Image Modeling using Hierarchical Conditional Random Field

A THESIS

submitted by

AAKANKSHA GAGRANI

for the award of the degree

of

MASTER OF SCIENCE

(by Research)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

June 2007

DEDICATION

This thesis is dedicated to my beloved grandmother Late Smt. Anoop Devi Gagrani, who expired during my term here as an M.S. Scholar. May God give peace to her kind soul.

THESIS CERTIFICATE

This is to certify that the thesis titled **"Image Modeling using Hierarchical Conditional Random Fields"**, submitted by **Aakanksha Gagrani** to the Indian Institute of Technology Madras, for the award of the degree of **Master of Science**, is a bonafide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. B. Ravindran
Research Guide
Assistant Professor
Department of Computer Science and Engineering
IIT Madras, CHENNAI-600036

Prof Koshy Varghese
Co-Guide
Professor
Department of Civil Engineering
IIT Madras, CHENNAI-600036

Date: June. 7, 2007

ACKNOWLEDGEMENTS

I am thankful to my advisor Dr. B. Ravindran for providing me a careful guidance throughout my graduate program at IIT Madras. His knowledge of the subject and enthusiasm in this research problem has been of great benefit to me. He has been very patient and critical all through my work.

I would like to thank my co-guide Dr. Koshy Varghese for his help, inspite of being from Civil Department he has benefited me with his valuable suggestions.

I am also thankful to my Project Coordinator Dr. Sukhendu Das. The project offered by him was of great help for my research work . Dr. Das has been very helpful by letting me work in the Visualization and Perception Lab and gaining experience in the domain of the computer vision.

I take this opportunity to thank the General Test Committee members, Dr. Sukhendu Das, Dr. C.S Ramalingam and Dr. Koshy Varghese for their interest, encouragement, valuable suggestions and thoughtful reviews.

My sincere thanks to Prof Timonthy A. Gonzalves (current-HOD) for providing the best possible facilities to carry out the research work. Prof B. Yegnanrayna , Dr. C. Chandrashekhar and Prof. Hema Murthy, for their role in building up the foundation in subjects of Pattern Recognition and Artificial Neural Networks, useful for my research.

My thanks are due to the Computer Science Engineering office and laboratory staff Mrs Sharada, Mr. Natrajan, Ms Poongodi, Mrs Prema, Mr Balu (at the department library) for their valuable cooperation and assistance.

A special vote of thanks for my colleague Pranjal Awasthi, his ideas and assistance in implementation have been of immense help in my research work.

My lab-mates Manika, Abhilash, Shivani, Sunando, Manisha, Deepti, Surya, Lalit, Pragya, Mirnalinee, Dyana, Poongodi, Vinod for being tolerant and cooperative. My special thanks to Sadhna , Lalit, Ramya ,Sunando and Manika for having long hours of research discussions with me and their advice during thesis writing.

A special word of thanks to my dear friend Vivek Sheel Singh for being a source of motivation and encouragement.

Last but not least, I would like to thank my parents for being a source of encouragement and strength all throughout.

ABSTRACT

Image modeling methods broadly fall under two categories, (**a**) Methods which consider bottom up cues , these methods take pixel statistics into consideration for classification. (**b**) Methods which consider top down cues as well, these methods utilize the context information as well for the modeling task. Natural images exhibit strong contextual dependencies in the form of spatial interactions among components. For example, neighboring pixels tend to have similar class labels, and different parts of an object are related through geometric constraints. Going beyond these, different regions e.g., objects such as, monitor and keyboard appear in restricted spatial configurations. Modeling these interactions is crucial for achieving good classification accuracy.

In this thesis, we present a method based on Hierarchical Conditional Random Fields that can handle both bottom up and top down cues simultaneously. This offers tremendous advantages over existing methods which use either bottom up cues, or model only limited contextual information. The Tree Structured Conditional Random Field (TCRF) as proposed, is capable of capturing long range correlations at various levels of scales. TCRF has non loopy graph structure which allows us to perform inference in time linear in number of nodes. The model is generic enough to be applied to several challenging computer vision tasks, such as object detection and image labeling; seamlessly within a single, unified framework.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

MRF	Markov Random Field
TCRF	Tree Structured Conditioanl Random Field
mCRF	Multiscale Conditional Random Field
DRF	Discriminative Random Field
CRF	Conditional Random Field
CD	Contrastive Divergence
MAP	Maximum A Posteriori
MPM	Maximum Posterior Marginals
LBP	Loopy belief Propagation
ML	Maximum Likelihood estimate
EM	Expectation Maximization
LR	Logistic Regression

CHAPTER 1

Introduction

1.1 Motivation

Among the various high level and low level computer vision tasks, *semantic scene interpretation* or *image understanding* is considered an important and challenging task. For example, suppose we need to design a system capable of discriminating an office scene from a kitchen scene (Figure 1.1). This involves identifying the objects of the scene as some meaningful entities. The initial step would be to parse the image, segment it into meaningful regions and label those segments as known entities. For classifying the scene as an office area, we need to first segment objects like a computer, keyboard, books etc. Similarly for classifying it into a kitchen scene we need to identify the kitchen objects like utensils, microwave, gas stove etc. The system should also be able to capture the *context relationship* between the various objects like keyboard and computer. This gives us a motivation to build an image modeling system, which would provide us with a different representation of the underlying image; representation which will be useful in carrying out high level tasks like scene interpretation or image understanding. In this thesis we propose a probabilistic hierarchical image modeling system based on Conditional Random Fields (Lafferty *et al.*, 2001). The model efficiently learns the complex class dependencies and produces labels over the input image. The performance

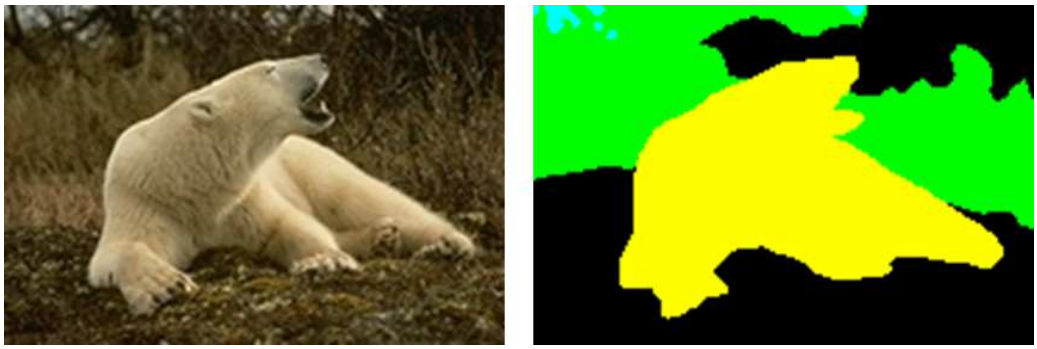
Fig. 1.1: Images showing office and kitchen scene. The discrimination can be done only by first identifying the objects contained in the scene like microwave or computer.

of our model has been evaluated and verified by experimenting with two key problems of computer vision, namely *image labeling* (a multiclass problem where each pixel belongs to one of the predefined classes) and *object recognition*(a two class problem where object of interest are identified) (Figure 1.2). The images considered for the two tasks are natural images encountered commonly in our surroundings. These images may contain both man-made as well as other natural objects such as sky, vegetation etc. It is assumed that the images are static in nature and no motion information is included.

1.2 Problem Statement

The proposed model has been applied on two different domains of image modeling task as defined below,

- Image Labeling - Given a test image, the aim is to segment it into regions and then label each region from one of the pre defined set of classes. The images considered for this task are wildlife scenes from the Corel dataset. It is a multiclass problem.
- Object Detection - Given a test image, the aim is to identify the object blocks. The objects to be detected here can be either man made structure or animals (different dataset). It is a two class problem.



(a)

(b)

Fig. 1.2: (a) Image labeling - a multiclass problem where each pixel belongs to one of the predefined classes. (b) Object Detection - Detecting objects of interest in a test image. For example, man made structure in this image.

1.3 Issues and Ambiguities

Traditional approaches for the task of image modeling mostly suffer due to the suboptimal choice of the feature set and ambiguity in spectral signatures. These ambiguities may arise either due to the physical conditions such as illumination and pose of the scene components with respect to the camera, or due to the intrinsic nature of the data itself. Classification of image components relying solely on the spectral features, becomes a difficult task due to these ambiguities. For example, as shown in Figure 1.3(a), discriminating between the two white blobs is not a trivial task even for a human eye. The same two white blobs can be easily classified into snow and sky, given Figure 1.3(b) because information about their context is also available. An image modeling system capable of incorporating

Fig. 1.3: In the left figure white blobs visually appear to belong to the same class, when context is added as shown in the right image, the blobs actually belong to different classes namely sky and snow.

context relationship in addition to visual information is intuitively more similar to a human visual system. Figure 1.4 explains how context can be captured by varying the context window scale. Context window is the viewing area of the image. As shown in Figure 1.4, when the viewing area is expanded, the ambiguities keep fading away. As shown, in the smallest scale window, discriminating A from B is ambiguous, but at the higher scales of context window, presence of rocks and ground near the water help disambiguate the confusion.

The two examples presented above give us an intuition that strong context information is present in natural images and can be captured by learning the label relationship. It is vital for any image modeling system to take into account the context relationships present among various objects at different scales for obtaining good classification accuracy. We call this *multiscale* nature of the context.

This thesis proposes a hierarchical model based on Conditional Random Fields. Our model captures context at different levels of scale by using a hierarchy of hidden variables in a tree like graph structure.

1.4 Contextual Interactions

Different types of contextual interactions can be observed in an image. Consider a small region of an object, the neighboring pixels will have high likelihood of belonging to the same class. This can be seen as local context. The other kind of context is global context, where objects appear in a constrained spatial configuration. Different regions follow plausible spatial interactions, like sky appears to be at the top, and snow appears to be at the ground or the lower level. Then, there can be interactions where two or more objects have higher probability of occurring in close proximity like, kettle and stove, keyboard and mouse etc.

In summary, we can classify context as *local* when interactions of pixels is with their neighboring pixels and *global* when interactions is among bigger regions. The solution provided in this thesis addresses problem of capturing context and complex dependencies in a single, unified framework in a principled manner.

In the vision literature most of the methods (Geman and Geman, 1984; Bouman and Shapiro, 1994) could only model context at the pixel or the region based level. Multiscale conditional random fields proposed by Xuming (He *et al.*, 2004) is an attempt to capture the context at various scales. However, the model has restriction over the number of scales it can handle tractably. The approach is also not generic enough to be extended to tasks other than image labeling.

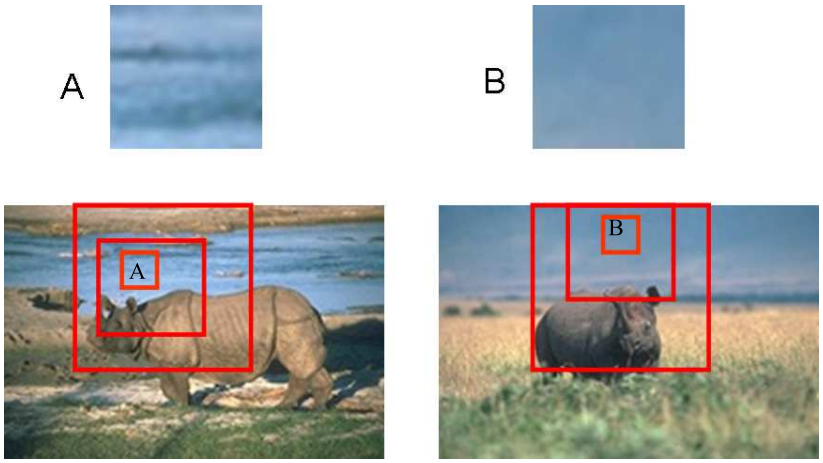


Fig. 1.4: The classes A and B become clearer as we expand our scale of view .

1.5 Background and Related work

A variety of signal processing, image processing and machine learning methods exist for meaningful and efficient image modeling tasks. We can broadly classify those methods as *non-probabilistic* and *probabilistic*.

The framework is categorized as non-probabilistic if the overall labeling objective is not given by a consistent probabilistic formulation, even if the framework utilizes probabilistic methods to address parts of it. Rule-based context (Ohta, 1980) and relaxation labeling (Rosenfeld *et al.*, 1976) are two main techniques among the non-probabilistic approaches, other than using weak measures to capture spatial smoothness of natural images using filters with local neighborhood supports. Ohta used a rule-based approach to assign labels to regions obtained from a single-pass segmentation. The stumbling block in case of rule based approaches was their inability to handle the statistical variations in the data. Singhal (Singhal *et al.*, 2003) proposed the use of conditional histograms to make a local decision regarding assigning a label to a new region given the previous regions labels, to avoid the absolute constraints imposed by the rule-based approaches.

However, such a sequential implementation of context will suffer if an intermediate region is assigned a wrong label. In late-1970s, the VISIONS schema system was proposed by Hanson and Riseman (1978), which provides a framework for building a general interpretation system based on the output of several small special purpose experts. Each scheme is an 'expert' at recognizing one type of object. The schema instances run concurrently and communicate asynchronously. Each expert gives its prediction about the presence and location of objects in the image. Based on hand coded if-then rules, the system analyzes the consistency among the different hypotheses in order to arrive at reliable decisions. Later, a similar idea was presented by Strat (1992) in a system called CONDOR to recognize natural objects for the visual navigation of an autonomous robot. The system used hand-written rules to encode the knowledge database of the system. A collection of rules (context sets) defines the conditions under which it is appropriate to use an operator to identify a candidate region or object. While analyzing generic 2D images, manually defining the context information is not a very convenient task. Instead, one needs to derive the context directly from the input image itself. A comprehensive review of the use of context for recognizing natural objects in color images of outdoor scenes is given in (Batlle *et al.*, 2000). Torralba and Sinha (2001) proposed a framework for modeling the relationship between context and object properties based on the correlation between the statistics of low-level features across the entire scene and the objects that it contains.

In Summary, the non probabilistic models suffered because they tried to manually visualize the context and extract them using rule based expert. Image analysis application have embedded uncertainty (Rao and Jain, 1988; Winston,

1970) in them, which needs a more robust and principled approach. Even though efforts were made to represent global uncertainty using graph structures, the tools available for learning and inference over these structures were limited. These ad-hoc procedures for resolving ambiguities using rules systems were unreliable or constrained to a narrow domain.

Image classification methods consider both spectral statistics and uncertainties in the dependencies for the classification task, which intuitively support probabilistic models. Probabilistic models, model the uncertainties in the form of an underlying probability distribution. The problem then reduces to a problem of learning the relevant dependency parameters, computing the probability distribution and inferencing using the distribution. The probabilistic techniques fall largely under the paradigm of probabilistic graphical models. Graphical models are very efficient and intuitive frameworks for building probabilistic models for a set of random variables which represent a particular domain. Quoted from (Jordan, 2004)-

> Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity.

Graphical models combine ideas from graph theory and probability theory in an elegant manner to represent complex systems. Efficient training and inference procedures make them a popular choice for many problem domains. They provide a powerful yet flexible framework for representing and manipulating *global* proba-

bility distributions defined by relatively *local* constraints. The nodes of the graph represent the random variables and the edges represent the constraints among the random variables. The undirected graphical models widely used in the vision literature are generally termed as *Random Fields*. Existing work on context modeling using probabilistic graphical models is detailed in Chapter 2.

1.6 Thesis Contribution

The work builds upon the conditional random fields, introducing following new ideas :

- A tree like graph structure is introduced for capturing long range dependencies.
- A single and unified framework is used for solving various image modeling problems like image labeling and object detection.
- The induced graph structure is a tree which allows inference in time linear in number of nodes.

1.7 Thesis Outline

This thesis is organized as follows

Chapter 2: Graphical Models- This section gives an insight on the mathematical and theoretical background of graphical models. The chapter discusses about various kinds of graphical models and their learning methods.

Chapter 3: Literature Review- This section discusses about the models and techniques proposed in the vision literature based on graphical models. The meth-

ods are classified on the basis of their generative and discriminative nature. Some results based on the existing techniques are also discussed.

Chapter 4: Tree Structured Conditional Random Field - This chapter discusses in detail about the hierarchical conditional random fields as proposed in this thesis. The potential functions used, parameter learning methods and the inference methods for the model are described in detail.

Chapter 5: Application of TCRF - This chapter details about the experimental evaluation of the proposed scheme. Results obtained on two different kind of image modeling tasks have been presented. The statistical evaluation is shown with confusion matrices and the classification accuracy tables. A comparative study with existing methods is also done.

Chapter 6: Conclusions and Discussions - This chapter concludes the thesis with the contribution of the thesis, the issues unsolved with model and the future work.

CHAPTER 2

Graphical Models

As quoted from (Jordan, 2004),

> Graphical Models provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering - uncertainty and complexity. Fundamental to the idea of a graphical model is the notion of modularity - a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.

In simpler terms, the nodes of the graph represent the random variables and the edges model the soft constraints among those nodes. In most of the problem domains there are various factors affecting the values of other factors, graphical models become an intuitive tool for modeling those interactions.

Graphical models are broadly studied under two categories on the basis of their graph structure *directed* vs *non directed*.

2.1 Directed Vs Undirected Graphical Models

2.1.1 Directed Graphical Models (DGM)

Directed graphical models are directed graphs which assume that the observed random variables have been produced by a causal latent process. Figure 2.1 shows an example of an arbitrary directed graph. It is commonly known as *Belief net/Bayesian network*. These are directed acyclic graphs (DAG) defined as $G = (\mathbf{V}, \mathbf{E})$, where every node $v \in \mathbf{V}$ is in one to one correspondence with a random variable $X_v \in \mathbf{X}$. The graph structure indicates direct dependencies among random variables. Any two nodes that are not in a descendant/ancestor relationship are conditionally independent given the values of their parents.

Independence Assumption

Every graphical model encodes a set of independence assumptions among the variables present in its graph structure. For a directed graphical model it can be clearly stated as "A node is conditionally independent of all the other nodes given its Markov blanket."

$$P(A|mb(A), B) = P(A|mb(A)) \tag{2.1}$$

The *Markov Blanket* for a node A in a directed graphical model is the set of nodes $mb(A)$, composed of $A's$ parents, its children and its children's parents.

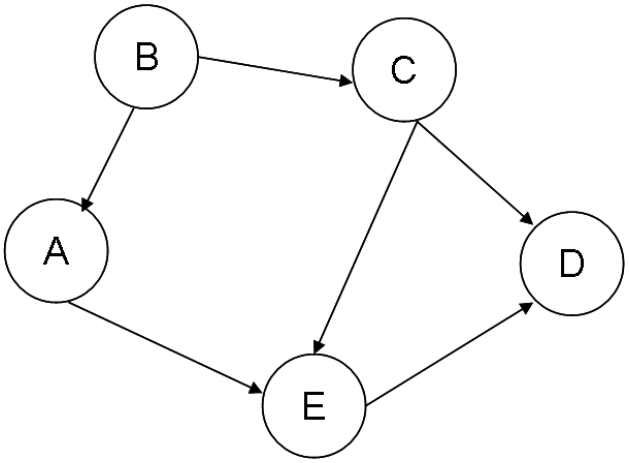


Fig. 2.1: A directed graph showing direct relationship among random variables.

Bayesian Belief Network

General nature of the DGM can be understood by an example of a Bayesian Belief Network.

A Bayesian network is a directed acyclic graph whose

- nodes represent random variables,
- arcs represent statistical dependence relations among the variables and local probability distributions for each variable given values of its parents.

If there is an arc from node A to another node B, then variable B depends directly on variable A, and A is called a parent of B. If for each variable X_i, $i \in 1, ..., N$, the set of parent variables is denoted by $pa(X_i)$, then the *joint distribution* of the variables is product of the local distributions.

$$p(X_1, X_2, \cdots, X_n) = \prod_i p(X_i|\mathbf{pa}(X_i)) \tag{2.2}$$

If X_i has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the variable represented by a node is observed, then

the node is said to be an evidence node. Figure 2.2 shows an example of a belief network and its conditional probability tables. The joint probability of the graph can be calculated as,

$$P(Grasswet, Sprinkler, Rain) = P(Grasswet|Sprinkler, Rain) * P(Sprinkler|Rain)Pr(Rain)$$

This example of bayesian network gives an intuition on how dependency is modeled using directed graphical models.

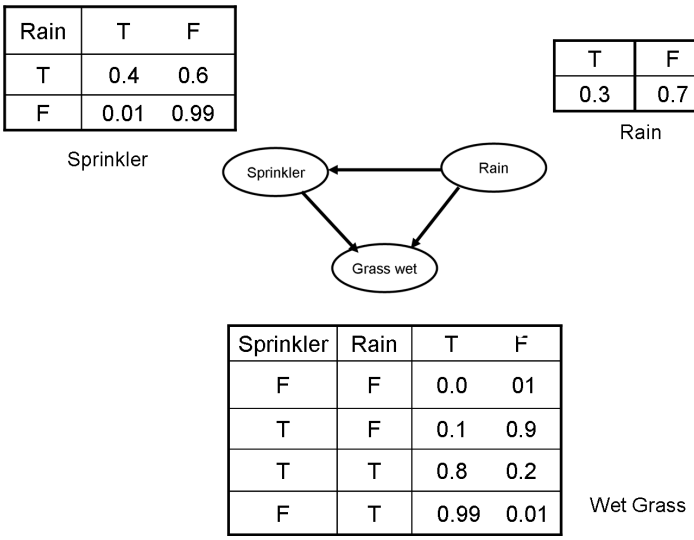


Fig. 2.2: Bayesian belief network showing that wet grass can be caused either by rain or sprinkler, and sprinkler can be caused by rain.

2.1.2 Undirected Graphical Models (UGM)

Undirected Graphical Models (commonly called as Random Fields) can be defined as $G = (\mathbf{V}, \mathbf{E})$, where every node $v \epsilon \mathbf{V}$ is in one to one correspondence with a random variable $X_v \epsilon \mathbf{X}$ and every directed edge $(u, v) \epsilon \mathbf{E}$ can be interpreted as "X_u effects X_v" and vice-versa. Domains, where the random variables do not have explicit causal dependence can be efficiently modeled using undirected graphical

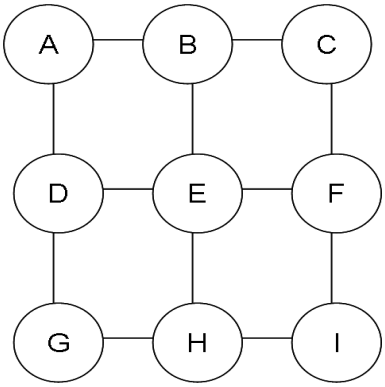


Fig. 2.3: An example of undirected graphical model. This undirected graphical model can be also visualized as an image lattice.

models. For example an image lattice can be intuitively expressed as an undirected graph as shown in Figure 2.3.

