0.3048

Table 7: Detection and false alarm rates for the various prediction methods using PL training with no prior (cf. Figure 13).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.5817	0.6022	0.6017
$[25, 50)$	0.6735	0.6842	0.6842
$[50, 75)$	0.7165	0.7034	0.7008
$[75, 100)$	0.7520	0.6551	0.6667
100	0.7398	0.7398	0.7295
$(0, 100]$	0.6723	0.6640	0.6637
False Alarm Rate	0.3220	0.4900	0.4980

Table 8: Detection and false alarm rates for the various prediction methods using PL training with a scale-free Gaussian prior (cf. Figure 14).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.2715	0.2953	0.3235
$[25, 50)$	0.4380	0.4344	0.4844
$[50, 75)$	0.5105	0.5013	0.5564
$[75, 100)$	0.6067	0.5975	0.6540
100	0.6591	0.6375	0.6784
$(0, 100]$	0.4556	0.4565	0.4995
False Alarm Rate	0.0572	0.0593	0.0702

Table 9: Detection and false alarm rates for the various prediction methods using PL training with a Gaussian prior (cf. Figure 15).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.1911	0.3546	0.4089
$[25, 50)$	0.4023	0.5004	0.5763
$[50, 75)$	0.5000	0.6089	0.6496
$[75, 100)$	0.6378	0.6909	0.7336
100	0.6307	0.7023	0.7477
$(0, 100]$	0.4204	0.5303	0.5838
False Alarm Rate	0.0289	0.0665	0.0938

Table 10: Detection and false alarm rates for the various prediction methods using TRP training with no prior (cf. Figure 13).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.1097	0.1097	0.1097
$[25, 50)$	0.3185	0.3185	0.3185
$[50, 75)$	0.4409	0.4409	0.4409
$[75, 100)$	0.6309	0.6309	0.6309
100	0.6227	0.6227	0.6227
$(0, 100]$	0.3654	0.3654	0.3654
False Alarm Rate	0.0287	0.0287	0.0287

Table 11: Detection and false alarm rates for the various prediction methods using TRP training with a scale-free Gaussian prior (cf. Figure 14).

Patch Sign Area	ICM	MAP	MPM
$(0, 25)$	0.2100	0.2914	0.3579
$[25, 50)$	0.4112	0.4719	0.5513
$[50, 75)$	0.5144	0.5420	0.6220
$[75, 100)$	0.6505	0.6794	0.7439
100	0.6602	0.6886	0.7443
$(0, 100]$	0.4374	0.4900	0.5590
False Alarm Rate	0.0261	0.0449	0.0686

Table 12: Detection and false alarm rates for the various prediction methods using TRP training with a Gaussian prior (cf. Figure 15).

References

[1] Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society*, B(26):192–236, 1974.

[3] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[4] J. Besag. Efficiency of pseudo-likelihood estimation for simple gaussian fields. *Biometrika*, 64:616–618, 1977.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] W. Buntine and A. Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

[7] R. Byrd, J. Nocedal, and R.B. Schnabel. Represenations of quasi-Newton matrices and their use in limited memroy methods. *Mathematical Programming*, 63:129–156, 1994.

[8] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Proc. European Conf. on Computer Vision*, volume 1, pages 350–362, 2004.

[9] Stanley F. Chen and Ronald Rosenfeld. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, 8(1), Jan. 2000.

[10] Xiangrong Chen and Alan L. Yuille. Detecting and reading text in natural scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 366–373, 2004.

[11] Xilin Chen, Jie Yang, Jung Zhang, and Alex Waibel. Automatic detection of signs with affine transformation. In *Proceedings of the 2002 IEEE Workshop on Applications in Computer Vision (WACV2002)*, pages 32–36, Dec. 2002.

[12] John G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20(10):847–856, 1980.

[13] John G. Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[14] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[15] H. Elliot, H. Derin, R. Cristi, and D. Geman. Application of the Gibbs distribution to image segmentation. In *Proceedings of the International Conference on Acoustic, Speech, and Signal Processing*, pages 32.5.1–32.5.4, 1984.

[16] Susana Eyheramendy, Alexander Genkin, Wen-Hua Ju, David D. Lewis, and David Madigan. Sparse bayesian classifiers for text categorization. *Journal of Intelligence Community Research and Development*. Submitted.

[17] Colin Fox and Geoff Nicholls. Exact MAP states and expectations from perfect sampling: Greig, Porteous and Seheult revisited. In *Proc. Twentieth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 252–263, 2001.

[18] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, October 2000.

[19] Jiang Gao and Jie Yang. An adaptive algorithm for text detection from natural scenes. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 84–89, December 2001.

[20] Jiang Gao, Jie Yang, Ying Zhang, and Alex Waibel. Text detection and translation from natural scenes. Technical Report CMU-CS-01-139, Carnegie Mellon University, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 2001.

[21] C. Garcia and X. Apostolidis. Text detection and segmentation in complex color images. In *Proceedings of 2000 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2000)*, volume 4, pages 2326–2330, June 2000.