Independence Assumption

The independence assumption encoded by an undirected graphical models can be clearly stated as "A node is conditionally independent of all the nodes in the graph given its neighbours $\mathcal{N}$."

Probability Distribution

The undirected nature of the graph eliminates the possibility of decomposing the joint distribution into the product of conditional probabilities of the nodes. The probability distribution of the undirected models is derived from the *Hammersely-Clifford Theorem*. Now, we discuss some of the basic definitions required to derive the probability distribution for UGM.

Definition 1 *A set of nodes C in a graph G constitute a* ***clique*** *if all the distinct nodes in the set are neighbors of each other. A set of nodes C is called a* ***maximal***

clique *if C is a clique and there is no other clique D that strictly contains C.*

Definition 2 *Let G be a finite graph. A* ***Gibbs distribution*** *w.r.t G is a probability mass function which can be expressed as,*

$$p(\boldsymbol{X}) = \prod_{c \in C} \phi_c(X_c) \tag{2.3}$$

where C is the set of maximal cliques of the graph G, X_c corresponds to the set of random variables present in the clique c and $\phi_c(X_c)$ is a real valued function known as the potential function.

In 1968 Hammersley and Clifford came up with the following theorem

Theorem 1 *[HAMMERSLEY-CLIFFORD] Suppose that $X = (X_1, X_2, \ldots, X_n)$ has a positive joint probability mass function. X is a Markov Random Field on G if and only if X has a Gibbs distribution w.r.t G.*

From the above theorem it is clear that the joint probability distribution of a Markov random field is equal to a Gibbs distribution w.r.t to the graph structure corresponding to the particular field. Hence, we conclude here that the probability distribution for an undirected graphical model is modeled as shown in equation (2.3).

Application of an undirected graphical model can be illustrated using a simple image denoising example (Besag, 1974; Bishop, 2006). Let us assume that an array of binary pixels values $x_i \in -1, +1$ represent the observed noisy image, where the index $i = 1, 2, \ldots D$ runs over all the pixels. Here we make an assumption that the

unknown noise-free image is described by binary pixel value $y_i \in -1, +1$. Given the noisy image, the aim is to recover the original noise-free image. The undirected graphical model representation of this problem is shown in Figure 3.1. This graph has two types of cliques. The energy associated with the cliques of the form x_i, y_i expresses the correlation between these variables. Let the energy function be of the form $-\eta x_i y_i$, where η is a positive constant. So, with this energy function we have lower energy(higher probability) when both the variables have the same sign and vice versa when they have opposite signs.

The other clique is of the form y_i, y_j, where i and j are indices of neighboring pixels. A similar energy function $-\beta y_i y_j$ can be used to represent this clique.

The complete energy function for the model can be written as

$$E(x, y) = h \sum_i y_i - \beta \sum_{\{i,j\}} y_i y_j - \eta \sum_i x_i y_j \tag{2.4}$$

which defines a joint distribution over x and y given by

$$p(x, y) = \frac{1}{Z} \exp\{-E(x, y)\} \tag{2.5}$$

an extra term hy_i is added for each pixel i in the noise free image, this takes care of biasing the model towards pixel values that have one particular sign in preference to the other. For the purpose of image restoration, we wish to find an image y having a high probability (ideally maximum probability). We use a simple iterative technique called *iterated conditional modes*(ICM) (Kittler and Föglein, 1984). Here, we first initialize the variables y_i, which is done by simply setting

the $y_i = x_i$ for all i. Then, one node y_j is taken at a time and the total energy is evaluated for the two possible states $y_j = +1$ and $y_j = -1$, keeping all other node variables fixed, and set y_j to whichever state has the lower energy. This will either leave the probability unchanged or will increase it. This process is followed for every site iteratively. For this example, the parameters are fixed to $\beta = 1.0, \eta = 2.1$ and $h = 0$. ICM is run until convergence leading to the denoised image. The above given example illustrates a simple case where undirected graphical models find its application.

This section elaborates on how the directed and undirected graphical models encode the dependency relationship and which model would be more appropriate in a particular kind of scenario, i.e: a UGM is intuitively a better representation of an image lattice. Graphical models can also be categorized as *generative* and *discriminative* models, based on the way they model their probability distribution. The next section examines these two models in detail.

2.2 Generative vs Discriminative

There is another popular dimension along which the class of graphical models is divided. This division corresponds to the difference in the modeling power of different models. The kind of stochastic processes which we are interested in typically contain two kinds variables $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{X}$ corresponds to the observations which we gather from the data. They are also referred to as the visible variables. $\mathbf{Y}$ refers to the variables whose state we want to predict using the given observations. They are also referred to as the hidden or latent variables. The problem of

predicting $\mathbf{Y}$ given $\mathbf{X}$ is equivalent to modeling $p(\mathbf{Y} \mid \mathbf{X})$. There are two ways of doing this

Generative Models - Create a model for $p(\mathbf{X}, \mathbf{Y})$. Then $p(\mathbf{Y} \mid \mathbf{X}) = \frac{p(\mathbf{X},\mathbf{Y})}{p(\mathbf{X})} \propto p(\mathbf{X}, \mathbf{Y})$. Such models are called generative models. Generative models are particularly useful when we want to create a model of the source which generates these observations. There are many reasons to do this, one of them is that getting to work with the actual source might not be practically feasible so we can use the model to simulate its effect. Second, they perform well in case of incomplete data as the model is inherently capable of generating the model distribution. However, this additional power has a trade off with the models flexibility in incorporating arbitrary dependence in the observed data. This assumption makes generative models inappropriate for the classification problems. Markov Random Fields are one of the most widely used generative models in the vision community, detailed in section 2.2.1.

Discriminative Models - Another way is to directly create a model of $p(\mathbf{Y} \mid \mathbf{X})$. Such models are called conditional or discriminative models. Though conditional models do not make any assumptions regarding the observations, they cannot be used to simulate the source i.e. the model cannot be used to generate the observations. Classification problems can be tractably handled using discriminative models. The modeling power of discriminative models is detailed in section 2.2.2.

2.2.1 Markov Random Field (MRF)

Markov Random Field theory (Li, 2001) is a branch of probability theory for analyzing the spatial or contextual dependencies of physical phenomena. MRF, a generative model is the state of art for image modeling. It is used in visual labeling to establish distribution of interacting labels. Some notations to understand MRF modeling are described below.

Random Field (Winkler, 1995)

Let S be a finite index set - the set of *sites* or pixels or locations. For every site $s \in S$ there is a (finite) space $\mathbf{Y}_s$ of *states* y_s. The space of *configurations* $y = (y_s)_{s \in S}$ is the product $\mathbf{Y} = \prod_{s \in S} \mathbf{Y}_s$. We consider *probability measures* or *distributions* on $\mathbf{Y}$; such that $P(y) \geq 0$ and $\sum_{y \in \mathbf{Y}} P(y) = 1$.
Subsets $E \subset \mathbf{Y}$ are called *events*; the probability of an event E is given by $P(E) = \sum_{y \in E} P(y)$. A strictly positive probability measure P on $\mathbf{Y}$, with $P(y) > 0$ for every $y \in \mathbf{Y}$, is called a *random field*.

Definition 3 *(Winkler, 1995) A collection $\delta = \{\delta\{s\} : s \in S\}$ of sets is called* ***neighbourhood system****, if $s \notin \delta\{s\}$ and $s \in \delta\{t\}$ if and only if $t \in \delta\{s\}$. The sites $t \in \delta\{s\}$ are called* ***neighbours*** *of s. A subset C of S is a* ***clique*** *if any two different elements of C are neighbours. The set of cliques will be denoted by C.*

Definition 4 *(Winkler, 1995) A random field is a* ***Markov Random Field*** *with respect to the neighbourhood system δ if for all $y \in \boldsymbol{Y}$,*

$$P(X_s = x_s \mid X_t = x_t, t \neq s) = P(Y_s = y_s \mid Y_t = y_t, t \in \delta\{s\}) \tag{2.6}$$

Note, the neighbourhood system can be multi-dimensional. According to the Hammersely-Clifford theorem (Besag, 1974), an MRF can equivalently be characterized by a Gibbs distribution. Thus,

$$P(\mathbf{Y}) = \frac{1}{Z} \exp(-U(\mathbf{Y})) \tag{2.7}$$

where $Z = \sum_{x \in (Y)} \exp(-U(\mathbf{Y}))$ is a normalizing constant, also called the partition function, and U($\mathbf{Y}$) is an *energy function* of the form $U(\mathbf{Y}) = \sum_{c \in C} V_c(\mathbf{Y})$ which is a sum of clique potentials $V_c(Y)$ over all possible cliques C.

Chapter 3 addresses some specific image models based on MRF graph structure.

2.2.2 Conditional Random Field (CRF)

Conditional Random Fields (Lafferty *et al.*, 2001) are conditional probabilistic sequence models, however, rather than being directed graphical models, CRFs are undirected graphical models, belonging to the family of discriminative models. This allows the specification of a single joint probability distribution over the entire label sequence given the observation sequence, rather than defining per-

state distributions over the next states given the current state. The conditional nature of the distribution over label sequences allows CRFs to model real-world data in which the conditional probability of a label sequence can depend on non-independent, interacting features of the observation sequence. In addition to this, the exponential nature of the distribution chosen by Lafferty *et al.* (2001) enables features of different states to be traded off against each other, weighting some states in a sequence as being more important than others. Recently, several researchers (Kumar and Hebert, 2003*a*, 2005; He *et al.*, 2004, 2006) have brought into light the powerful nature of discriminative models, in the domain of computer vision. This section develops a mathematical background for the understanding of CRFs.

Let $\mathbf{X}$ and $\mathbf{Y}$ be jointly distributed random variables ranging over observation sequences to be labeled and their corresponding label sequences respectively. A conditional random field ($\mathbf{X}$,$\mathbf{Y}$) is an undirected graphical model globally conditioned on $\mathbf{X}$, the observation sequence

Definition 5 *(Lafferty* et al.*, 2001) Let* $G = (\boldsymbol{V}, \boldsymbol{E})$ *be a graph such that the set of vertices* $\boldsymbol{V}$ *is in one-to-one correspondence with the set of label sequence* $Y = (Y_v)_{v \in \boldsymbol{V}}$*. Then* $(\boldsymbol{X}, \boldsymbol{Y})$ *is a conditional random field if the random variables* Y_v *when conditioned on* $\boldsymbol{X}$*, obey the following markov property with respect to the graph:*

$$p(Y_v|\boldsymbol{X}, Y_w, w \neq v) = p(Y_v|\boldsymbol{X}, Y_w, (w, v) \in \boldsymbol{E}) \tag{2.8}$$

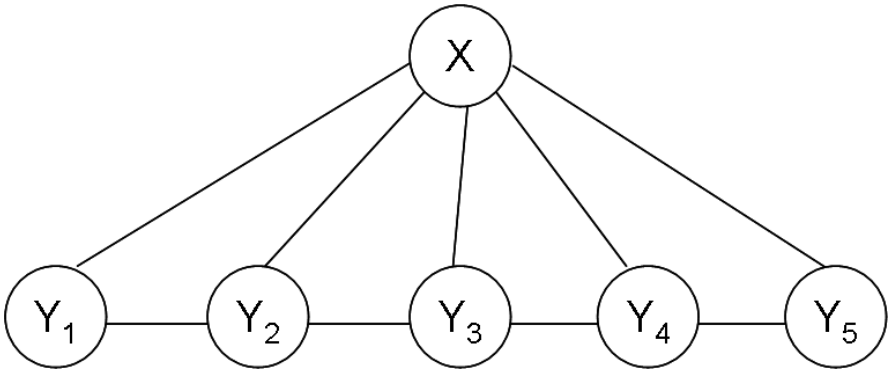


Fig. 2.4: The CRF graph structure. Notice that X is conditioned upon and is not a part of the graph.

The maximum entropy principle

As with any other kind of graphical model, the complete specification of CRFs requires a knowledge of the form of the probability distribution $p(\mathbf{Y} \mid \mathbf{X})$. Since CRFs belong to the class of undirected graphical models, we can apply Hammersley-Clifford theorem to write $p(\mathbf{Y} \mid \mathbf{X})$ as:

$$p(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z} \prod_{c \in C} \phi(c) \tag{2.9}$$

where C is the set of cliques in the graph and $\phi : C \rightarrow \Re$ is called the potential function.

The choice of the potential functions in CRFs is inspired by the principle of maximum entropy (Berger *et al.*, 1996). The principle of maximum entropy states that

The best model of a given set of observations is the one which respects the constraints encoded by the observations but otherwise is as uniform as possible.

Each of the constraints is represented in the form of a binary valued feature function $f(\mathbf{X}, \mathbf{Y})$. The feature function is turned on whenever the corresponding pattern occurs in the training set. The condition that the model should respect the constraint $f(\mathbf{X}, \mathbf{Y})$ can be written as,

$$E_{\tilde{p}}[f(\mathbf{X}, \mathbf{Y})] = E_p[f(\mathbf{X}, \mathbf{Y})] \tag{2.10}$$

Where the L.H.S. is the expectation of the feature w.r.t the data distribution and the R.H.S. is the expectation w.r.t the model distribution (produced by prolonged Gibbs sampling). It can be shown (Berger *et al.*, 1996) that the potential functions which satisfy the above constraint take an exponential form,

$$\phi(\mathbf{X}, \mathbf{Y}) = exp(\lambda f(\mathbf{X}, \mathbf{Y})) \tag{2.11}$$

where, λ are the parameters.

Parameter Estimation for CRFs

The probability distribution for CRF is expressed in terms of the potential functions, which are actually functions of some parameters Θ. Hence, parameter estimation in CRFs, usually involves coming up with an optimal set of parameters $\{\Theta\}$ which maximize the log-likelihood of the training data w.r.t the model distribution.

$$l(\Theta) = \sum_{m=1}^{M} \log P(\mathbf{y}^m | \mathbf{x}^m, \Theta) \tag{2.12}$$

For simplicity lets consider a linear chain CRF. The probability distribution for such a model has the form (Lafferty *et al.*, 2001):

$$p(\mathbf{y}|\mathbf{x},\Theta) = \frac{1}{Z(\mathbf{x})} exp(\sum_{i=1}^{n+1}\sum_{k}\lambda_k f_k(\mathbf{y}_i,\mathbf{y}_{i-1},\mathbf{x}) + \sum_{i=1}^{n}\sum_{k}\mu_k g_k(\mathbf{y}_i,\mathbf{x})) \tag{2.13}$$

where λ_k and μ_k are the model parameters. $f_k()$ and $g_k()$ represent the feature functions defined over different sets of cliques.

One of the first methods proposed for parameter estimation was the Generalized Iterative Scaling(GIS) algorithm.

Generalized Iterative Scaling This method proposed by Lafferty *et al.* (2001) requires that all the features sum up to a constant B. In other words,

$$B = \sum_{i,k} f_k(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x}) + \sum_{i,k} g_k(\mathbf{y}_i,\mathbf{x}) \tag{2.14}$$

In case the features do not sum up to a constant value a global correction feature is added to the model.This feature has the form,

$$b(\mathbf{x},\mathbf{y}) = B - \sum_{i,k} f_k(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x}) + \sum_{i,k} g_k(\mathbf{y}_i,\mathbf{x}) \tag{2.15}$$

The weight update equations turns out to be,

$$\delta\lambda_k = \frac{1}{B} log\frac{\tilde{E}[f_k]}{E[f_k]}, \qquad \delta\mu_k = \frac{1}{B} log\frac{\tilde{E}[g_k]}{E[g_k]} \tag{2.16}$$

where $\tilde{E}[f_k] = \sum_{\mathbf{x},\mathbf{y}} \tilde{p}(\mathbf{x},\mathbf{y}) \sum_{i=2}^{n} f_k(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x})$ is the expectation of the feature f_k w.r.t the empirical distribution of the training data and

$E[f_k] = \sum_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \sum_{i=2}^{n} f_k(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x}) \approx \sum_{\mathbf{x},\mathbf{y}} \tilde{p}(\mathbf{x}) p(\mathbf{y}|\mathbf{x}) \sum_{i=2}^{n} f_k(\mathbf{y}_{i-1},\mathbf{y}_i,\mathbf{x})$

is the expectation of f_k w.r.t the model distribution.

The convergence rate of the above method is very slow due to the presence of the constant B in the denominator term of the weight update equations. If all the features are binary valued then B is equal to the total number of features which is typically very large for most problems (To give the reader an idea, the number of features for a typical NP chunking task on a standard data set is around 3.8 million (Rabiner and Juang, 1993)). Later, Wallach (2002) showed that the GIS algorithm is in general intractable because of the presence of the global correction feature. [1] Recently the use of limited memory quasi-newton (LBFGS) (Nocedal and Wright, 1999) and piecewise training (Sutton and McCallum, 2005) methods has drastically reduced the training time and has made CRFs a practical tool for various classification tasks.[2]

Inference

The general problem of inference can be stated as computing the optimal label sequence $\mathbf{y}^*$ such that,

$$\mathbf{y}^* = argmax_y p(\mathbf{y}|\mathbf{x}) \tag{2.17}$$

For linear-chain CRFs, efficient dynamic-programming algorithms exist for exact inference (Sutton and McCallum, 2006). For more general structures e.g. 2-D CRFs we must resort to approximate methods which fall into two categories

[1] The original CRF paper Lafferty *et al.* (2001) proposes a dynamic programming based approach to efficiently calculate the feature expectations but fails to account for the presence of the global feature

[2] The reader is pointed to the following URL which contains a list of all the significant papers on CRFs. www.inference.phy.cam.ac.uk/hmw26/crf/

1. Markov Chain Monte Carlo(MCMC) methods : Based on approximating a probability distribution by generating samples from the conditional distributions.

2. Variational methods : Based on deterministic approximation to the probability distribution.

However, we do not apply any of the above mentioned method for our proposed model, which, as a special case allows for exact inferencing.

2.3 Chapter Summary

This chapter gives a mathematical overview of the directed and undirected graphical models. It also elaborates on the nature of the generative and discriminative models. The next Chapter (chapter 3) details about some of the relevant work based on MRF and CRFs, and their merits and demerits.

CHAPTER 3

Literature Review

3.1 Methods based on Generative Frameworks

3.1.1 Markov Random Field

Powerful tools like Ising model has been known for long to physicists (Baxter, 1982) and statisticians (Besag, 1974) for stochastic modeling. They were introduced in the computer vision community in a comprehensive way by Geman and Geman (1984) in the form of MRFs. Ising model makes the basis for MRFs. Since then, MRFs have been consistently studied and experimented by several researchers.

MRFs have been widely used for image modeling and synthesis task in the domain of computer vision. MRF incorporate the local contextual constraints in the labeling problems in a very principled manner. Early works of (Cross and Jain, 1983; Geman and Geman, 1984) have really popularized the use of MRF for image modeling. MRFs have been also applied for image synthesis, but we will limit our discussion to image classification application of MRF. MRFs are usually used in a probabilistic generative framework that models the joint distribution of the observed data and their corresponding labels (Li, 2001).

Let $\mathbf{X}$ be the observed data such that $\mathbf{X} = (x_i)_{i \in S}$, x_i is the data from the i^{th}

site, and S is the set of the sites. Let $\mathbf{Y}$ represents the label field at the image sites, where $\mathbf{Y} = y_i$. In the MRF framework, the posterior of the labels over the data distribution is expressed using the Bayes rule as,

$$P(\mathbf{Y}|\mathbf{X}) \propto p(\mathbf{Y}, \mathbf{X}) = P(\mathbf{Y}) * p(\mathbf{X}|\mathbf{Y}) \tag{3.1}$$

For computational tractability, the observation or likelihood model, $p(\mathbf{X} \mid \mathbf{Y})$ is assumed to have a factorized form (Besag, 1974; Xiao *et al.*, 2002; Li, 2001; Feng *et al.*, 2002)

$$p(\mathbf{X}|\mathbf{Y}) = \prod_{i \in S} p(x_i|y_i) \tag{3.2}$$

In MRF formulations of binary classification problems, the label interaction field, $P(\mathbf{Y})$, is commonly assumed to be a homogeneous and isotropic Ising model (or Potts model for multiclass labeling problems) with only pairwise nonzero potentials. If the data likelihood $p(\mathbf{X}|\mathbf{Y})$ is approximated by assuming that the observed data is conditionally independent given the labels, the posterior distribution over labels can be written as per the Hammersely-Clifford theorem,

$$p(\mathbf{X}, \mathbf{Y}) = \frac{1}{Z} \prod_i \phi(x_i, y_i) \prod_{i,j} \psi(y_i, y_j) \tag{3.3}$$

where the partition function $Z = \sum_{x,y} \prod_i \phi(x_i, y_i) \prod_{i,j} \psi(y_i, y_j)$. The potential functions $\phi()$ (data likelihood) and $\psi()$ (prior over labels) are defined below.

$$\begin{aligned} \phi(x_i, y_i) &= \exp(\sum_{i \in S} \log p(x_i|y_i)) \\ \psi(y_i, y_j) &= \exp(\sum_{i \in S} \sum_{j \in \mathcal{N}_i} \beta y_i y_j) \end{aligned}$$

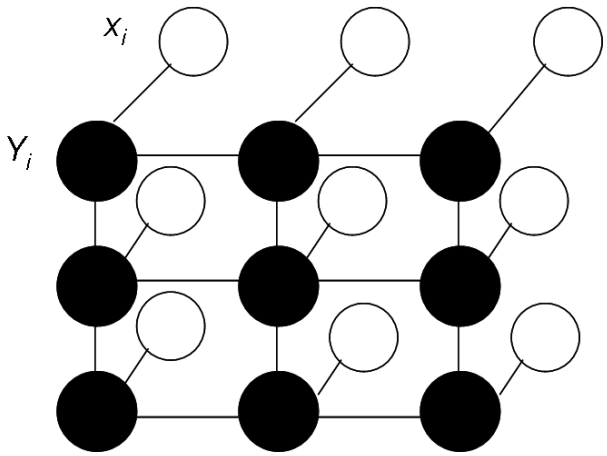


Fig. 3.1: Graph structure of the MRF, here X represents the observed and Y represents the label variable. As can be observed in this structure that no edge connects the observation nodes, since they are assumed to be independent in a MRF model.

β is the interaction parameter of the MRF and $\mathcal{N}_i$ is the set of neighbors of site i.

An illustration of a typical MRF graph structure is shown in Figure 3.1. As can bee seen in this figure, there exists two types of cliques in the graph structure, (x_i, y_i) and (y_i, y_j). These cliques represent two different types of relationships in the graph structure. The dependencies are modeled using the potential functions defined above. $\phi(x_i, y_i)$ represents the relationship between observed nodes and the label nodes. Similarly, $\psi(y_i, y_j)$ represents dependency among the adjacent labels.

Parameter Estimation of MRF

Parameter Estimation for MRFs is a difficult task because of the computational complexity involved in modeling the partition function. Some of the commonly used methods for estimating the parameters are Maximum Likelihood, Pseudo-Likelihood, Coding Method, Mean Field Approximation, least Squares Fit, Markov Chain Monte Carlo Methods etc. The partition function needs to be evaluated for

both training and inference and it makes the MRF model computationally slower and inefficient. The exact computation of the partition function requires a sum of exponential number of terms. Hence, all the methods for parameter estimation and inference are approximate and these approximate methods themselves are computationally time consuming (Geman and Geman, 1984).

Further, most of the researchers, (Bouman and Shapiro, 1994; Kumar and Hebert, 2003*b*; Pieczynski and Tebbache, 2000), noted that conditional independence of data is a very restrictive assumption. In practice, the image sites are not independent given its labels. For example, when solving problems like detecting man made structures in an image, where a long range of image sites define a structure, the data independence assumption could be detrimental.

3.1.2 Tree Structured Bayesian Network (TSBN)

Tree Structured Bayesian Network as proposed by Williams and Feng (1998) modified the MRF model by introducing a hierarchy of layers above the flat architecture of the MRF structure. This hierarchical nature of TSBN readily induces long range dependency. The tree based graph structure is shown in Figure 3.2. The observed data $\mathbf{X}$ is assumed to be generated from an underlying process $\mathbf{Y}$ (label field). $\mathbf{Y}$ is a tree structured belief network. At the highest level (level 0) there is one node Y^0, which has children in level 1. As per Williams and Feng (1998) convention, each node has four children other than leaf nodes, giving rise to a quad tree architecture similar to that of Bouman and Shapiro (1994). Let $\mathcal{L} = \{l_1, l_2, \ldots l_c\}$ be the label set representing the possible labels a pixel can take.

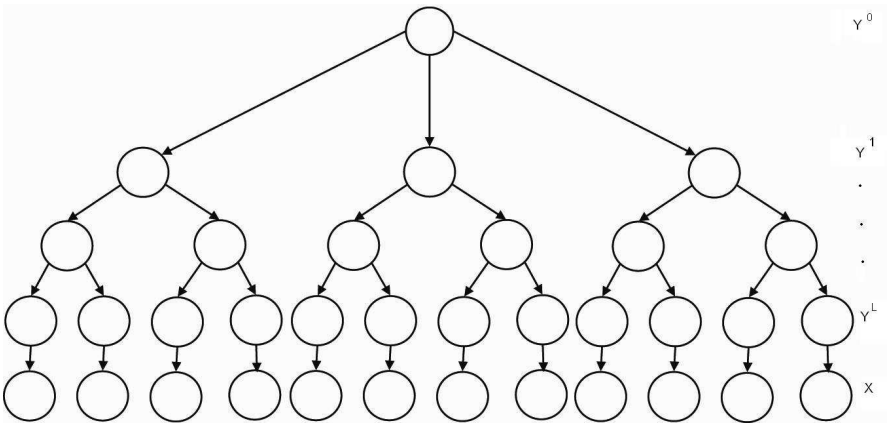


Fig. 3.2: A tree structured belief network

Each y_i-node is a multinomial variable, taking on one of the l_j class label. At the lowest level L of the tree, the nodes $\mathbf{Y}^L$ correspond to the individual image pixel labels. The i^{th} leaf node at level L is denoted by Y_i^L The model for the observation X_i, in each region is that it is generated according to,

$$P(\, X_i \,|\, \mathbf{Y} \,) = P(\, X_i \,|\, Y_i^L \,) \tag{3.4}$$

i.e. X_i depends only on the leaf node Y_i. In addition, it is assumed that the distribution $P(\, X_i \,|\, Y_i^L \,)$ is independent of i.

The overall model can be written as,

$$P(\, \mathbf{Y}, \mathbf{X} \,) = P(\mathbf{Y}) \prod_{i=1}^{K} P(\, X_i \,|\, Y_i^L \,) \tag{3.5}$$

TSBN can be trained using the Expectation-Maximization (EM) algorithm (Williams and Feng, 1998). As known, for the MRF frameworks, exact inference is NP-hard (Barahona, 1982) and approximate inference is computationally expensive. A potential advantage offered by the tree structure of the TSBN over MRF is that, inference can be carried in time linear in the number of nodes using Pearl's

message-passing scheme (Pearl, 1988). But, generative nature of the model introduces prior term, which in turn complicates the computation.

3.2 Methods based on Discriminative Frameworks

3.2.1 Discriminative Random Field (DRF)

The potential application of CRFs was first introduced in the domain of computer vision with a seminal paper by Kumar and Hebert (2003*a*). One of the key feature of DRF is its ability to capture arbitrary dependencies between the observations without resorting to any model approximations. As defined by Kumar and Hebert (2003*a*), assuming only the pairwise potentials to be non zero, the conditional distribution over all the labels $\mathbf{Y}$ given the observation $\mathbf{X}$ can be written as,

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_{i\in S}\phi_i(y_i,\mathbf{X}) + \sum_{i\in S}\sum_{j\in\mathcal{N}_i}\psi_{ij}(y_i,y_j,\mathbf{X})\right) \tag{3.6}$$

where Z is the normalizing constant, ϕ_i and ψ_{ij} are the *association* and the *interaction* potential respectively. The association potential, $\phi(y_i,\mathbf{X})$ can be seen as a measure of how likely a site i will take label y_i given image $\mathbf{X}$, ignoring the effects of other sites in the image. Interaction potential can be seen as a measure of how the labels at the neighboring sites i and j should interact given the observed image $\mathbf{X}$. The graph structure of DRF is illustrated in Figure 3.3.

There are two main differences between the conditional model given in equation (3.6) and the traditional MRF framework given in equation (3.4). First, in

the conditional fields, the association potential at any site is a function of all the observations $\mathbf{X}$ while in MRFs (with the assumption of conditional independence of the data), the association potential is a function of data only at that site, i,e., x_i. Second, the interaction potential for each pair of nodes in MRFs is a function of only labels, while in the conditional models it is a function of labels as well as all the observations $\mathbf{X}$.

The model shows appreciable advantage over MRFs by overcoming the restrictive independence assumption. However, the flat architecture of DRF is incapable of capturing the long range dependencies. Since, only pairwise interactions are taken into account, the model performance on a multiclass problem might deteriorate (the framework has not been applied for a multiclass problem till now).

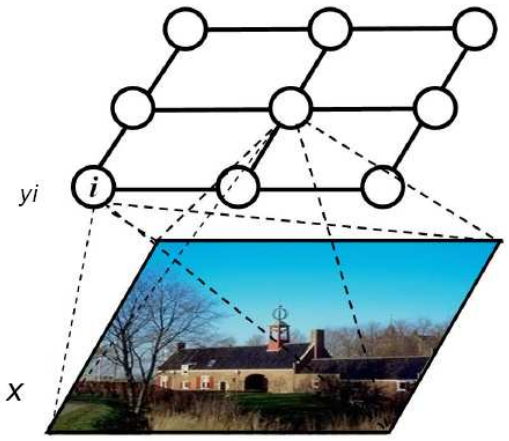


Fig. 3.3: The DRF graph structure. Figure taken from (Kumar, 2005) .

3.2.2 Multiscale Conditional Random Field (mCRF)

Major drawback of modeling images with DRF was its flat graph structure which ignored the importance of the multiscale nature of the image. The multiscale

nature of an image implies that as an image is viewed with an expanding context window, the visual ambiguities gets resolved significantly (refer section 1.4). This idea of capturing multiscale nature of an image was recently proposed by Xuming (He *et al.*, 2004). The model incorporated CRF into a larger framework called product of experts (Hinton, 2002) i.e: combining the outputs of several components. These components/experts work at the different resolution, some components focus on fine features while others on the global structure.

The model architecture defines three different scales of classifiers, namely;

local, regional and *global classifier*.

Local Classifier

Local classifier is trained to learn the local information at each pixel level. This is any statistical classifier which classifies every pixel of an image based on visual properties only. Independently at each site i, the local classifier produces a distribution over label variables y_i given filter outputs x_i within an image patch centered on pixel i:

$$P_C(\,\mathbf{Y}\,|\,\mathbf{X},\lambda\,) = \prod_i P_C(\,y_i\,|\,x_i,\lambda\,) \tag{3.7}$$

where λ are the classifier parameters.

Regional Classifier

Regional Classifier models the label field as a CRF. This component is intended to represent local geometric relationships between objects, such as edges, corners or

T-junctions. They specify the actual objects involved, thus avoiding impossible combinations such as a ground-above-sky border. As shown in Figure 3.4, the smaller (regional) label feature encodes a pattern of ground pixels above water pixels. A collection of such features are learnt from the training data. The whole image is divided into overlapping regions on which these features are defined. Let r index the regions, a index the different regional features within each region, and $j = 1, ..., J$ index the label nodes (sites) within region r. The parameter $w_{a,j}$ connecting hidden regional variables $f_{r,a}$ and label node $y_{r,j}$ specifies preferences for the possible label value of $y_{r,j}$. The probabilistic model describing regional features has the joint distribution,

$$P_R(\mathbf{Y},\mathbf{f}) = \frac{1}{Z} exp\{\sum_{r,a} f_{r,a}(\mathbf{w}_a^T \mathbf{y}_r + \alpha_a)\} \tag{3.8}$$

where, $\mathbf{f} = \{f_{r,a}\}$ represents all the binary hidden regional features, $\mathbf{w}_a = [w_{a,1}, \ldots, w_{a,J}]$, $\mathbf{y}_r = [y_{a,1}, \ldots, y_{a,J}]$ and α_a is a bias term.

Finally, the regional component of this model is formed by marginalizing out the hidden variables in this sub-model: $P_R(\mathbf{Y}) \propto \prod_{r,a} [1 + exp(\mathbf{w}_a^T \mathbf{y}_r)]$

Global Classifier

Global Classifier models the label field as a CRF to capture the coarser aspects of the image. As illustrated in Figure 3.4, the bigger (global) label feature encodes sky pixels at the top of the image, rhino/hippo pixels in the middle, and water pixels near the bottom. Each coarse-resolution global feature has as its domain the label field for the whole image. Let b index the global label patterns encoded

in the parameters $\{u_b\}$ and $\mathbf{g} = \{g_b\}$ be the binary hidden global variables. The label field is divided into non overlapping patches p_m. For each hidden global variable g_b, its connections with the label nodes within patch p_m are assigned a single parameter vector u_{b,p_m}. The global feature model has the joint distribution,

$$P_G(\mathbf{Y},\mathbf{g}) = \frac{1}{Z} exp\{\sum_b g_b(u_b^T \mathbf{Y} + \alpha_b)\} \tag{3.9}$$

where, α_b is the bias term. For this component too, the joint model is marginalized to obtain the global feature component: $P_G(\mathbf{Y}) \propto \prod_b [1 + exp(u_b^T \mathbf{Y})]$.

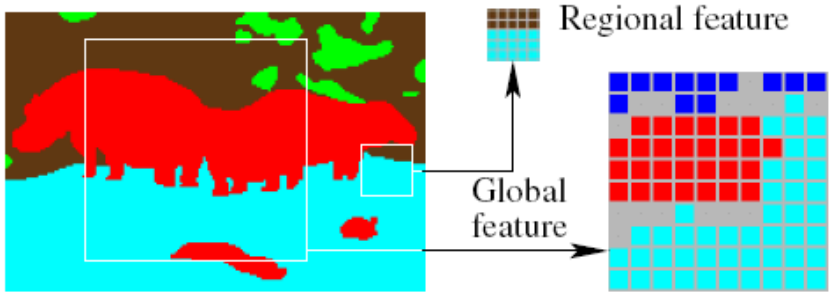


Fig. 3.4: The multiscale framework captures information at different scales. Figure taken from (He *et al.*, 2004).

All the above components are combined using a product of experts (POE) model according to the equation below:

$$P(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z} P_C(\mathbf{Y} \mid \mathbf{X}) P_R(\mathbf{Y}) P_G(\mathbf{Y}) \tag{3.10}$$

here $P_C(\mathbf{Y} \mid \mathbf{X})$ denotes the probability distribution produced by the local classifier while $P_R(\mathbf{Y})$ and $P_G(\mathbf{Y})$ denote the probability distribution produced by the regional and global label features respectively. Figure 3.5 shows some of the labeling results obtained by using mCRF on a landform segmentation problem.

The landform segmentation has been detailed in appendix. As per this approach, the features are incorporated at different scales of context but model is computationally expensive and less generic in nature. The nature of the global and regional features are highly domain specific, hence a new window size of features has to be chosen every time the model has to be used for a new image set. The choice of number of such context encoders is also questionable. Ideally several such classifiers would be needed for better performance but increasing the number of classifiers would explode the parameter space and the model will become computationally intractable. Further, the model assumes the conditional independence of the hidden variables at the global and regional scales which restricts the power of context modeling. Last but not least, the local classifier and the CRF classifiers work independently, i.e. a trained local classifier is used as an input for the CRF. Hence, the interaction among the context encoders is not properly handled.

3.3 Discussion

To summarize, this chapter discusses various models applicable for the image modeling task, built on the probabilistic framework. Generative models have empirically and theoretically been proved to be computationally expensive and slow. They make unwarranted independence assumptions and hence are rather less significant for the classification task (Kumar and Hebert, 2003*a*). DRFs overcome most of the issues faced by MRF for a classification problem by releasing the independence assumption . However, flat architecture of DRF is incapable of capturing long range dependency. mCRF captures context at three different

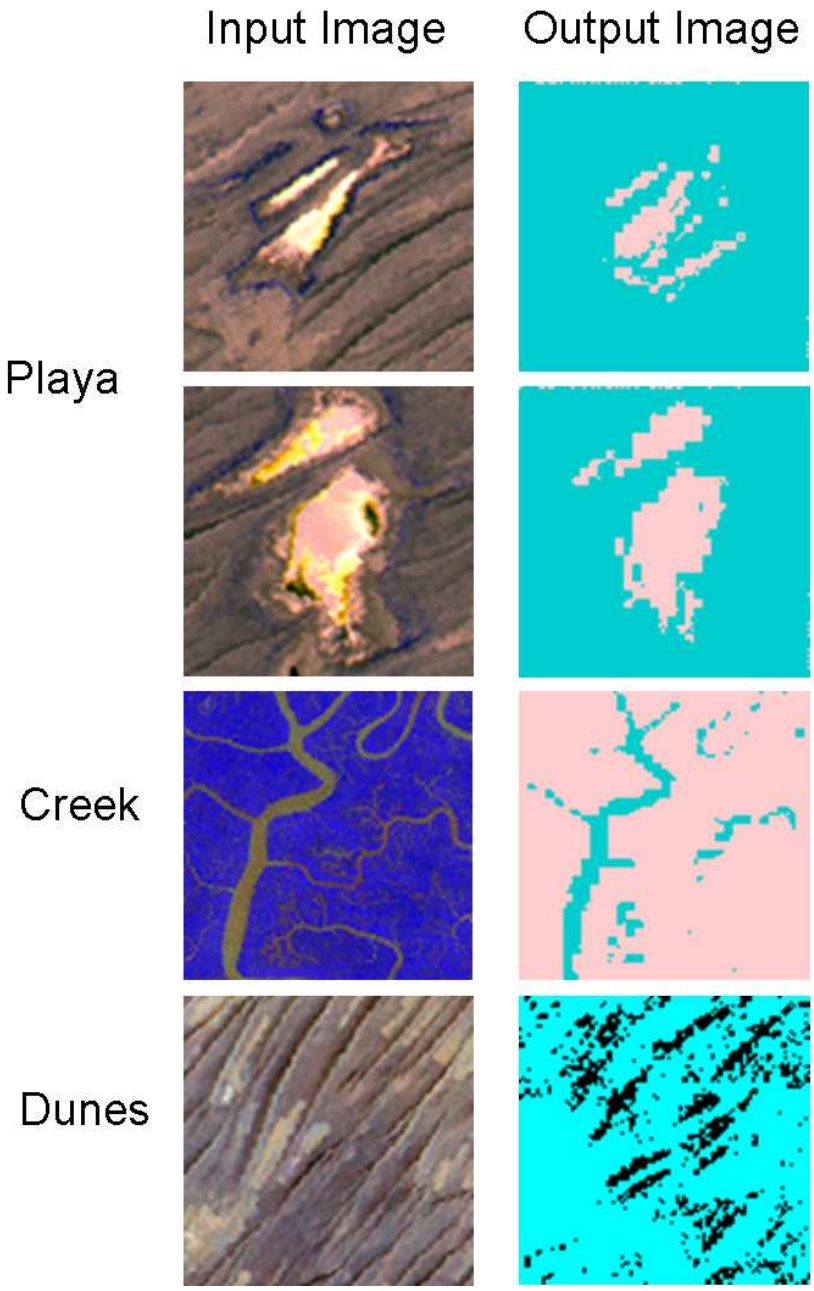


Fig. 3.5: Results obtained by using mCRF model on a landform segmentation problem

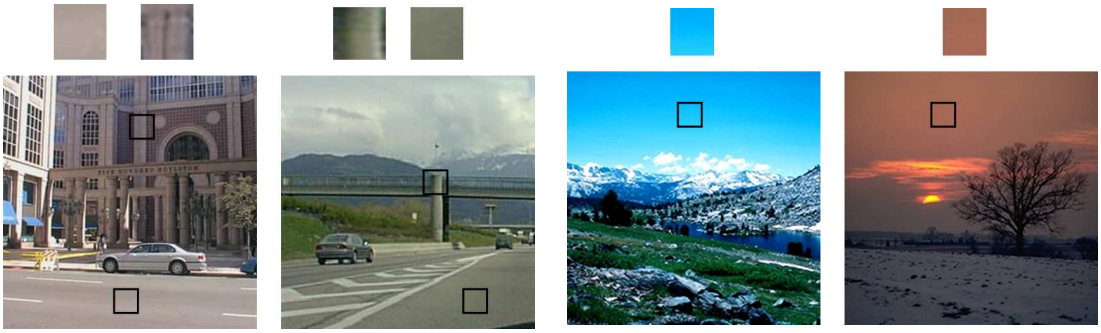
scales but makes the unwarranted independence assumptions about the hidden variables. In the next Chapter (4) we discuss that how are the above mentioned drawbacks are addressed by the proposed model, *Tree Structured Conditional Random Fields*.

CHAPTER 4

Tree Structured Conditional Random Field (TCRF)

4.1 Introduction

In this thesis we present *Tree Structured Conditional Random Fields* for stochastic modeling of images in a hierarchical fashion. Various image modeling tasks like, image labeling, texture segmentation, image restoration, edge detection, texture synthesis, object detection etc have been quiet successfully solved using MRFs and their variants (Li, 2001). However, as discussed in Chapter 3, MRFs are too restrictive for image labeling problems. The framework proposed in this thesis has the advantage of being a discriminative and a multiscale model. The TCRF framework has the ability to incorporate a rich set of interactions among the image sites, which is achieved by inducing a hierarchy of hidden variables over the given label field. The tree like structure of this model can handle more number of scales with almost same number of parameters required for handling much fewer scales in other proposed multiscale models (He *et al.*, 2004) and at the same time permits the use of exact and efficient inference procedures based on belief propagation. The model is generic enough to be applied to different computer vision tasks like, *image labeling* and *object detection*. In Chapter 5, we discuss the results and experimental details on the above mentioned image modeling problems. In



(a) (b)

Fig. 4.1: (a) This figure shows two different classes having similar visual properties. The road and the building features are so similar that its almost impossible to classify them as different objects on the basis of their color properties. The presence of vehicles on the road can be useful for disambiguation. (b) This figure shows two different patches of sky, representing the same class label, but are visually highly variable.

order to understand the requirements of any good image modeling system let us consider an example, shown in Figure 4.1(a). In this figure, building and road; both look very similar as per their visual properties. It is a challenging task to discriminate among the two classes by considering the color features, or even the small neighbourhood properties. In order to resolve the ambiguity of this nature we need some additional object association, like vehicles or humans moving on the road. The extent of the road pixels is quiet wide as compared to that of vehicles or humans, hence in order to capture an association between far apart pixels, it is crucial to have a graph structure capable of capturing context at varying scales. Figure 4.1(b) shows a different example, where two different patches belong to the same class sky, but are highly variant in their visual properties. However, in such cases the presence of other objects, like birds, clouds or sun can be used for disambiguation. Let us consider models like DRF (Kumar and Hebert, 2003a) or MRF (Geman and Geman, 1984) for disambiguating building from road. With these models we would be able to encode utmost the local smoothing of the labels, since these models consider only pairwise potentials (refer to section 3.2.1). As

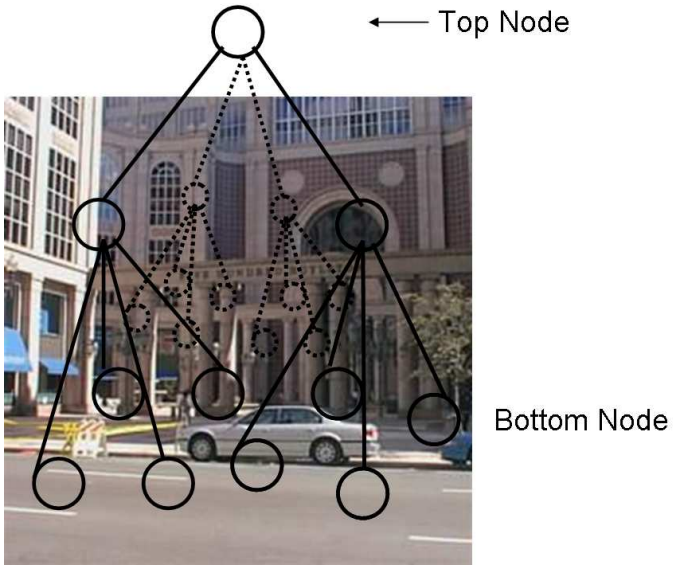


Fig. 4.2: Example showing that how effectively a tree like structure interacts with the far apart image nodes. The information about the class to which a pixel or a patch of a pixel belongs is passed as a message from base nodes to the root node and vice versa. Each region influences the labels of the other regions.

per the flat architecture of these models, mutual influence among the labels of the distant regions is not there. Now, let us assume a tree like structure built above this image as shown in Figure 4.2. The node at the top has an indirect link with all the nodes at the bottom level. The far apart regions have a better interaction because of this kind of structure.