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

[23] Simona E. Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, 2002.

[24] J.M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. Unpublished, 1971.

[25] Zuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Proceeedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2004)*, pages 695–702, 2004.

[26] A.K. Jain and S. Bhattacharjee. Text segmentation using Gabor filters for automatic document processing. *Machine Vision Applications*, 5:169–184, 1992.

[27] E.T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[28] E.T. Jaynes. *Probability Theory : The Logic of Science*. Cambridge University Press, 2003.

[29] B.W. Jeon and D.A. Landgrebe. Classification with spatio-temporal interpixel class dependency contexts. *IEEE Transactions on Geoscience and Remote Sensing*, 30(4):663–672, July 1992.

[30] Peter Kruizinga and Nikolay Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 8(10):1395–1407, 1999.

[31] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. 2003 IEEE International Conference on Computer Vision (ICCV '03)*, volume 2, pages 1150–1157, 2003.

[32] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[33] Steffen L. Lauritzen. *Graphical Models*. Number 17 in Oxford Statistical Science Series. Clarendon, 1996.

[34] Guy Lebanon and John Lafferty. Boosting and maximum likelihood for exponential models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 447–454, 2001.

[35] Huiping Li, David Doermann, and Omia Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, 2000.

[36] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. Computer Science Workbench. Springer-Verlag, Tokyo, second edition, 2001.

[37] J. Liu and L. Wang. MRMRF texture classification and MCMC parameter estimation. In *Visual Interface '99*, volume 20, pages 171–182, 1999.

[38] David J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–448, 1992.

[39] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):837–842, 1996.

[40] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. http://www.cs.umass.edu/∼mccallum/mallet, 2002.

[41] Joseph L. Mundy and Tom Strat, editors. *IEEE Worskshop on Context-Based Vision*, June 1995.

[42] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI (UAI'99)*, pages 467–475, 1999.

[43] Giorgio Parisi. *Statistical Field Theory*. Perseus, 1998.

[44] N. Petkov and P. Kruizinga. Computational model of visual neurons specialised in the detection of period and aperiodic oriented visual stimuli: bar and grating cells. *Biological Cybernetics*, 76:83–96, 1997.

[45] Javier Portilla and Eero P. Simoncelli. Texture modeling and synthesis using joint statistics of complex wavelet coefficients. In *Proceedings of the IEEE Workshop on Statistical and Computational Theories of Vision*, June 1999.

[46] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.

[47] Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. In *Proc. 10th Intl. Workshop on Artificial Intelligence and Statistics (AISTATS05)*, 2005.

[48] Yuan Qi, Martin Szummer, and Thomas P. Minka. Diagram structure recognition by bayesian conditional random fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 191–196, 2005.

[49] Ariadna Quattoni, Michael Collins, and Trevor Darrel. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, 2005.

[50] Trygve Randen and John Håkon Husøy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.

[51] Stefan Riezler and Alexander Vasserman. Incremental feature selection and L1 regularization for relaxed maximum-entropy modeling. In *EMNLP04*, 2004.

[52] Piyanuch Silapachote. Hierarchical approach to sign classification learning methods and vision-based sign recognition techniques. Master's thesis, University of Massachusetts-Amherst, Computer Science Research Center, University of Massachusetts, Amherst, MA 01003-4601, 2004.

[53] Piyanuch Silapachote, Jerod Weinman, Allen Hanson, Richard Weiss, and Marwan A. Mattar. Automatic sign detection and recognition in natural scenes. In *IEEE Workshop on Computer Vision Applications for the Visually Impaired*, June 2005.

[54] Eero P. Simoncelli and William T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.

[55] T.M. Strat and M.A. Fischler. Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, Oct. 1991.

[56] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, 2005.

[57] Antonio Torralba, Kevin P. Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 273. IEEE Computer Society, 2003.

[58] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Image Processing*, 49(5):1120–1146, 2003.

[59] Jerod Weinman, Allen Hanson, and Andrew McCallum. Sign detection in natural images with conditional random fields. In *IEEE Intl. Workshop on Machine Learning for Signal Processing*, pages 549–558, September 2004.

[60] J.R. Welch and K.G. Salter. A context algorithm for pattern recognition and image interpretation. *IEEE Transactions on Systems, Man, and Cyberneics*, 1(1):24–30, January 1971.

[61] Thomas P. Weldon, William E. Higgins, and Dennis F. Dunn. Gabor filter design for multiple texture segmentation. *Optical Engineering*, 35(10):2852–2863, 1996.

[62] Peter M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7:117–143, 1995.

[63] Gerhard Winkler. *Image Analysis, Random Fields, and Markov Chain Monte Carlo Methods*. Springer-Verlag, Berlin, second edition, 2003.

[64] Victor Wu, R. Manmatha, and Edward M. Riseman. Finding text in images. In *DL'97: Proceedings of the 2nd ACM International Conference on Digital Libraries*, pages 3–12, 1997.

[65] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2004-040, Mitsubishi Electric Research Laboratories, May 2004.

[66] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.

[67] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME)–towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):1–20, 1998.