This gives us an intuition of how a hierarchical structure can naturally encode the long range dependencies. Tree structured Bayesian networks proposed by Williams and Feng (1998) also model a tree like structure, but due to the generative nature of Bayesian networks the model tends to make unwarranted independence assumptions about the observed data. The discriminative framework, as discussed in section 2.2.2 resolves the problem of making unwarranted independence assumptions by computing the conditional probability of labels given the observation. Hence, this key feature of discriminative models makes it more

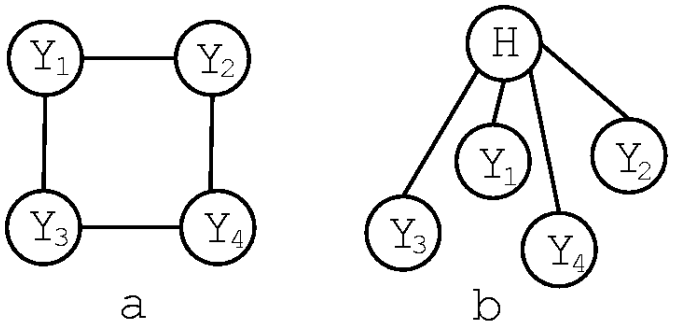


Fig. 4.3: Two different ways of modeling interactions among neighboring pixels.

appropriate in an image labeling problem. In summary, we propose a model, the TCRF, which combines the significant attributes of discriminative and hierarchical models to encode context in a more robust and principled manner.

4.2 TCRF Graph Structure

Consider a 2×2 neighborhood of labels (defined over pixels) as shown in Figure 4.3. One way to model the association between the labels of these pixels is to introduce a weight vector for every edge (Y_i, Y_j) which represents the compatibility between the labels of the nodes Y_i and Y_j (Figure 4.3(a)). An alternative is to introduce a hidden variable H which is connected to all the four nodes. For every value which variable H takes, it induces a probability distribution over the labels of the nodes connected to it (Figure 4.3(b)). The lowest layer or the leaf nodes are the hidden nodes which are observed at the time of training ($\mathbf{Y}$), while the hidden nodes at the higher levels ($\mathbf{H}$) are unobserved during training. Dividing the whole image, and the associated label field, into regions of size $m \times m$ and introducing a hidden variable for each of them (Figure 4.4(a)) gives a layer of hidden variables over the given label field. In such a configuration each label

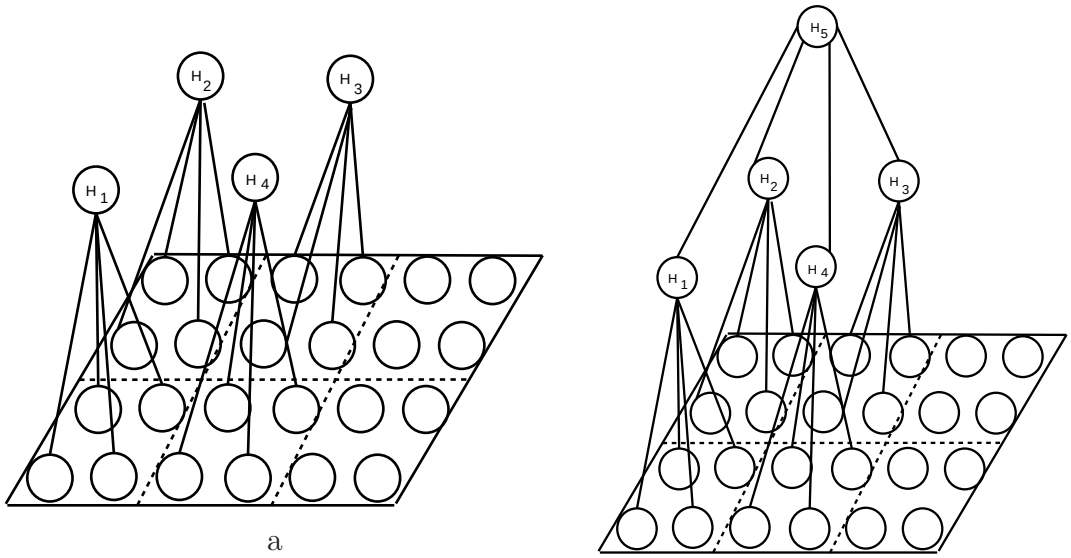


Fig. 4.4: Hidden variable arrangement in the tree structure.

node is associated with a hidden variable in the layer above. Following this, we introduce multiple layers of hidden variables above given label field (Figure 4.4(b)). Each layer of hidden variables tries to capture label relationships at a different level of scale by affecting the values of the hidden variables in the layer below. The main advantage of following such an approach is that long range correlations among non-neighboring pixels can be easily modeled as associations among the hidden variables in the higher layers. Another advantage is that the induced graph structure is a tree (acyclic) which allows inference to be carried out in time linear in the number of nodes (Felzenszwalb and Huttenlocher, 2006).

Formally, let $\mathbf{Y}$ be the set of label nodes, $\mathbf{X}$ the set of observations and $\mathbf{H}$ be the set of hidden variables. We call $(\{\mathbf{Y}, \mathbf{H}\}, \mathbf{X})$ a TCRF if the following conditions hold:

1. There exists a connected acyclic graph $G = (V\,, E)$ whose vertices are in one-to-one correspondence with the variables in $\mathbf{Y} \cup \mathbf{H}$, and whose number of edges $|E| = |V| - 1$.
2. The node set V can be partitioned into subsets $(V_1\,, V_2\,, \ldots, V_L)$ such that

$\bigcup_i V_i = V$, $V_i \cap V_j = \emptyset$ and the vertices in V_L correspond to the variables in $\mathbf{Y}$.

3. For every $(u\,,\,v) \in E$, if $u \in V_k$ then either $v \in V_{k+1}$ or $v \in V_{k-1}$.
4. $deg(v) = 1, \quad \forall v \in V_L$.

The definition of TCRF comprises of a family of tree structured graphs, which are more general in nature. While this is a mathematically rigorous characterization, in practice not all such graphs might be useful. There are several ways in which a tree like graph structure can be defined. The tree could be unbalanced, binary tree, quad tree, or even a tree with varying arity at each level, depending on the image set to be modeled.

In our work, we model a quad tree, by taking the value of m to be equal to 2. The quad tree structure of our model is shown in Figure 4.5. There are several reasons which makes quad tree structure, a more appropriate choice for images. First, ideally one would like to capture context at increasing levels of scale where the scale is increasing uniformly. Second, increasing or decreasing the arity of the tree might either capture redundant and unwanted context, or would miss some of the important context relationship. Hence, we can say that though a quad tree is compromise choice, it also happens to be the most popular method of modeling tree over images (Bouman and Shapiro, 1994; Williams and Feng, 1998). Similar to CRFs, the conditional probability $P(\mathbf{Y}, \mathbf{H} \,|\, \mathbf{X})$ of a TCRF factors into a product of potential functions. We next describe the form of these functions.

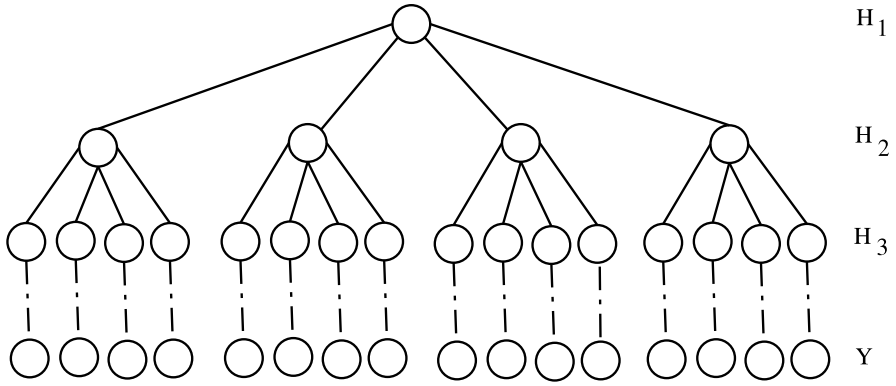


Fig. 4.5: A tree structured conditional random field.

4.3 Potential Functions

Since the underlying graph for a TCRF is a tree, the set of cliques C corresponds to nodes and edges of this graph. We define two kinds of potentials over these cliques:

4.3.1 Local Potential

Local Potential is intended to represent the influence of the observations $\mathbf{X}$ on the label nodes $\mathbf{Y}$, i.e. how likely a site i will take a label l_j given the observation $\mathbf{X}$, (Figure 4.6) . For every $\mathbf{y_i} \in \mathbf{Y}$ this function takes the form $\exp(\Gamma_i(\mathbf{y_i}, \mathbf{X}))$. $\Gamma_i(\mathbf{y_i}, \mathbf{X})$ can be defined as,

$$\Gamma_i(\mathbf{y_i}, \mathbf{X}) = \mathbf{y_i}^T \mathbf{W} \mathbf{f}_i(\mathbf{X}) \tag{4.1}$$

Let $\mathcal{L}$ be the label set and $|\mathcal{L}|$ be the number of classes. Then, $\mathbf{y}_i$ is a binary vector of length $|\mathcal{L}|$, such that all the values are zero, other than the one at the index j, where the values is 1. Then, $\mathbf{f_i}(.) = \{f_1, \ldots, f_{\mathcal{F}}\}$ is a transformation (possibly

non-linear), computed for the i^{th} pixel or a patch of pixel. It is a function which transforms an arbitrary pixel or a patch of pixels, corresponding to a label in the label field, into a feature vector. The raw image pixels are not used directly for training. A set of operations is applied on them to obtain a feature vector (refer to section 4.6). $\mathbf{W}$ is a weight matrix of size $|\mathcal{L}| \times \mathcal{F}$ estimated from the training data. This form is an example of a *Linear Regression* (LR) classifier and is similar to the traditional MRF models where one can use arbitrary local generative classifier to model the unary potential. Another approach is to take $\Gamma_i(\mathbf{y_i}, \mathbf{X}) = \log P(\mathbf{y}_i \,|\, \mathbf{X})$, where $P(\mathbf{y}_i \,|\, \mathbf{X})$ is the probability estimate computed by a separately trained local classifier (He *et al.*, 2006)

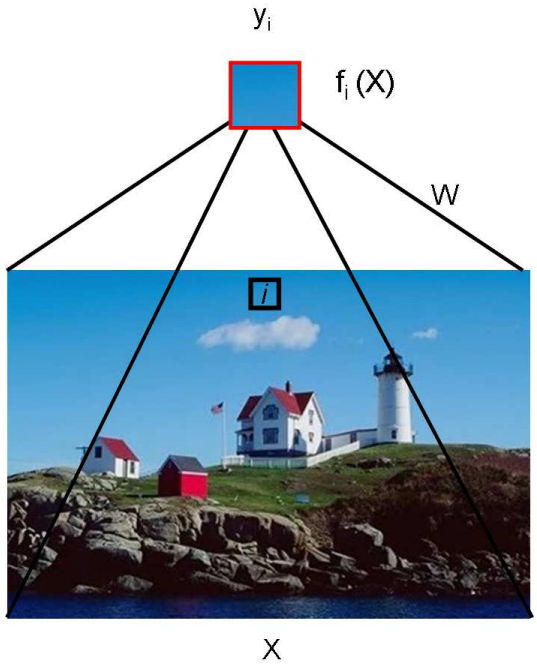


Fig. 4.6: Given a feature vector $f_i(X)$, local potential in a TCRF encodes the probability of a site i getting the label l_j. W represents the weight vector.

4.3.2 Edge Potential

The edge potentials encode the interaction among the hidden nodes and the label nodes at the lowest level, and among hidden nodes at the upper levels. Each hidden node can take any one of the values from the label set $\mathcal{L}$. Edge potential is a measure of how likely, the hidden node i at level t, is to take a value l_k given that the node j takes a value l_r at level $t+1$. Hence it is defined over the edges of the tree as shown in Figure 4.7. Let $\phi^{t,b}$ be a matrix of weights of size $|\mathcal{L}| \times |\mathcal{L}|$ which represents the compatibility between the values of a hidden variable at level t and its b^{th} neighbor at level $t+1$. Compatibility is a measure of how likely a class i occurs with class j. For our quad-tree structure $b = 1, 2, 3$ or 4. Let L denote the number of levels in the tree, with the root node at level 1 and the image sites at level L and let $\mathbf{H}_t$ denote the set of hidden variables present at level t. Then for every edge $(\mathbf{h}_i, \mathbf{y}_j)$ such that $\mathbf{h}_i \in \mathbf{H}_{L-1}$ and $\mathbf{y}_j \in \mathbf{Y}$ the edge potential can be defined as,

$$\Upsilon_{i,j}(\mathbf{h}_i, \mathbf{y}_j) = \exp(\mathbf{h}_i^T \phi^{L-1,b} \mathbf{y}_j). \tag{4.2}$$

In a similar manner the edge potential between the node $\mathbf{h}_i$ at level t and its b^{th} neighbor $\mathbf{h}_j$ at level $t+1$ can be represented as,

$$\Upsilon_{i,j}(\mathbf{h}_i, \mathbf{h}_j) = \exp(\mathbf{h}_i^T \phi^{t,b} \mathbf{h}_j). \tag{4.3}$$

The overall joint class conditional probability distribution for the TCRF model is a product of the potentials defined above, and can be written as,

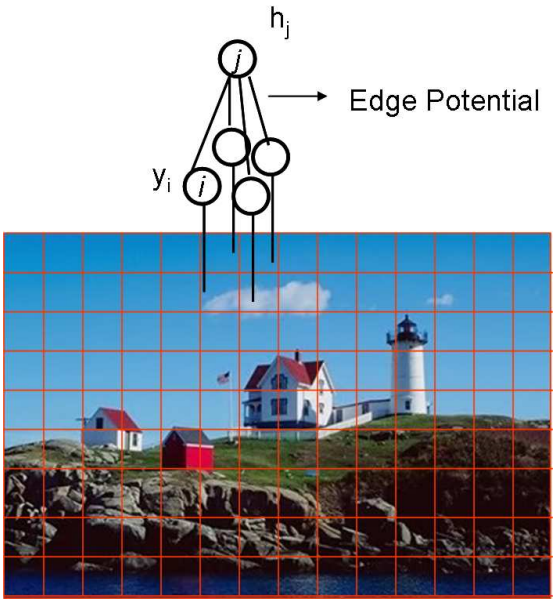


Fig. 4.7: Given the four children, edge potential in a TCRF is a measure of how likely, a hidden node j at level t, is to take a value l_k given that a node i takes a value l_r at level $t+1$

.

$$P(\mathbf{Y},\mathbf{H}|\mathbf{X}) = \frac{1}{Z}\exp\Big(\sum_{\mathbf{y}_i\in\mathbf{Y}}\Gamma(\mathbf{y}_i,\mathbf{X}) + \sum_{\mathbf{y}_j\in\mathbf{Y}}\sum_{\substack{\mathbf{h}_i\in\mathbf{H}_{L-1}\\(\mathbf{y}_j,\mathbf{h}_i)\in E}}\Upsilon_{i,j}(\mathbf{h}_i,\mathbf{y}_j) + \sum_{t=1}^{L-2}\sum_{\mathbf{h}_i\in\mathbf{H}_t}\sum_{\substack{\mathbf{h}_j\in\mathbf{H}_{t+1}\\(\mathbf{h}_i,\mathbf{h}_j)\in E}}\Upsilon_{i,j}(\mathbf{h}_i,\mathbf{h}_j)\Big) \tag{4.4}$$

$$P(\mathbf{Y},\mathbf{H}\,|\,\mathbf{X}) \propto \exp\Big(\sum_{\mathbf{y}_i\in\mathbf{Y}}\mathbf{y_i}^T\mathbf{W}\mathbf{f}_i(\mathbf{X}) + \sum_{\mathbf{y}_j\in\mathbf{Y}}\sum_{\substack{\mathbf{h}_i\in\mathbf{H}_{L-1}\\(\mathbf{y}_j,\mathbf{h}_i)\in E}}\mathbf{h}_i^T\phi^{L-1,b}\mathbf{y}_j + \sum_{t=1}^{L-2}\sum_{\mathbf{h}_i\in\mathbf{H}_t}\sum_{\substack{\mathbf{h}_j\in\mathbf{H}_{t+1}\\(\mathbf{h}_i,\mathbf{h}_j)\in E}}\mathbf{h}_i^T\phi^{t,b}\mathbf{h}_j\Big) \tag{4.5}$$

4.4 Parameter Estimation

In order to efficiently exploit the model properties, choice of the most appropriate parameter learning algorithm is an important step. Given the set of training images, $\mathcal{T} = \{(\mathbf{Y}^1, \mathbf{X}^1) \ldots (\mathbf{Y}^M, \mathbf{X}^M)\}$, the aim is to estimate the optimal set of parameters $\boldsymbol{\Theta} = \{\mathbf{W}, \boldsymbol{\Phi}\}$. Commonly, the optimal set of parameters $\boldsymbol{\Theta}$ are obtained by maximizing the log-likelihood of the training data w.r.t the model distribution (equation 2.12). Exact maximum likelihood parameter learning is feasible for $1D$ sequential CRFs proposed by Lafferty *et al.* (2001), because the induced graph structure does not contain any loops. The loop-free graphs allow easy computation of the partition function using Dynamic Programming. Several efficient techniques have been proposed to learn parameters in these models, e.g., iterative scaling (Gentle, 1997; Darroch and Ratcliff, 1972; Lafferty *et al.*, 2001), quasi-Newton methods (Sarawagi and Cohen, 2005; Sha and Pereira, 2003), conjugate gradient (Wallach, 2002) and gradient boosting (Dietterich *et al.*, 2004). TCRF also proposes a loop free graph structure, but still it does not support the use of exact parameter learning algorithm due to the presence of hidden variables. For our model, we have performed training using the Contrastive Divergence (CD) (Hinton, 2002) algorithm, which is an approximate Maximum Likelihood (ML) estimate. In the next section we discuss different parameter learning algorithms applicable for TCRF.

4.4.1 Maximum Likelihood Parameter Learning

Given M i.i.d labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood,

$$l(\mathbf{\Theta}) = \sum_{m=1}^{M} \log P(\mathbf{y}^m | \mathbf{X}^m, \mathbf{\Theta}) \tag{4.6}$$

with the presence of hidden variables, equation 4.6 can be written as,

$$l(\mathbf{\Theta}) = \sum_{m=1}^{M} \sum_{\mathbf{H}} \log P(\mathbf{y}^m, h | \mathbf{X}^m, \mathbf{\Theta}) \tag{4.7}$$

i.e.,

$$\hat{\Theta} = \arg\max_{\Theta} l(\mathbf{\Theta}) \tag{4.8}$$

$$\hat{\Theta} = \arg\max_{\Theta} \sum_{m=1}^{M} \sum_{\mathbf{H}} \{\mathbb{E}(\mathbf{y}, h; \mathbf{X}, \mathbf{\Theta}) - \log(Z^m)\} \tag{4.9}$$

where $\sum_{\mathbf{H}}$ means it is summed over all the possible values of $\mathbf{H}$. $\mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{x}, \Theta)$ is the summation of the potential functions shown in equation 4.4, it can be also termed as the energy function. The partition function Z^m for the m^{th} image is $Z^m = \sum_{\mathbf{Y},\mathbf{H}} P(\mathbf{y}, \mathbf{h} | \mathbf{X}^m)$. Note here that due to the presence of the hidden variables, we cannot directly apply the ML estimate. However, it can be approximated by Expectation Maximization (EM).

The E-step in EM is to take expectation of the log likelihood, which for our

model, can be written as (for simplicity we have removed the superscript m),

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\mathbf{old}}) = \sum_{m=1}^{M} \sum_{\mathbf{H}} P(h|\mathbf{y}, \mathbf{X}, \boldsymbol{\Theta}^{old}) \log P(\mathbf{y}, h|\mathbf{X}, \boldsymbol{\Theta}) \quad (4.10)$$

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{\mathbf{old}}) = \sum_{m=1}^{M} \sum_{\mathbf{H}} P(h|\mathbf{y}, \mathbf{X}, \boldsymbol{\Theta}^{old}) \left[\mathbb{E}(\mathbf{y}, h; \mathbf{X}, \boldsymbol{\Theta}) - \log(Z)\right] \quad (4.11)$$

It should be noted here that the M-step in EM is to maximize the expectation, i.e., equation 4.10. There are two reasons which make EM inappropriate for TCRF training. First, the Z term is not a constant, it is a summation of exponential terms over all the labels and hence evaluating Z is not algebraically tractable. Second, the partial derivative of equation 4.10 with respect to each of our model parameters produces nonlinear equations, dependent on other parameters, which is not a closed form solution, so optimal parameters cannot be computed from this equation. Hence, it can be concluded here that EM proves to be computationally an expensive choice for TCRF. The next section details how Contrastive Divergence is a feasible choice for training TCRF.

4.4.2 Contrastive Divergence (CD)

CD is an approximate ML estimate proposed by (Hinton, 2002). CD estimates the model parameters by estimating the energy gradient of our model. The joint conditional class distribution (equation 4.4) of the model can be rewritten as below

$$P(\mathbf{Y}, \mathbf{H}|\mathbf{X}, \boldsymbol{\Theta}) = \frac{1}{Z} \exp(\mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}, \boldsymbol{\Theta})) \quad (4.12)$$

$$Z(\boldsymbol{\Theta}) = \sum_{\mathbf{Y}} \exp(\mathbb{E}(y, \mathbf{h}; \mathbf{X}, \boldsymbol{\Theta})) \quad (4.13)$$

On differentiating the log likelihood of the parameters we obtain,

$$\frac{\partial \log P(\mathbf{Y}, \mathbf{H}|\mathbf{X}, \mathbf{\Theta})}{\partial \mathbf{\Theta}} = \sum_m \frac{\partial}{\partial \mathbf{\Theta}} \log \left[\frac{1}{Z(\mathbf{\Theta})} \exp(\mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}^m, \mathbf{\Theta})) \right] \tag{4.14}$$

$$= \sum_m \left[\frac{\partial \mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}^m, \mathbf{\Theta})}{\partial \mathbf{\Theta}} - \frac{1}{Z(\mathbf{\Theta})} \frac{\partial Z(\mathbf{\Theta})}{\partial \mathbf{\Theta}} \right] \tag{4.15}$$

$$= M \left[\left\langle \frac{\partial \mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}, \mathbf{\Theta})}{\partial \mathbf{\Theta}} \right\rangle_{P(\mathbf{Y}|\mathbf{X}, \mathbf{\Theta})} - \left\langle \frac{\partial \mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}, \mathbf{\Theta})}{\partial \mathbf{\Theta}} \right\rangle_m \right] \tag{4.16}$$

The maximum likelihood weight update for this density can then be rewritten in simpler notation as,

$$\Delta \mathbf{\Theta} \propto \langle g \rangle_\infty - \langle g \rangle_0 \tag{4.17}$$

where $\langle g \rangle_0$ is the average of the gradient $g = \frac{\partial \mathbb{E}(\mathbf{y}, \mathbf{h}; \mathbf{X}, \mathbf{\Theta})}{\partial \mathbf{\Theta}}$ evaluated at the data points and $\langle g \rangle_\infty$ is the average of the gradient for the observed points drawn from the proposed distribution $P(\mathbf{Y}, \mathbf{H}|\mathbf{X}, \mathbf{\Theta})$. Samples cannot be drawn directly from this distribution as we do not know the partition function. Though, we can approximate them using MCMC sampling, but the issue is that we have to run several MCMC iterations before we can reach equilibrium. It is computationally very expensive to run so many iterations for every sample to be approximated. Empirically, Hinton (2002) found that only one cycle of MCMC is sufficient for the algorithm to converge to the ML estimate.

With the above background, the weight update equation can be written as,

$$\Delta \mathbf{\Theta} \propto \langle g \rangle_1 - \langle g \rangle_0 \tag{4.18}$$

This difference is known as contrastive divergence. Let P be the equilibrium distribution over the visible variables, produced by prolonged Gibbs sampling from the discriminative model and Q^0 be the data distribution of label variables, then it can be proven (Hinton, 2002) that,

$$\frac{\partial(Q^0||P - Q^1||P)}{\partial\mathbf{\Theta}} \propto \langle g\rangle_0 - \langle g\rangle_1 \tag{4.19}$$

where $Q \parallel P = \sum_{\mathbf{y}\in\mathbf{Y}} Q(\mathbf{y}) \log \frac{Q(\mathbf{y})}{P(\mathbf{y})}$ is the Kullback-Leibler divergence between Q & P. Formally, CD is an approximate learning method which minimizes a different objective function (Hinton, 2002) i.e. contrastive divergence,

$$\mathcal{D} = Q^0 \parallel P - Q^1 \parallel P \tag{4.20}$$

Q^1 is the distribution defined by the one step reconstruction of the training data vectors. The one step reconstruction is obtained as given below (Hinton, 2002)

1. Pick a data vector, d, from the distribution of the data Q^0.
2. Compute, for each component (potential function), the posterior probability distribution over its latent (i.e., hidden) variables given the data vector, d.
3. Pick a value for each latent variable from its posterior distribution.
4. Given the chosen values of all the latent variables, compute the conditional distribution over all the visible variables.
5. Pick a value for each visible variable from the conditional distribution. These values constitute the reconstructed data vector, $\hat{d}$

Let $\mathbf{Y}^1$ be the one step reconstruction of training set $\mathbf{Y}^0$. The weight update

equations specific to our model are,

$$\Delta w_{uv} = \eta[\sum_{\mathbf{y}_i \in \mathbf{Y}^0} y_i^u f_i^v(\mathbf{X}) - \sum_{\mathbf{y}_i \in \mathbf{Y}^1} y_i^u f_i^v(\mathbf{X})] \tag{4.21}$$

$$\Delta \phi_{uv}^{L-1,b} = \eta\Big[\sum_{\mathbf{y}_j \in \mathbf{Y}^0} \sum_{\substack{\mathbf{h}_i \in \mathbf{H}_{L-1} \\ (\mathbf{y}_j, \mathbf{h}_i) \in \mathbf{E}}} \mathcal{E}_{P(\mathbf{h}_i \,|\, \mathbf{Y}^0)}[h_i^u y_j^v] - \sum_{\mathbf{y}_j \in \mathbf{Y}^1} \sum_{\substack{\mathbf{h}_i \in \mathbf{H}_{L-1} \\ (\mathbf{y}_j, \mathbf{h}_i) \in \mathbf{E}}} \mathcal{E}_{P(\mathbf{h}_i \,|\, \mathbf{Y}^1)}[h_i^u y_j^v]\Big] \tag{4.22}$$

$$\Delta \phi_{uv}^{t,b} = \eta\Big[\sum_{\mathbf{h}_i \in \mathbf{H}_t} \sum_{\substack{\mathbf{h}_j \in \mathbf{H}_{t+1} \\ (\mathbf{h}_i, \mathbf{h}_j) \in \mathbf{E}}} \mathcal{E}_{P(\mathbf{h}_i, \mathbf{h}_j \,|\, \mathbf{Y}^0)}[h_i^u h_j^v] - \sum_{\mathbf{h}_i \in \mathbf{H}_t} \sum_{\substack{\mathbf{h}_j \in \mathbf{H}_{t+1} \\ (\mathbf{h}_i, \mathbf{h}_j) \in \mathbf{E}}} \mathcal{E}_{P(\mathbf{h}_i, \mathbf{h}_j \,|\, \mathbf{Y}^1)}[h_i^u h_j^v]\Big] \tag{4.23}$$

Here we can observe that the Z term cancels out due to the ratio of the two probabilities, hence evaluating partition function for drawing samples is no longer needed.

We can expect CD to get stuck in some local minima, which is the issue with almost every hill climbing method. For the TCRF model which has a probability distribution given by product of experts (Hinton, 2002), CD proves to be a suitable choice.

4.5 Inference

The problem of inference is to find the optimal label field $\mathbf{Y}^* = argmax_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X})$. This can be done using the maximum posterior marginals (MPM) criterion which minimizes the expected number of incorrectly labeled sites. According to the

MPM criterion, the optimal label for pixel i, $\mathbf{y}_i^*$ is given as,

$$\mathbf{y}_i^* = argmax_{\mathbf{y}_i} P(\mathbf{y}_i \,|\, \mathbf{X}), \quad \forall \mathbf{y}_i \in \mathbf{y} \tag{4.24}$$

When a graph is tree-structured, efficient exact algorithms (Pearl, 1988) exist for the computation of the marginal distributions $P(\mathbf{y}_i|\mathbf{X})$. One of them is Belief Propagation (BP). Belief propagation, also known as the *sum-product* algorithm, is an iterative algorithm for computing marginals of functions on a graphical model. We have used BP for the MPM estimate. It is equivalent to the sum-product algorithm developed by the coding community. This algorithm functions by passing real-valued messages across edges in a graphical model. More precisely, in trees: a vertex sends a message to an adjacent vertex if (a) it has received messages from all of its other adjacent vertices and (b) hasn't already sent one. So in the first iteration, the algorithm will send messages from all leaf nodes to the one vertex adjacent to its respective leaf and continues sending messages in this manner until all messages have been sent exactly once, hence explaining the term propagation. It is easily proven that all messages will be sent. Upon termination, the marginal of a variable is simply the product of the incoming messages of all its adjacent vertices.

Careful examination of the message-passing equations (Sudderth *et al.*, 2002), show that for tree-structured graphs, the resulting messages can produce the exact conditional marginals in only $\mathcal{O}(M^2N)$ operations, a huge savings over the direct cost of $\mathcal{O}(M^N)$. Here M is the total number of values each random variable can take and N is the number of nodes in the graph.

4.6 Feature Selection

TCRF model is mostly based on learning the context relationship, but for any image modeling system feature selection still remains a crucial task. Visual contents of an image such as color, texture, and spatial layout are vital discriminatory features when considering real world images.

The color information is extracted by calculating the first three color moments of an image patch in the $CIEL^*a^*b^*$ color space[1]. These three moments – mean, variance, and skew can be calculated as,

$$M_i^q = \begin{cases} \frac{1}{N}\sum_{j=1}^{N} p_{ij}, & q = 1 \\ (\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - M_i^1)^q)^{\frac{1}{q}} & q = 2, 3, \ldots \end{cases} \tag{4.25}$$

Here, p_{ij} denotes the i^{th} color component e.g. R, G, B or L^*, a^*, b^* of the j^{th} pixel, N is the total no of pixels and h is the order of the moment.

We incorporate the entropy of a region to account for its randomness. A uniform image patch has a significantly lower entropy value as compared to a non-uniform patch. Entropy can be calculated using $-sum(p. * log(p))$ where p contains the histogram counts. The calculation of image entropy for change detection was first proposed by (Schneider and Fernandes, 2002). Texture is extracted using the discrete wavelet transform (Mallat, 1996). Spatial layout is accounted for by including the coordinates of every image site. This in all accounts for a total of 20 statistics corresponding to one image region. We term them as spectral features.

Color features are not very good discriminators of buildings from background.

[1] 1Commission Internationale d'Eclairage, -color model proposed by International Commission on Illumination, used conventionally to describe all the colors visible to the human eye.

Man made structures are observed to be more linear and structured in nature. Hence for man made structure detection problem we have computed multiscale features proposed by Kumar and Hebert (2003*b*). These features capture the linear properties of the image blocks. In all, a 14 dimensional feature vector is computed for each image block.

4.7 Discussions

To summarize, TCRF as discussed in this Chapter has various advantages over the existing discriminative models like DRF (Kumar and Hebert, 2003*a*), mCRF (He *et al.*, 2004) because of its tree like hierarchical structure. Not only this, all the training parameters are trained simultaneously which gives the model the power to model better interaction among the local and the edge parameters unlike the mCRF (He *et al.*, 2004, 2006). The hidden variables are also not assumed to be independent of each other like in mCRF. The tree like structure enables the inference in time linear in number of nodes using belief propagation. The model performance is evaluated by applying it on two different image processing problems. Applications are discussed in the Chapter 5.

CHAPTER 5

Application of TCRF

5.1 Experiments

In this chapter we evaluate the modeling power of TCRF by applying it to two real world computer vision problems namely, image labeling and object detection. Our main aim was to demonstrate the ability of TCRF in capturing label relationships. The graph structure of TCRF has strong potential to capture patterns in images, present at different levels of scale. For example, in a multiclass problem a rich set of patterns can be found because of the presence of many labels, which are efficiently captured by the TCRF. However, when there are very few labels, like in object detection problem, TCRF is not likely to improve over other non-multiscale models because there are not enough patterns to be captured.

The input image is first preprocessed such that one training point is equivalent to a fixed block size. The total number of training blocks for each image is chosen such that it can be accommodated at a leaf level of the quad tree. This preprocessing of the input image is necessary to limit the number of levels in the tree structure, and in turn limit the number of parameters. The predicted label for each block is assigned to all the pixels contained within it. This is done in order to compare the performance against models like mCRF which operate directly at the pixel level. A generalized form of the TCRF is shown in algorithm 1.

Algorithm 1 Training and Testing of TCRF

A. Training Phase

Input: Set of training images each of size $M \times N$

Output: Weights

1. Divide image into regions of size $r \times c$ such that $(M/r \times N/c) = P \times P$ and P is divisible by 4 (quad tree structure)
2. Compute feature vector for each block
3. TRAINING -
   (a) Initialize $\mathbf{W}$ and $\mathbf{\Phi}$ randomly, $\eta = 0.01$
   (b) **Loop** for Number of maximum Iterations
      i. **Loop** for Number of training images
      ii. Run BP to initialize hidden nodes $\mathbf{H}$ and compute the local and edge potential
      iii. Compute one step reconstruction of the label nodes $\mathbf{Y}^1$
      iv. Compute the first and the second terms of equations (4.21),(4.23),(4.23), by running BP
      v. Sum them for all the images
      vi. **End of Loop** over images
   (c) Update weights as in equation (4.21),(4.23),(4.23)
   (d) **End of Loop** over Iterations
   (e) Save the Weights $\mathbf{W}$, $\mathbf{\Phi}$

B. Testing Phase

Input: Test image of size $M \times N$, Weights

Output: Label over the image

1. Divide image into regions of size $r \times c$ such that $(M/r \times N/c) = P \times P$ and P is divisible by 4 (quad tree structure)
2. Compute feature vector for each block
3. TESTING-
   (a) Compute local and edge potential using Weights
   (b) Run BP to estimate probability distribution P (equations 4.4)
   (c) **for** $j = 1$ to $NoOfLeafNodes$ **do**
      $\mathbf{Y}_j = max(P_j)$
      end for
      Save Label $\mathbf{Y}$

5.1.1 Image Labeling

The problem here was to label each pixel of the input image into one of the predefined set of classes. The Corel data set (only a small version is available in public domain[1]) consisting of wildlife images was considered for this experiment. In a set of total 100 images, 60 images were taken for training (Figure 5.1 shows some of the training images) and 40 for testing. The training and test image set is kept similar to that of mCRF experimentation (He *et al.*, 2004). There were total of seven classes to be identified, *rhino/hippo, polar bear, vegetation, sky, water, snow* and *ground*. Each image was of size 128×192 pixels. In order to adjust the image into the quad tree structure, each region of size 4×6 was considered as one block, which would make the image of size 32×32 blocks. Hence, there were 1024 nodes at the leaf level. A 20 dimension feature vector as discussed in section 4.6 was computed for each of these entities. The learning rate η was set to 0.01. The target label for each block was assigned the class which occurs majority of times. TCRF was trained using contrastive divergence algorithm. The results were obtained for test images by running BP inference algorithm as detailed in section 4.5. Labels obtained by different models like LR, mCRF and TCRF on some of the Corel images is shown in Figure 5.2. As can be observed, TCRF labels are better in comparison to the other models. LR almost misses the entire hippo, mCRF captures it but with lot of misclassification, while there are quiet few misclassification with TCRF. Some more results are shown in Figure 5.3. A comparison on classification accuracy of different models is illustrated in Table 5.1. Table 5.2 shows the confusion matrix for the Corel set, the column indicates

[1]The dataset has been taken from the URL http://www.cs.toronto.edu/ hexm/label.htm.

the true classes and the row shows the predicted class label by TCRF. If we observe the matrix we can see that the maximum accuracy is along the diagonal elements. In the first row, the maximum confusion is among the hippo and the ground class, which is very obvious because of there similar color properties. In the second row, bear is mostly confused with the ground pixels. The reason being, the ground pixels which are at the boundary of the bear, might get classified into bear because of the block input instead of the pixel input. This is the same reason why most of the vegetation pixels get misclassified into ground. In the last row, we can see that most of the sky pixels are misclassified into vegetation class, the reason here is that the percentage ratio of sky class in the dataset is too low, hence this class is not learnt properly.

Fig. 5.1: Some of the images from the Corel data set used for training the TCRF model. It can be observed that the images have varying illumination. The color and texture properties of rhino/hippo are also not uniform. Visually also polar bear and snow are very ambiguous in nature. Such properties of the image makes labeling a difficult and challenging problem.

5.1.2 Object Detection

Object Detection is the problem of identifying the presence of an object in a given image, without actually identifying the type of the object i.e. given an image with

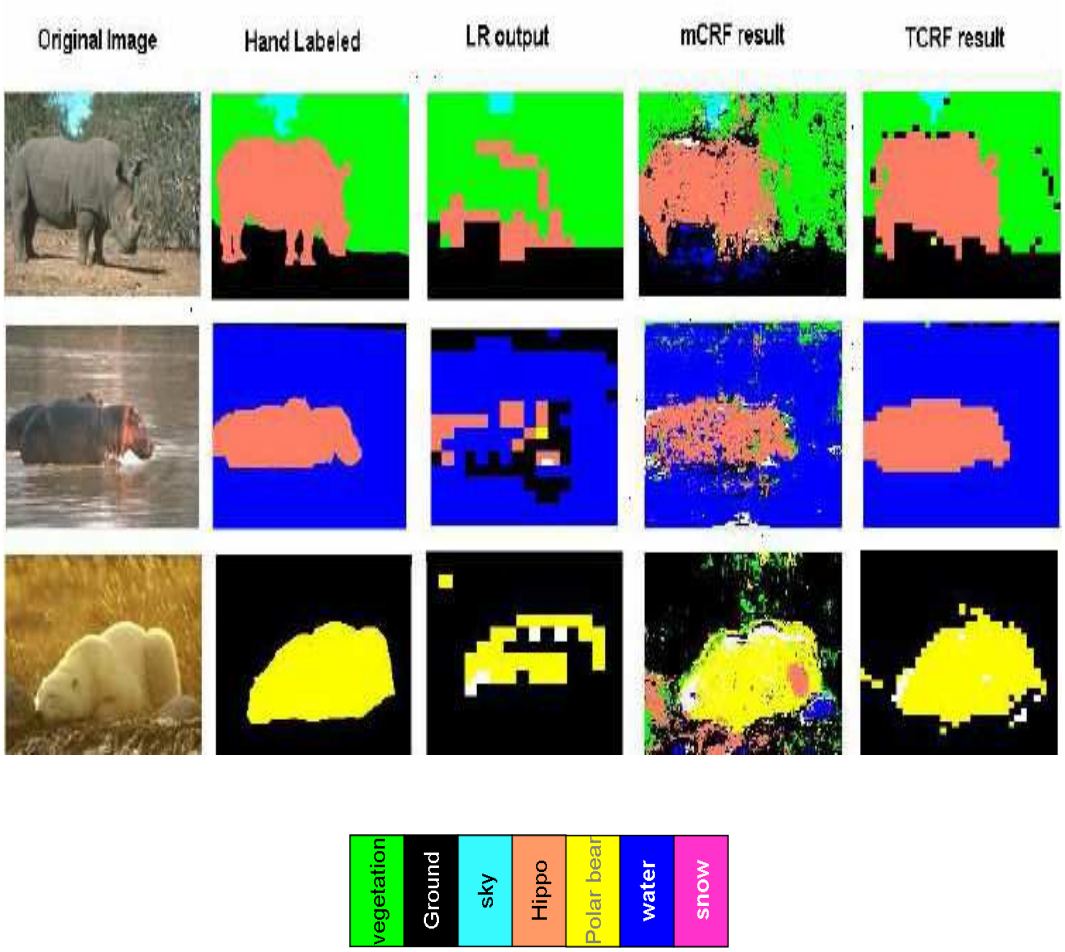


Fig. 5.2: Image labeling results on the Corel test set. The TCRF achieves significant improvement over the logistic classifier by taking label relationships into account. It also gives good performance in cases where the mCRF performs badly as shown above. The color coding of labels is shown in the color bar.

Table 5.1: Classification accuracy for the task of image labeling on the Corel data set. A total of 40 test images were considered each of size 128×192.

Model	Classification Accuracy (%)
LR	65.03
mCRF	74.3
TCRF	**77.76**

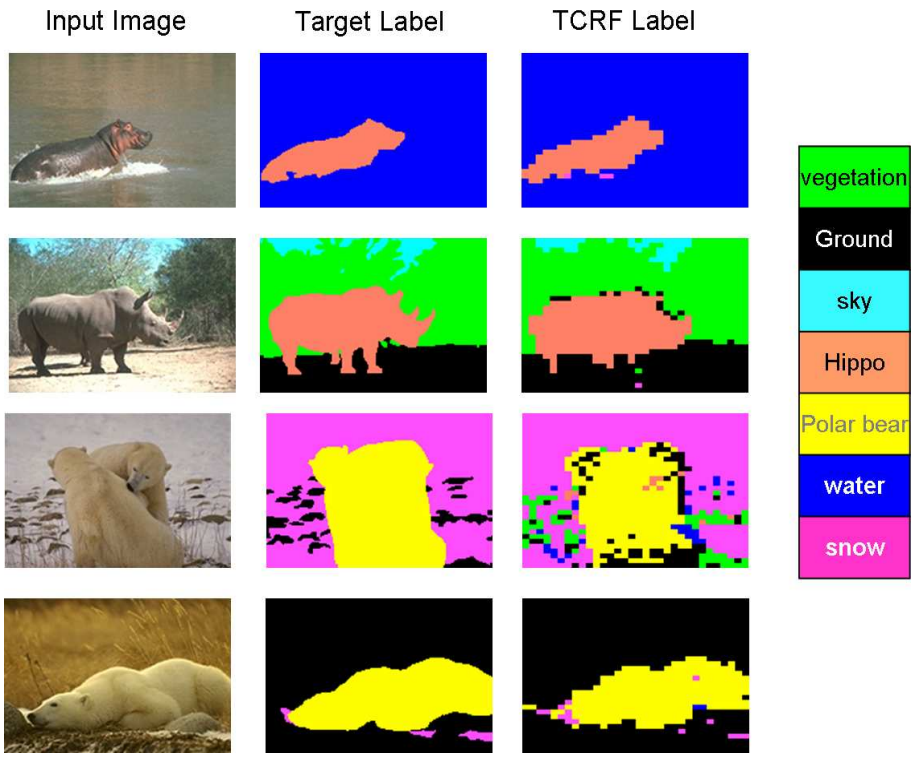


Fig. 5.3: Some more results on obtained on corel images with TCRF.

Table 5.2: Confusion Matrix for the TCRF model on the corel image set. Each entry in the table represents the percentage of pixels classified into a class shown by the first row out of the total pixels of that class, where the first column represents the true class. For example, out of all the hippo pixels in the test set 80.67% are classified as hippo, 4.5% is classified as Water and so on. The right most column represents the percentage of pixels of that class into the test data set. Here 'Veg' corresponds to 'Vegetation', 'Grnd' corresponds to the 'Ground' class.

	Hippo	Bear	Water	Snow	Veg	Grnd	Sky	%age of each class in the test set
Hippo	80.67	1.0	4.5	0.31	3.14	9.6	0.53	13.12
Bear	2.7	66.55	4.4	4.09	1.54	20.59	0	8.64
Water	2.55	0.25	82.02	12.7	1.2	1.23	0	24.01
Snow	0.7	6.02	12.6	66.12	7.7	5.1	1.5	11.29
Veg	3.1	1.29	2.21	1.68	71.5	19.8	0.25	21.32
Grnd	3.1	1.8	7.4	2.5	7.8	77.158	0.03	20.93
Sky	0	0	0.20	0.3	32	1.6	65.8	0.69
Total								100

chairs and tables, we identify them as furniture and not particularly as a 'chair' or 'table'. Object detection problem assumes that there are only two classes, object vs non-object. We have shown test results of object detection on two different data sets. In first case the task is to detect man made structures in natural images. In the second data set, images contain different animals on the ground and the task is to detect those animals.

In the first task, we apply our model to the problem of detecting man-made structures in a set of natural images from the Corel image database.[2] Each image is 256×384 pixels. It is a two class problem where in each image we identify man made and non-manmade blocks. From the dataset of total 237 images, for our experiments we considered 108 training and 129 test images. Figure 5.5 shows some of the images from the training set. In order to fit the image into the quad tree structure, the image was divided into the regions of size 8×12 pixels each, which would make the image of size 32×32 blocks. A 14 dimensional multiscale feature vector was computed as detailed in section 4.6. We also implement a simple Logistic Regression (LR) based classifier in order to demonstrate the performance improvement obtained by capturing label relationships (Table 5.3). TCRF achieves improvement over the LR based classifier which takes only local image statistics into account. The detection rate achieved by TCRF is 53% with false positive rate of 3.4%, where as, the false positive rate obtained with LR, keeping detection rate at 52.36%, is 6.35%. As followed in literature (Kumar and Hebert, 2003*a*), the false positive rate has been computed by ignoring the blocks which occur at the man made structure boundary and are identified as

[2]The dataset has been downloaded from http://www.cs.cmu.edu/ skumar/manMadeData.tar

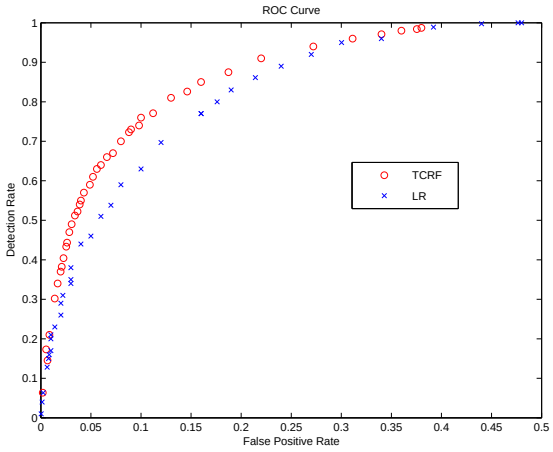


Fig. 5.4: This figures shows the ROC curve of TCRF (learns context)and LR(no context information is included).

false positives. Figure 5.4 represents a ROC curve of TCRF and LR. It can be observed that the increase in false positive rate of LR is higher than that of the TCRF, since no context information is included in LR classifier.

The best reported result on man made structure detection is a little higher as compared to the TCRF, a reason for this is that there are not sufficient patterns present for TCRF to capture. The comparison of the accuracy with other models is shown in table 5.3. Some of the results obtained for man made structure detection problem are shown in figures, 5.7-5.14. It can be observed in the Figure 5.7 that TCRF has captured even building blocks with very small extent, very accurately. Figure 5.6 show labels on a singularly tough image, where DRF model (Kumar and Hebert, 2003a) completely fails to identify the man made structure, LR captures man made blocks, but with several false positives and TCRF has accurately identified the man made blocks.

It can be observed that TCRF has a very interesting property because of its tree like structure. If we rotate the image by 180 degrees, the labels do not get

Fig. 5.5: Some example images from the training set for the task of man-made structure detection in natural scenes. This task is difficult as there are significant variations in the scale of the objects (row 1), illumination conditions (row 2), perspective distortions (row 3).Row 4 shows some of the negative samples that were also used in the training set

affected much. This is because the TCRF graph structure encodes the neighborhood relationship, hence even if the image is rotated such that the neighborhood does not change, the labeling does not get effected. Observing a man made structure in a rotated image is not a very trivial task even for a human eye, but since the TCRF models context using the graph structure which does not get affected by the respective orientation of the image as far as the aspect ratio of the image is kept same. Hence we can claim here that TCRF is also rotation invariant conditioned that image is rotated keeping aspect ratio of the image unchanged. This claim is made only for the man made structure detection problem. In case of multiclass problem, the respective position of the classes is part of the context information learnt, hence TCRF is not rotation invariant for those cases. Figure 5.15 shows some of the results with the rotated images.

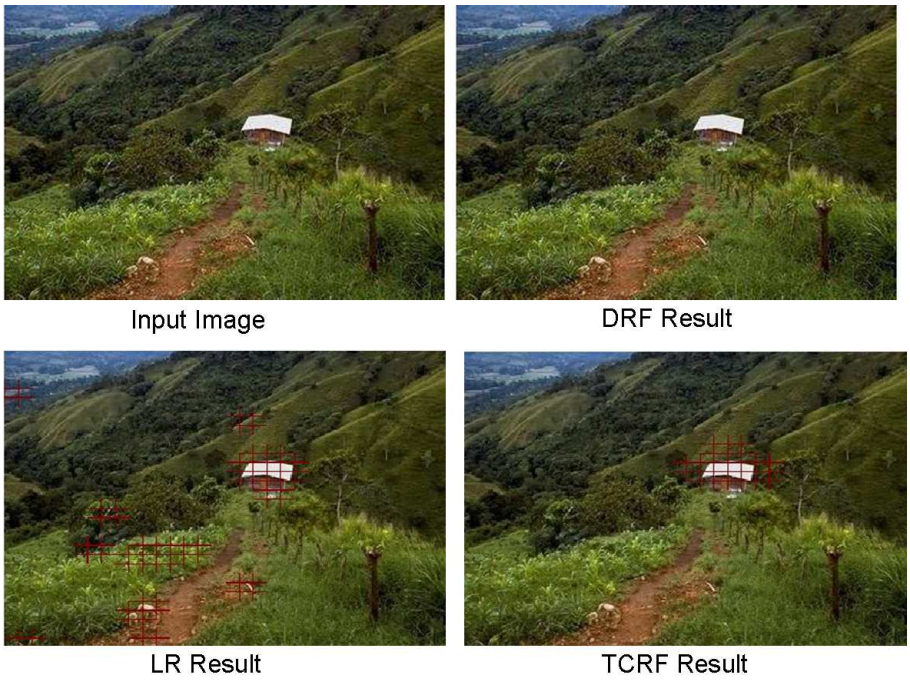


Fig. 5.6: This shows result on a difficult image, here the man made structure is very small in size. Discriminative Random Fields(DRF) completely fail to identify it and LR identifies, but with several false positives. TCRF output is quite accurate without any false positives. The red blocks show the correctly identified man made blocks.

Table 5.3: Comparison of classification Accuracy for detecting man-made structures calculated for the test set containing 129 images.

Model	Classification Accuracy (%)
LR	86.8
DRF [4]	94
TCRF	90

The performance of TCRF is observed to be poor when the non-manmade entities display linear properties. Figures 5.16 show some of the results where TCRF shows relatively poor performance. In the topmost figure some of the flowers are also identified as man made block because of their arrangement, which is showing some linear features. In the second image, if observed carefully the false positives are the block which have very linear arrangement, secondly the color properties also match with the building roof. In the last figure, the tree stems are also long and structured like a building, which makes it difficult to disambiguate it just on the basis of its color features. Performance of TCRF as an object detection tool has also been shown on a simpler animal database where the animal in the image is considered to be the object to be detected. Figures 5.17 and 5.18 show some of the results from this database. These images have been obtained from the Microsoft Research Cambridge Database. [3] Total 40 images were taken for training and 20 images were considered for testing. Each image is of size 224×320. The block size for input was 7×10. The classification accuracy obtained for this dataset was 89.63% at the block level. The feature descriptors were same as used for image labeling case, detailed in section 4.6. The performance measures are detailed in Table 5.4.

[3]The database can be freely downloaded from the url http://research.microsoft.com/vision/cambridge/recognition/default.htm

Table 5.4: The performance measures on the animal dataset. 'DR' represents the detection rate, 'FP' represents the false positive rate and 'CA' represents the classification accuracy at the block level.

Model	DR (%)	FP (%)	CA (%)
LR	70.66	4.9	88.6
TCRF	78	5	89.6

5.2 Discussions

In the above sections we have illustrated the TCRF performance by applying it on two different problems. TCRF, as an image labeling tool has performed better than the other existing techniques hence supporting the claim that the hierarchical structure of the TCRF is very effective in capturing long range contextual relationship.

The performance of TCRF is highly dependent on the extent of context information present in the dataset. In case of two class problem, as an example, TCRF has not performed better over some of the existing methods because there are not enough patterns for TCRF to learn. Secondly, the dataset is highly skewed in nature which makes learning even more difficult, in general.

Due to the non-availability of sufficient number of labeled dataset, an extensive experimentation could not be possible to show the performance of TCRF as an image labeling tool on some more multiclass problems.

[4]This score as reported by Kumar and Hebert (2003*a*) was obtained by using a different region size.

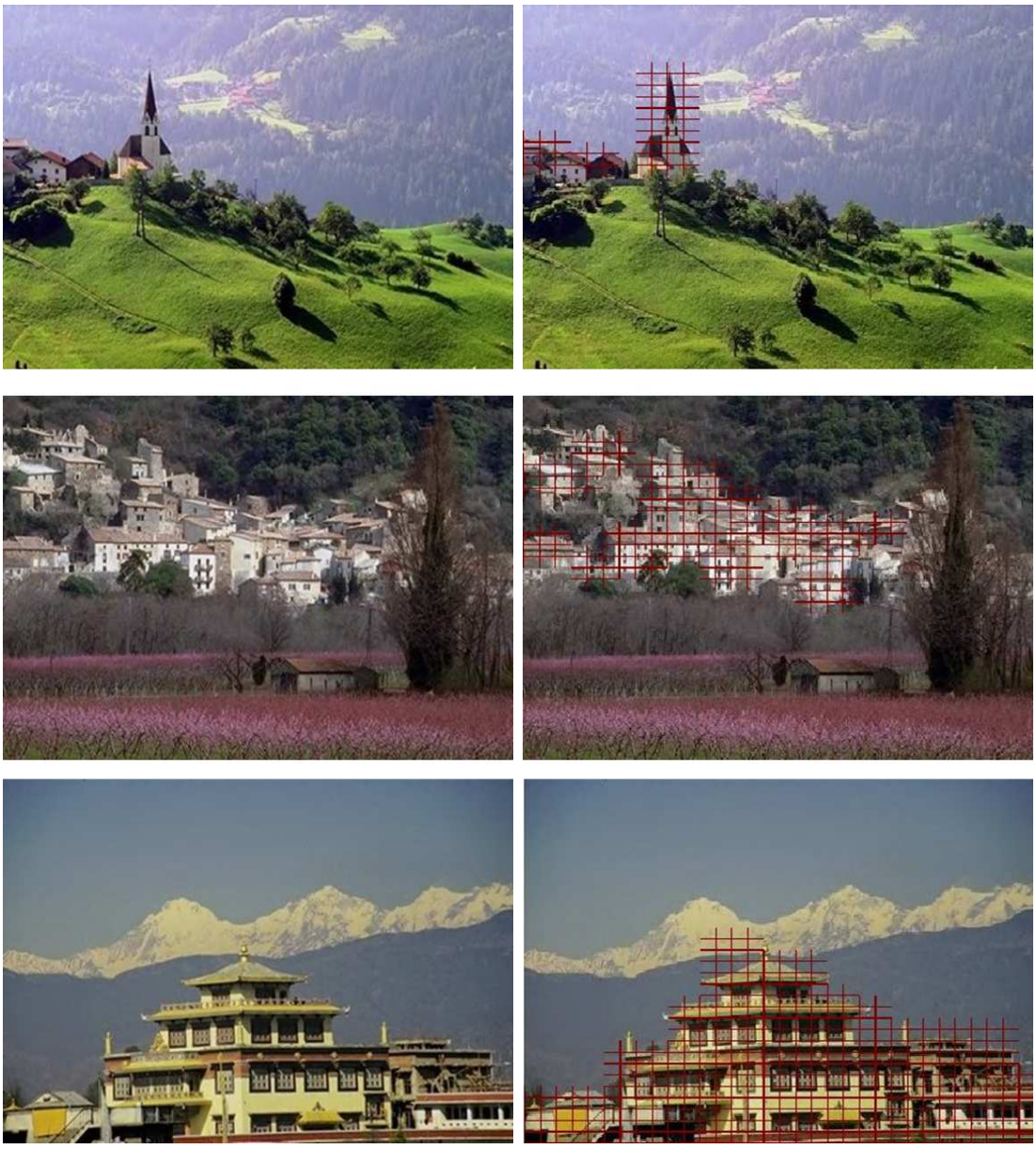


Fig. 5.7: The result obtained on a man made structure detection database. The red blocks show the correctly identified man made structure. The images are quiet varying in their spectral properties and so are the man made structures in their sizes. Because of the multiscale nature of TCRF and the features used, man made structures of every size has been correctly identified.

Fig. 5.8: The result obtained on a man made structure detection database. The red blocks show the correctly identified man made structure. The first figure shows a hard example where the color properties of the man made structure are very similar to the background, however label produced by TCRF is accurate.

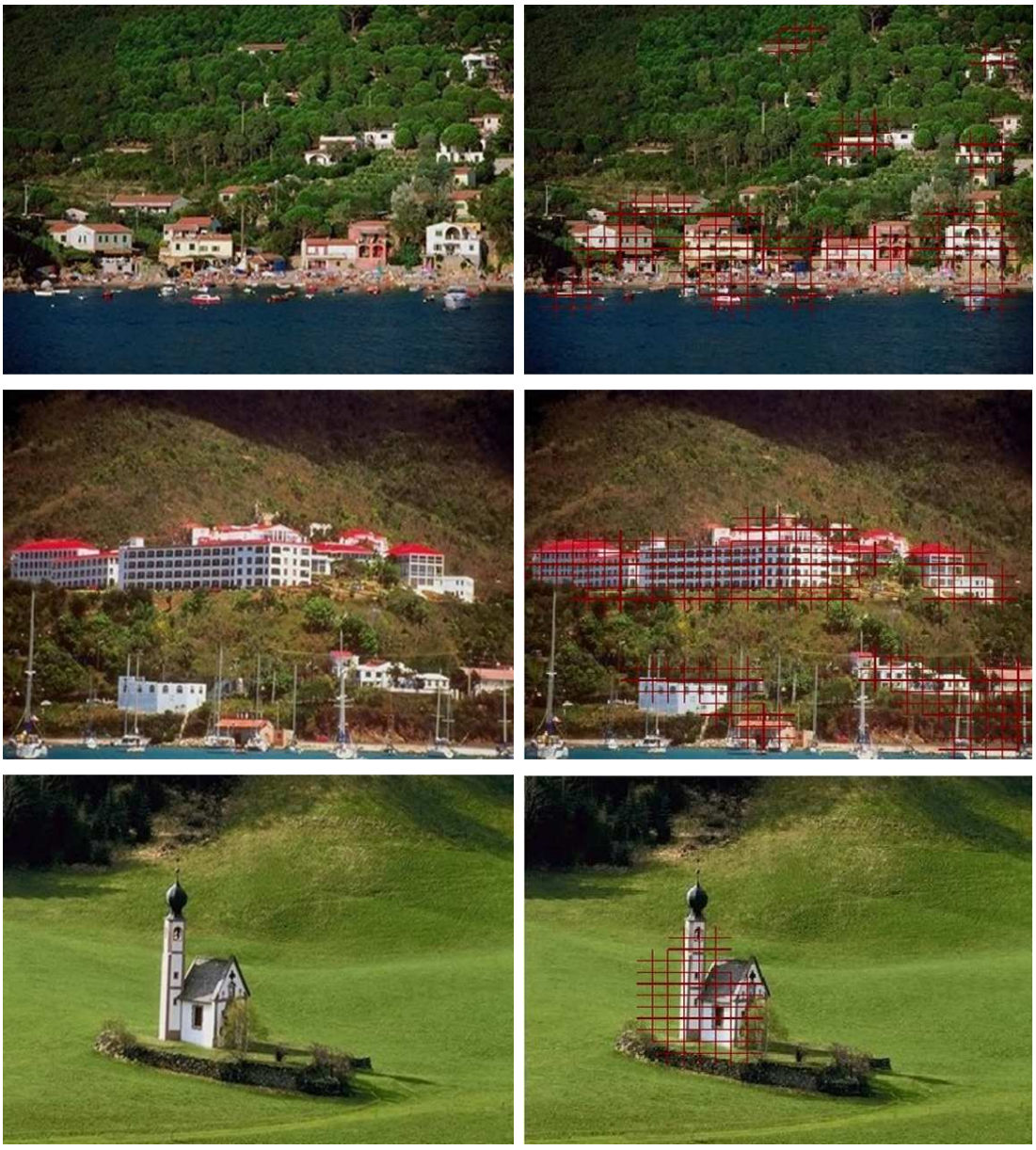


Fig. 5.9: The result obtained on a man made structure detection database. As can be seen even very small man made structures hidden in bushes have been identified.

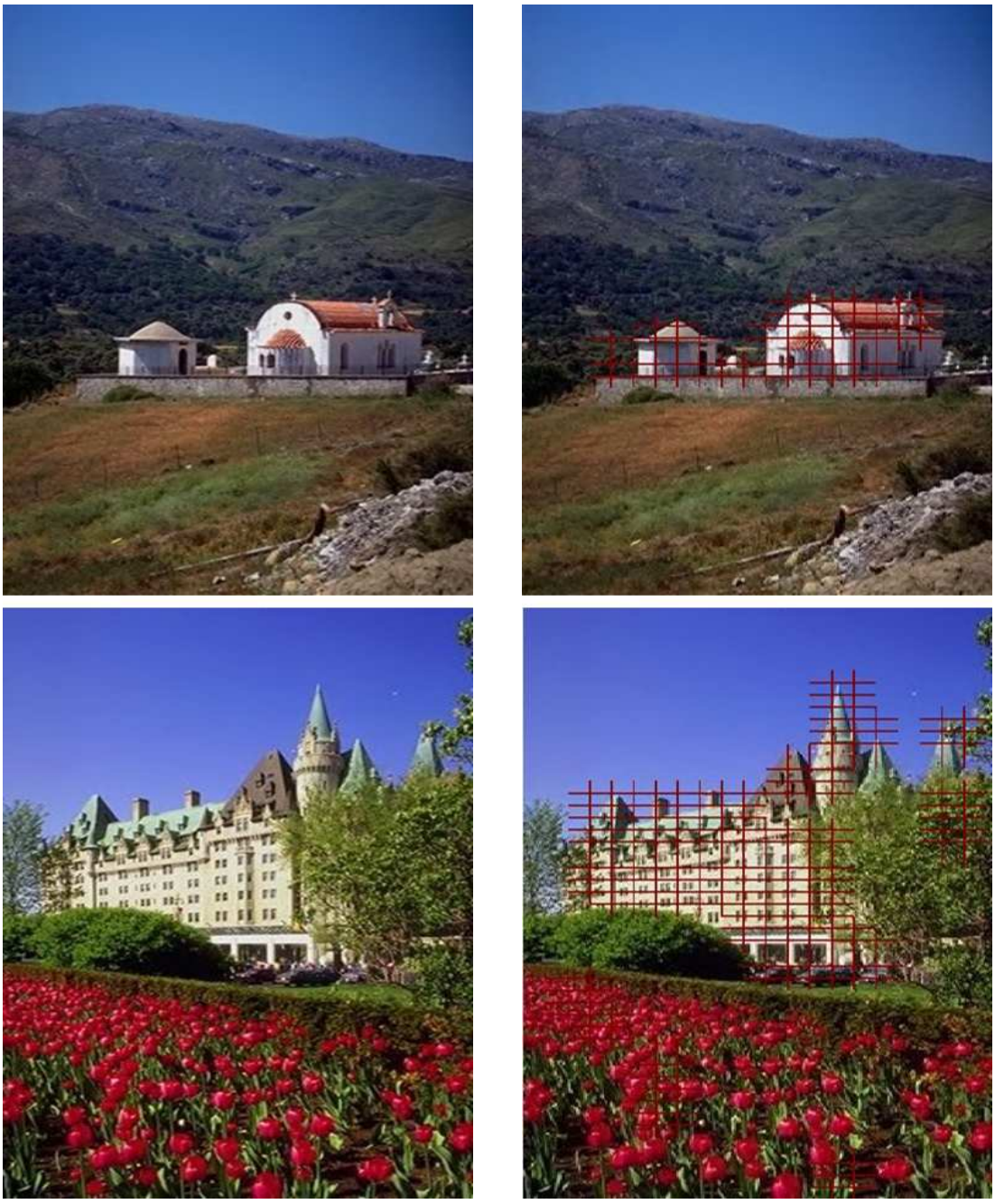


Fig. 5.10: The result obtained on a man made structure detection database. The man made blocks are identified with very few false positives.

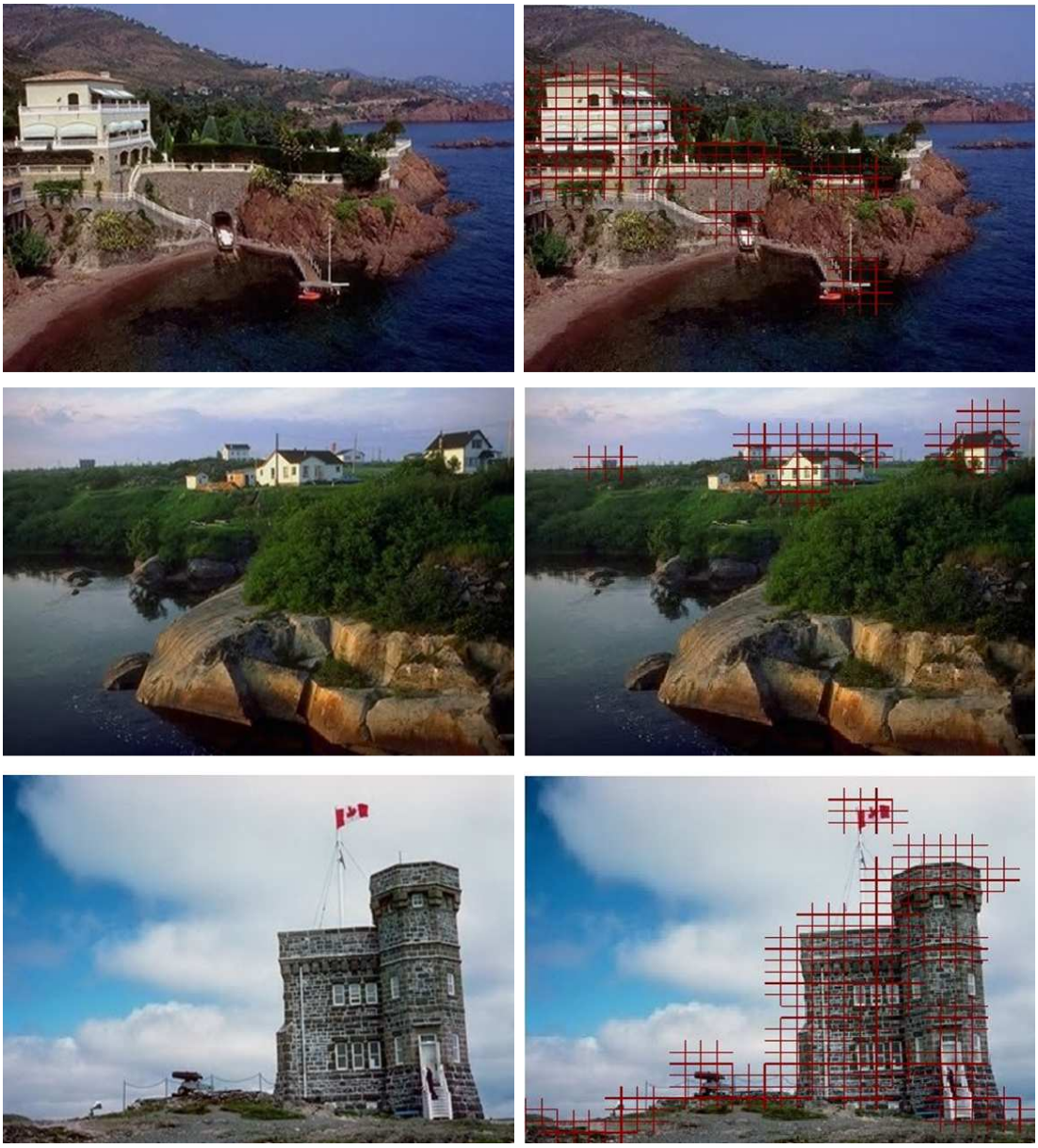


Fig. 5.11: In this image again TCRF shows outstanding performance for smaller sized man made structure.

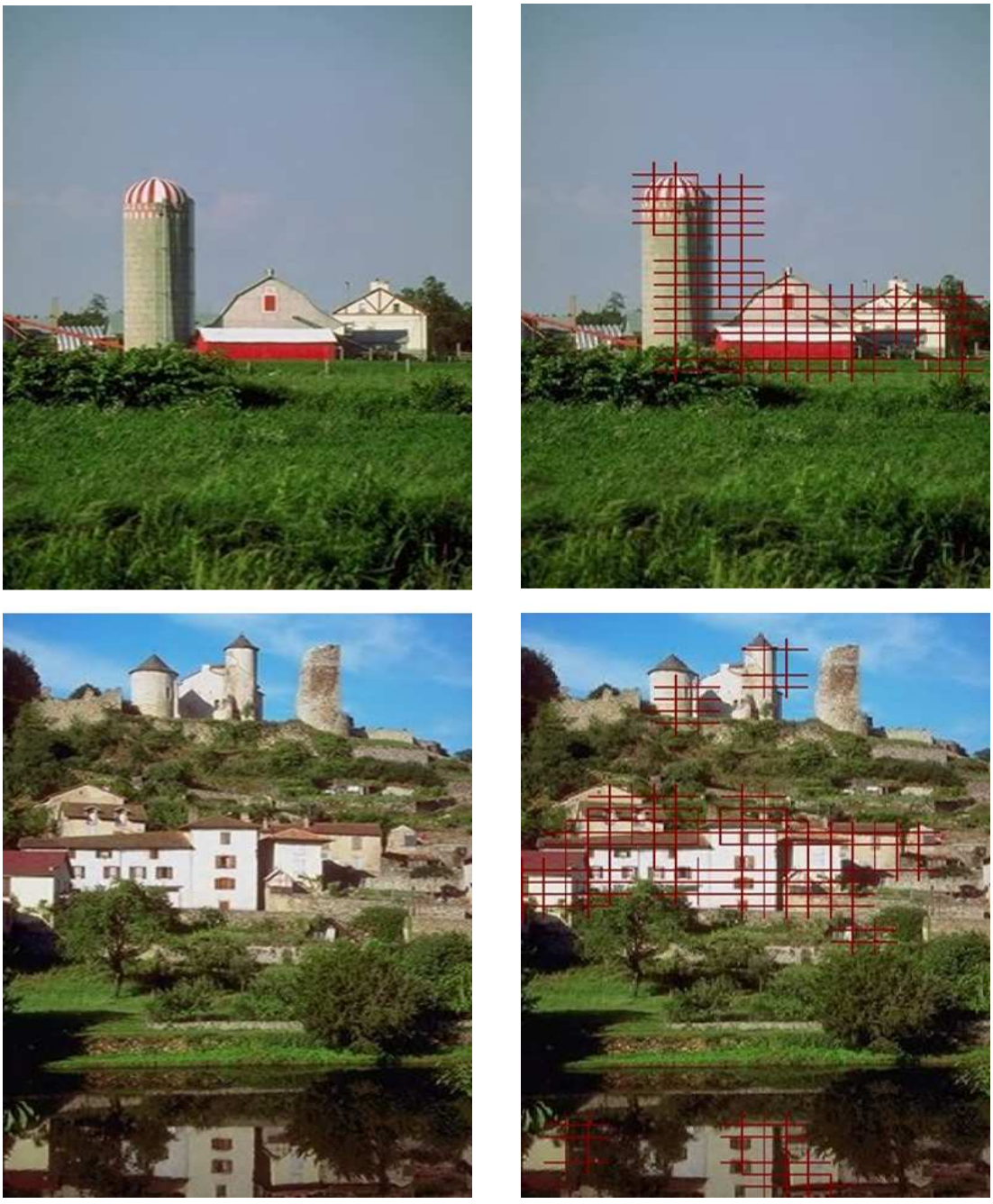


Fig. 5.12: In the lower image, the reflection of the building has also been identified as man made blocks, but it is debatable whether reflection is man made or not.

Fig. 5.13: The roof of the building has not been detected as man amde because of its smooth properties.

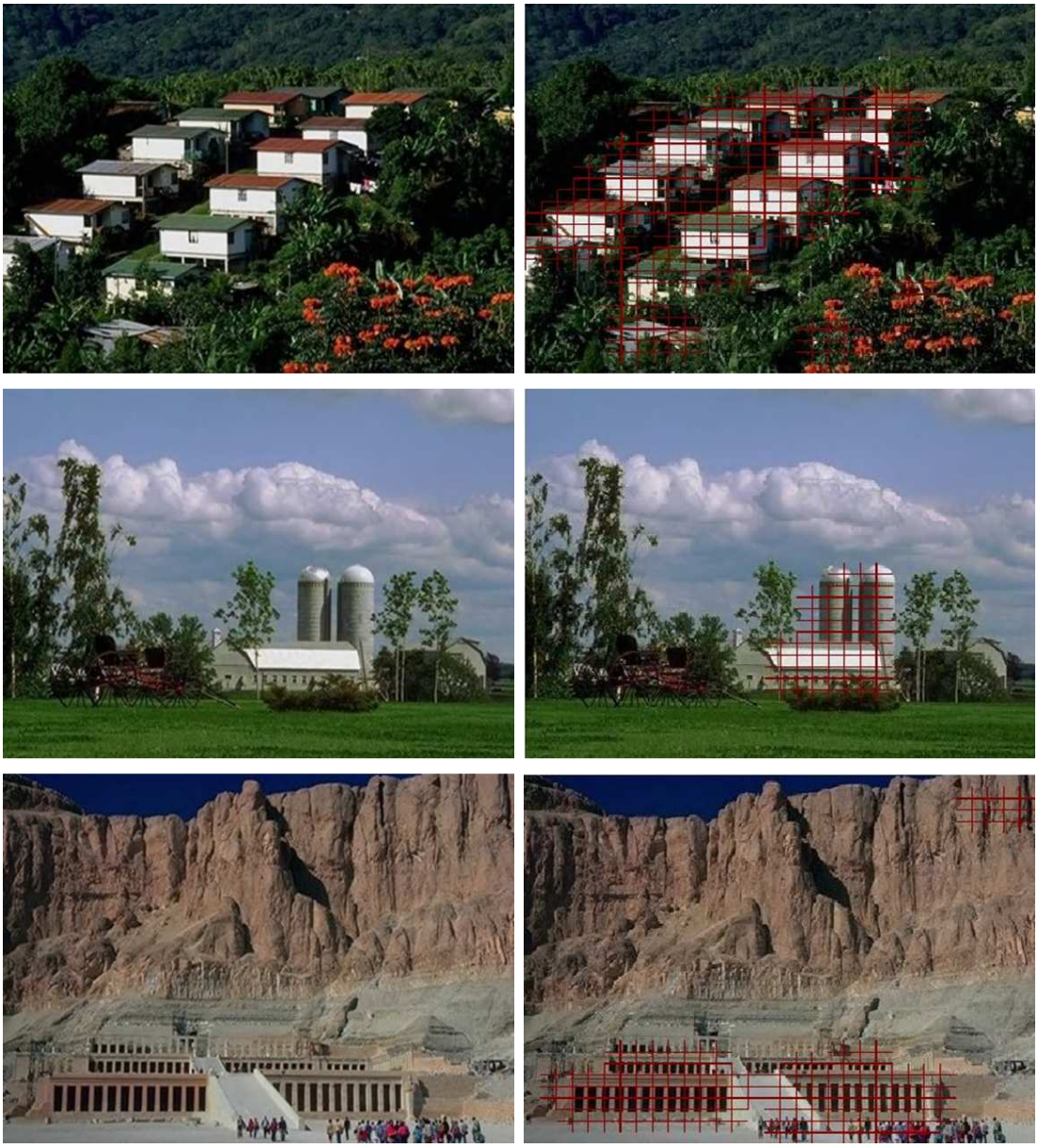


Fig. 5.14: In this image again TCRF shows outstanding performance for smaller sized man made structure.

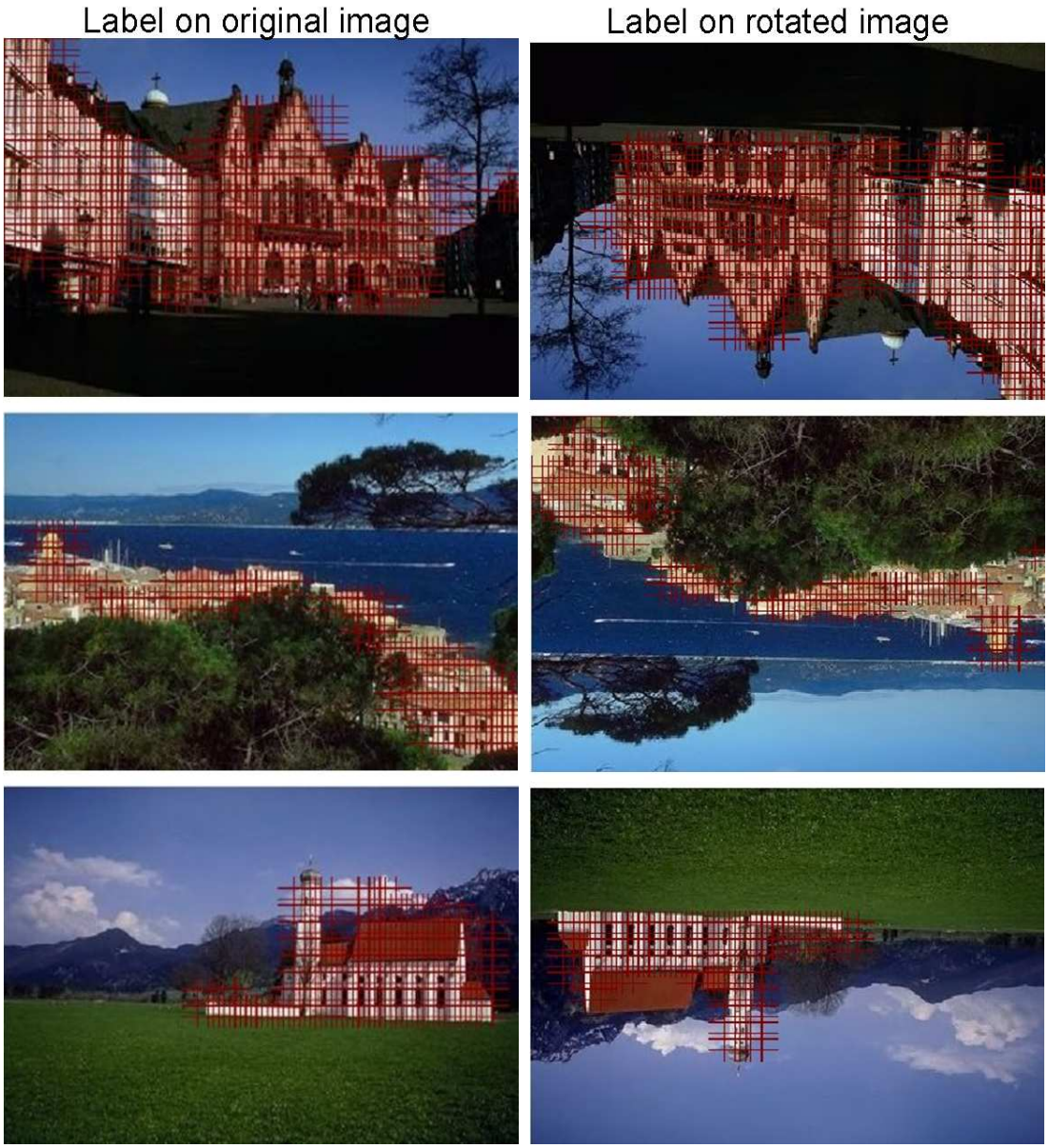


Fig. 5.15: This experiment shows that the rotation of an image(assuming that the aspect ratio has been not changed) does not change the result.

Fig. 5.16: This shows some of the poor results. The flower arrangement is quiet linear in structure hence gets misclassified as man made. In the second image again there are some false positives because they are displaying linear nature. The tree stems are long and well structured hence they also get misclassified.

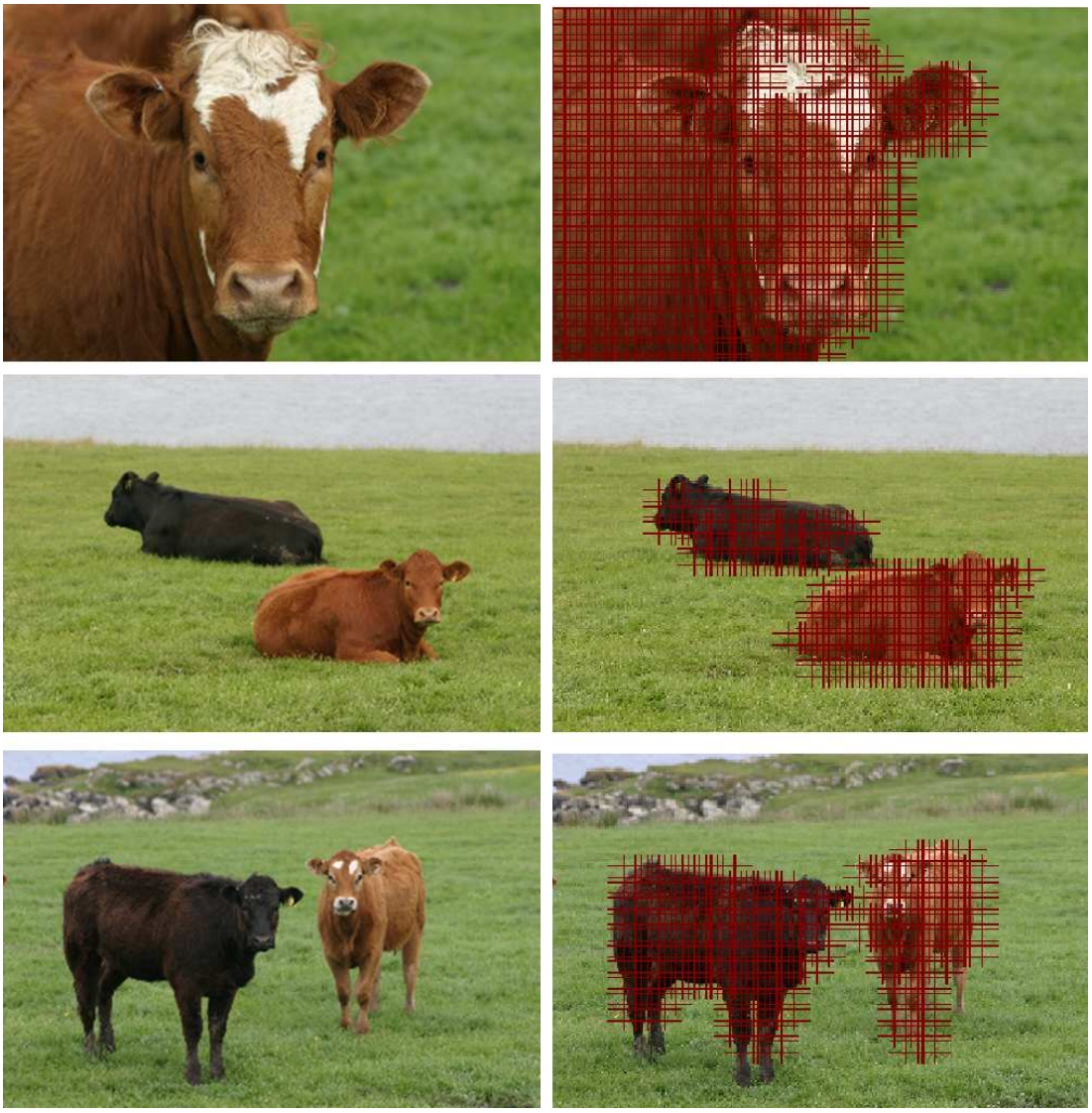


Fig. 5.17: Results on animal data set. The column on the left is the input image and that on right shows the TCRF output . The red colored blocks represent the correctly labeled animal block.

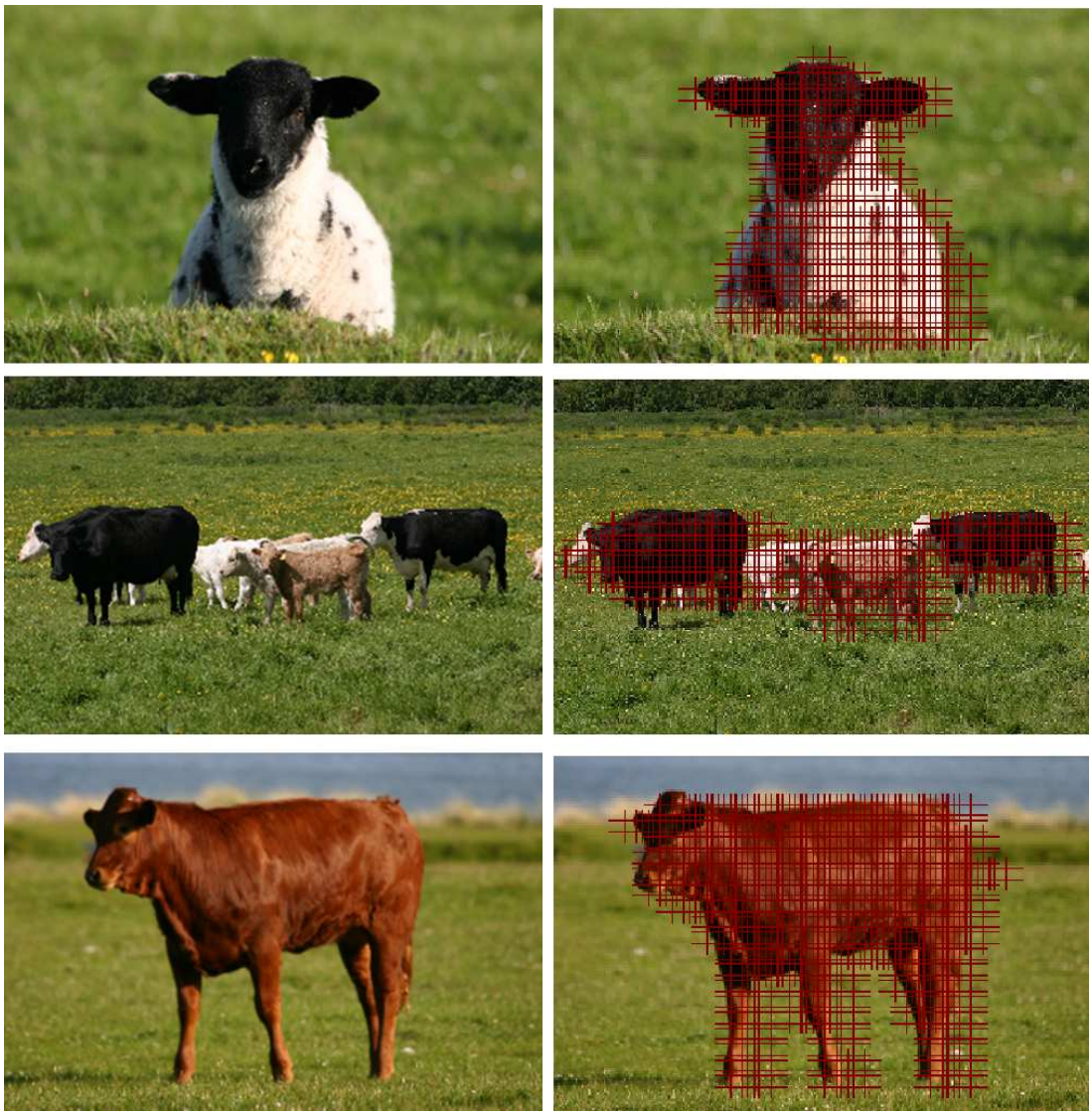


Fig. 5.18: Some more results on animal data set. The column on the left is the input image and that on right shows the TCRF output . The red colored blocks represent the correctly labeled animal block.

CHAPTER 6

Conclusion and Discussions

In this thesis we have presented TCRF, a novel hierarchical model based on CRFs for incorporating contextual features at several levels of granularity. The discriminative nature of CRFs combined with the tree like structure gives TCRFs several advantages over other proposed multiscale models. TCRFs yield a general framework which can be applied to a variety of image classification tasks. We have empirically demonstrated that a TCRF outperforms existing approaches on some tasks.

6.1 Contribution

Towards this, the thesis makes the following key contributions,

- A unified and novel framework is proposed which combines the ideas from hierarchical and discriminative models to capture contextual information at varying scales.
- Single framework is shown to work for both binary and multiclass problems without much change in the model. Hence it is not tuned for a specific task.
- The model architecture has been specifically chosen such that inference can be efficiently done in time linear in number of nodes.

6.2 Key Observations

In this thesis we have explored the effect of introducing hierarchy in capturing the context in the classification of image components. We also observed the performance of the model on different image modeling tasks. On the basis of the theoretical and the experimental observations made in this thesis, we summarize the key insights as follows.

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

Among the noncausal models, traditional generative MRF formulations are too restrictive to model the rich interactions in data and labels required for a variety of classification tasks in computer vision. In particular, MRFs do not allow data dependent label interactions that are critical for many vision applications such as parts-based object detection. The discriminative random fields overcome most of the limitations posed by the traditional MRF frameworks by allowing arbitrary interactions in the observed image data, and data-dependent interactions in the labels. The experimental results conducted on hundreds of real-world images verify the power of these models on several applications. For robust classification, both the short-range context (e.g., pixelwise label smoothing) as well as the long-range context (e.g., relative configurations of objects or regions) in images must be exploited. This has been achieved by introducing a hierarchy of hidden nodes.

The hierarchical structure defined, is non loopy and hence empowers us to do inference in time linear in number of nodes.

6.3 Issues and Future Scope of Work

The model shows good performance for some of the image modeling tasks, but there are some limitations which come inherently with its graph structure. In order to limit the number of layers in the tree, it becomes crucial to consider blocky input, which in turn leads to blockiness effect in the labels. In case of very small sized images the model can be directly applied at the pixel level, but for higher resolution images, the model has to be trained at the block level. It is important to limit the tree size because the parameter size grows exponentially with the levels in the tree.

Due to the presence of hidden nodes and the normalizing term, it is difficult to apply more efficient parameter learning algorithms. Secondly, the size of the training data has to be quiet huge for the model to learn all possible context information embedded into the training set. The model is observed to perform poorly if the training data is skewed. Hence it is important for all the classes to be sufficiently balanced for the model to learn them robustly.

Fully labeled training data is usually more expensive than unlabeled or partially labeled data. As the scope of computer vision expands to handle more complex objects and scenes, it will be increasingly hard to get enough fully labeled training samples. Thus, the development of unsupervised or semisupervised learning methods for these models is important for their wide applicability. Re-

cently, attempts have been made in this direction for the application of object detection (Fergus *et al.*, 2003; Quattoni *et al.*, 2005).

In order to completely exploit the TCRF model potential, we need to do more experimentation, on rather more complex problems. The experimentation shown in this thesis are limited due to the unavailability of the labeled data. It would be also interesting to explore the TCRF potential in the domain of video segmentation.

Variable length segments have been tested and implemented for $1D$ CRF's (Sarawagi and Cohen, 2005), extending them to $2D$ CRF's would find a very promising application in the domain of computer vision.

In summary, TCRF can be considered as a viable alternative to other multiscale models. In future we wish to explore other training methods which can handle larger number of parameters. Further, we need to study how our framework can be adapted for other image classification tasks.

Appendix

APPENDIX A

Landform Classification of Satellite Images

There is increasing need for effective delineation of meaningfully different landforms due to the decreasing availability of experienced landform interpreters. Any procedure for automating the process of landform segmentation from satellite images offer the promise of improved consistency and reliality. We propose a hierarchical method for landform classification for classifying a wide variety of landforms. At stage 1 an image is classified as one of the three broad categories of terrain types in terms of its geomorphology, and these are: desertic/rann of kutch, coastal or fluvial. At stage 2, all different landforms within either desertic/rann of kutch , coastal or fluvial areas are identified using suitable processing. At the final stage, all outputs are fused together to obtain a final segmented output. The proposed technique is evaluated on large number of optical band satellite images that belong to aforementioned terrain types. Landform Classification is a problem of identifying the predefined class of landforms, given a satellite image of the area. In order to explore the navigable areas, identification of the exact landform becomes a crucial task. Due to the varying geographic nature of landforms and existence of large number of classes, landform segmentation is very much different from a conventional image segmentation problem. Geographical definitions give only a very theoretical aspect of the size, shape and several other features of the landforms. For e.g. “Barchan dunes” are caused by highly uniform environmental conditions

and wind blowing only in one direction. Barchans can become aligned together along a plane perpendicular to the wind. If the line becomes somewhat straight, dune scientists refer to these forward marching ridges as "transverse dunes". For such kind of landforms shape is an important feature. However the definitions do not clarify the type of shape features to be used for processing. Another category is the coastal bar. Coastal bars have no specific color, shape or size. Formulation of these abstract geographical definitions into a single set of features and rules is a difficult task for the purpose of segmentation or classification. Hence a single classifier or a single set of features cannot efficiently handle various types of landforms from a satellite image, we propose a hierarchy of classifiers in a unified framework.

A few approaches have dealt with the problem of landform identification in the past. However, only a limited set of landforms were used for classification. Pennock et al. Pennock *et al.* (1987) has dealt with the problem by using self organizing feature map. They calculate the DEM (Digital Elevation Model) and the land cover map as features. The DEM map normally divides the area into rectangular pixels and store the elevation of each pixel. These features are then fed to the SOM for further classification. The method is used to classify the landform of Kobe city in Japan into hill, plateau, fan and reclaimed land. These classified landforms were adopted for an earthquake damage evaluation of the 1995 Hyogoken Nanbu earthquake in Kobe. Gorsevski et al. Gorsevski *et al.* (2003) proposed a method to assign digital terrain attributes into continuous classes. They used fuzzy k-means for classifying the continuous landforms. The method finds its usefulness in overcoming the problem of class overlap. The aim

is to describe landslide hazard in roaded and road less areas of a forest. As the size of the data increases and when there are artifacts introduced by the derivation of landform attributes from DEM, the performance of the fuzzy k-means suffers. Burrough et al. Burrough *et al.* (2000) proposed a method to overcome the limitations of the above given model by using spatial sampling, statistical modeling of the derived stream topology and fuzzy k-means using the distance metric. Results are shown on images obtained from Alberta, Canada, and the French pre-Alps.

SVMs is a state-of-art pattern recognition technique whose foundations stem from statistical learning theory Vladimir N. Vapnik (1999). They have widely been used in literature for image segmentation and classification. Chen et al. ying Chen and Yang (2004) presented an algorithm for image segmentation using support vector machines. They used two different sets of features for image segmentation - first, the gray levels of 5x5 neighboring pixels and second, the gray level and grad orientation of 9x9 neighboring pixels. They concluded that to obtain good segmentation results feature set should be chosen appropriately, for instance they achieved superior results using second feature set. Results on these two different set of features using SVM as classifier, are shown on two images in their work. Kim et al. Kim *et al.* (2002) proposed an algorithm for texture classification using multi-class SVM. The gray levels in a window of 17x17 were used as features and multi-class SVM based on one-against-others decomposition is used for classification. They have compared the results with different kernels and by varying window sizes. They concluded that polynomial kernel with degree 5 gives superior results than other kernels. Results are shown on images composed

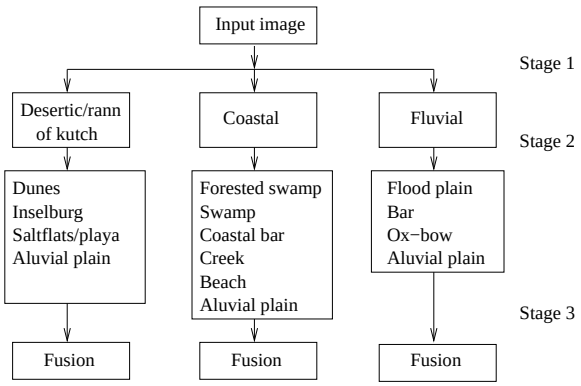


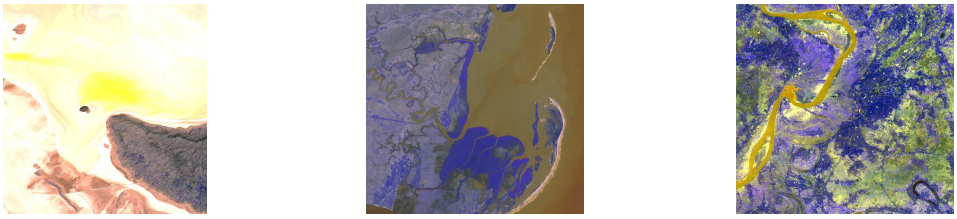
Fig. A.1: Flowchart of the proposed hierarchical landform classification scheme.

of two-five textures.

A.0.1 Overview of Landform Classification

We attempt to solve the problem of landform classification from satellite images using a hierarchical method of segmentation. This is a divide-and-conquer strategy, which divides the complex problem into smaller solvable units. We have obtained training and testing samples of about 20 different landforms. The complexity lies in the fact that the rules governing a decision to obtain a landform widely varies from one to another. For example, some landform such as, dunes, inselberg, flood-plains have very distinct texture features, whereas water bodies, salt flats/playas have distinct band signatures, and others have very distinct shapes (OX-Bow, Meanders and Parabolic dunes) and sizes (swamps, plains etc.). The signatures, adjacency and association rules of these landforms are also fuzzy (uncertain), according to geo-morphologists who provide us with this ground truth.

The task is complex, as no classifier would be able to handle the wide variety of features (texture, color, size and shape), rules of association across all different landforms, and in some cases even for a particular landform. A large set of features



(a) (b) (c)

Fig. A.2: Examples of a few set of satellite images for the three major terrains (a) Desertic terrian/Rann of kutch; (b) Coastal terrian; (c) Fluvial (river side) terrian.

extracted based on certain features will confuse a classifier, which will suffer from the following major disadvantages: correct and weighted combination of features, curse of dimensionality and lack of adequate training samples to capture the large variability within a class/landform.

The complete methodology for Landform classification can now be divided into three stages, which is depicted in Fig. A. At the first stage, a SVM is used to classify an image belonging to either one of the three major terrains types found in the bed of earth (at least in India). These are Desertic/Rann of kutch (we are considering rann of kutch and the desertic in a single category), Fluvial (river side) and Coastal landforms. This is a fundamental assumption in our approach and it works well for certain application, such as trafficability and disaster management for defense, GIS and resource mapping. As we are not interested in land-use patterns, urban areas are not considered. Examples of a few set of satellite images for the three major terrains are given in Fig. A.2. We have assumed that coastal, fluvial and desertic are non-overlapping classes, which we typically found to be true in practical scenarios. For example, dunes can only occur in a desertic area, and coastal bars can only be found in a coastal area. Similarly, OX-BOW patterns can occur only in fluvial zones. This enables us to identify the probable set of

landforms occurring in the input image, only under a particular super-group that has been determined at the first stage. Once the image is classified as desertic, fluvial or coastal, each pixel of the image is classified into the actual landforms with SVM, trained using mean of intensity features, computed as:

$$\mathbf{x}_{i,j} = \{\mu(I^r_{i,j}) \quad \mu(I^g_{i,j}) \quad \mu(I^n_{i,j})\} \tag{A.1}$$

where, $\mathbf{x}_{i,j}$ represents a 3D feature vector corresponding to $(i,j)^{th}$ pixel. $I^r_{i,j}$, $I^g_{i,j}$ and $I^n_{i,j}$ represent intensity values of $(i,j)^{th}$ pixel in Red, Green and NIR bands (the three spectral bands used for processing) of the input image, respectively and $\mu(h)$ represents mean of h in a 3x3 window. Other methods such as moments for shape matching Alt (1962) and pixel connectivity Haralick *et al.* (1992) are used to obtain other major landforms. Finally, outputs of different landforms are fused using a criteria to obtain final classification result. The complete methodology to obtain all landforms and fusion strategy employed to obtain final classification results is described in the following sections.

A.0.2 Description of the methods used for classification

Supergroup classification

This is the topmost stage of the proposed hierarchical classification as shown in Fig. A. A Support Vector Machine (SVM) based classification technique has been adopted in our design for the task of identifying an input image as belonging to one of the desertic, coastal or fluvial landform super-groups. In order to capture and

exploit the variability among the different multi-spectral images belonging to each of the super-groups, histograms of all the 3 bands: Red, Green and NIR bands are used as features for classification. Thus, the SVM-classifier in our case has been trained using histograms of all the three bands of multi-spectral training samples belonging to each one of the three: Desertic, Coastal and Fluvial categories. A high degree of success has been achieved at this stage which will be discussed in Sec. A.0.3

Desertic/Rann of kutch Landform Classification

The flowchart of proposed methodology for the classification of landforms in a desertic/rann of kutch area is shown in Fig. A.3. It can be observed from image shown in Fig. A.8 that saltflats/playas (barren areas with highly saline and alkaline soils, formed through the concentration of mineral residues in salt water) appear bright and inselberg/rocky exposure (a steep ridge or hill left when a mountain has eroded and found in an otherwise flat, typically desert plain) appear dark as compared to dunes/sandy plains (mounds of loose sand grains shaped up by the wind). We exploit this property to differentiate between these three landforms. The steps of processing used for classification are as follows:

1. A multi-class SVM (using one-against others decomposition Kim *et al.* (2002)) trained using mean of pixel intensity values of all three spectral bands, is used to differentiate between dunes/sandy plains, rocky exposure and salt-flats/playas.
2. The output obtained is fused using algorithm described in Sec. A.0.2.

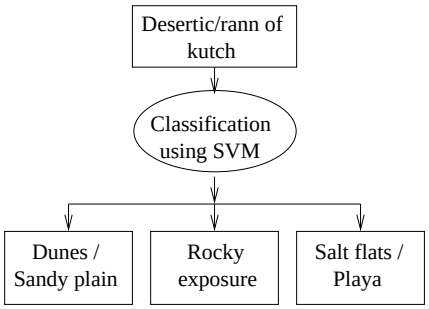


Fig. A.3: Flowchart showing stages of classification of desertic landforms. Features used for SVM are mean of pixel intensity values of all three spectral bands.

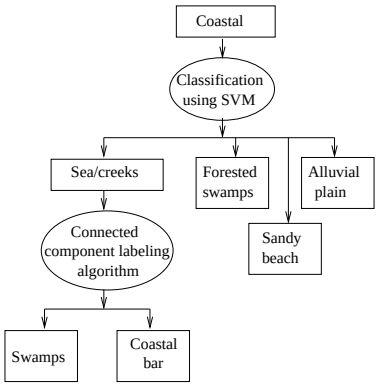


Fig. A.4: Flowchart showing stages of classification of coastal landforms. Features used for SVM are mean of pixel intensity values of all three spectral bands.

Coastal Landform Classification

The flowchart of proposed methodology for the classification of landforms in a coastal area is shown in Fig. A.4. It can be observed from the image shown in Fig. A.9(a) that intensity-based features have a major role to play for extraction of coastal landforms. Association rules have also been employed in order to encode human-knowledge in observing certain key characteristics of coastal landforms within the system. The steps of processing for identification of landform in coastal images are as follows:

1. A multi-class SVM (using one-against others decomposition Kim *et al.* (2002)) trained using mean of pixel intensity values of all three spectral bands, is used to differentiate between sea, forested swamp (a wetland containing trees), sandy beach and alluvial plain.
2. Since coastal bars are landforms that possess unique characteristic property of being enclosed by sea on all sides, a connected component Haralick *et al.* (1992) labeling algorithm is employed to determine all connected components surrounded by sea.
3. Similarly, swamps (a wetland that features permanent inundation of large areas of land by shallow bodies of water) are patches of land that possess high water-content and have been obtained by identifying segments classified as sea in step 1 surrounded by land.

4. The outputs obtained in steps 1,2 and 3 are fused using the algorithm described in Sec. A.0.2, to obtain final classification results.

Fluvial Landform Classification

The flowchart of methodology followed for the classification of landforms in a fluvial area is shown in Fig. A.5. An example of fluvial image is shown in Fig. A.10(a) Since fluvial landforms are produced by the action of river or an active channel, a satellite image taken of a fluvial area mostly contain an active channel within it. The steps of processing for identification of landfroms in fluvial images are as follows:

1. A multi-class SVM (using one-against others decomposition) trained using mean of pixel intensity values of all three spectral bands, is used to differentiate between active channel, flood plain (the low area along a stream or river channel into which water spreads during floods) and alluvial plain.
2. Flood plains in general occur adjacent to active channel, a connected component Haralick *et al.* (1992) labeling algorithm is employed to confirm that all segments identified as flood plains in step 1 are connected to active channel. The segments that are not connected to active channels (river) are classified as alluvial plains.
3. A SVM trained using moment features Alt (1962) (shape) is used to distinguish ox-bow (a U-shaped bend in a river or stream) and active channel among the segments which are classified as active channel in step 1.
4. Since bars are landforms that possess unique characteristic property of being enclosed by active channel on all sides, a connected component labeling algorithm is employed to determine all connected components surrounded by active channel.
5. The outputs obtained in steps 1,2,3 and 4 are fused using algorithm described in Sec. A.0.2 to obtain final classification results.

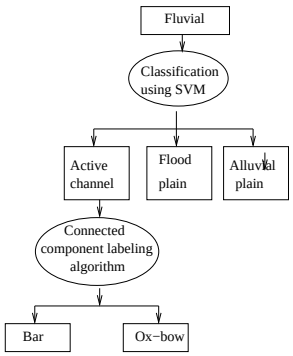


Fig. A.5: Flowchart showing stages of classification of fluvial landforms. Features used for SVM are mean of pixel intensity values of all three spectral bands.

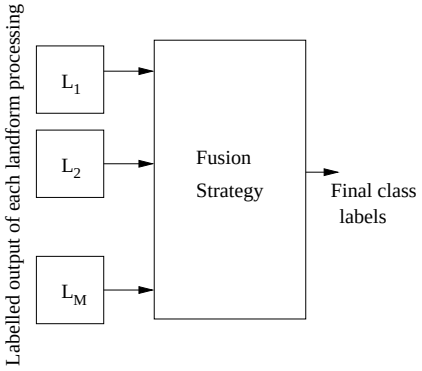


Fig. A.6: Block diagram of the fusion strategy.

Fusion

As mentioned in Sec. A.0.1, an input image may contain multiple landforms within it. However, due to the diverse characteristics (properties) possessed by different landforms, specific processing algorithms have been designed and implemented for extraction of a few landforms. As mentioned above, all segmentation results produced from the different processing algorithms, need to be merged and combined appropriately. We need an efficient process of merging or fusing the outputs of different classifier, as a perticular pixel may be assigned to two or more number of classes by different classifiers.

The strategy adopted by the current system design, attempts to fuse segmentation results of individual landforms on the basis of their association and adjacency phenomena to occur together in nature. Using knowledge acquired from domain experts in geomorphology three adjacency Tables A.1 - A.3 have been built in order to encode the adjacency relationships that exist among different landforms under each super-group. Before fusing results of two different landforms under the same super-group, their corresponding entry in the adjacency table is checked. In

Table A.1: Adjacency table for desertic/rann of kutch landforms.

	Dunes (L_1)	**Rocky exposure** (L_2)	**Saltflats** (L_3)
Dunes (L_1)	-	L_2	L_3
Rocky exposure (L_2)	L_2	-	L_2
Salfflats (L_3)	L_3	L_2	-

case their association is invalid (as indicated by 'NA'), there is no chance whatsoever for the two candidate landforms to occur together and therefore cause an uncertainty. In the other case when their association is valid (as indicated by a landform index with higher precedence), the two landforms under consideration may have a pixel overlap and in such cases their fusion is done by assigning the area of overlap to the landform with higher precedence. The block diagram of the fusion stage has been shown in Fig. A.6.

The fusion strategy adopted for combination of labeled outputs of each landform processing is given below. For combination of two labeled outputs $L_k(X, Y)$ and $L_j(X, Y)$ to form the combined output $O(X, Y)$, (where k and j are the highest labels in precedence among all the class labels assigned before fusion, $1 \leq k, j \leq M$). M being the number of possible landform classes with in that super-class (desertic, fluvial or coastal).

Algorithm for Fusion

1. If landforms k and j do not occur together then output $O(X, Y)$ is given as:

$$O(X, Y) = \underset{1 \leq j \leq M}{\operatorname{argmax}} c_j(X, Y) \tag{A.2}$$

where, c_j is the number of times label j appears in the neighborhood of point (X, Y).

Table A.2: Adjacency table for coastal landforms.

	Swamp (L_1)	**Forested swamp** (L_2)	**Coastal bar** (L_3)	**Beach** (L_4)	**Creek/ sea**(L_5)	**Alluvial plain** (L_6)
Swamp (L_1)	-	NA	L_3	L_4	NA	L_1
Forested swamp (L_2)	NA	-	Both	L_4	NA	L_2
Coastal bar (L_3)	L_3	Both	-	L_4	L_3	L_3
Beach (L_4)	L_4	L_4	L_4	-	L_4	L_4
Creek/Sea (L_5)	NA	NA	L_3	L_4	-	L_5
Alluvial plain (L_6)	L_1	L_2	L_3	L_4	L_5	-

2. If landforms k and j may occur together then output $O(X, Y)$ is given as:

$$O(X,Y) = \left\{ \begin{array}{ccc} L_k(X,Y) & if & prec(k) > prec(j) \\ L_j(X,Y) & if & prec(j) > prec(k) \\ \Psi(X,Y) & if & prec(j) = prec(k) \end{array} \right\} \tag{A.3}$$

where, the function $prec()$ is encoded in the adjacency table and $\Psi(X, Y)$ is the new label assigned to the pixel (X, Y).

The adjacency table for all super-group classes (types of terrains) are shown in Tables A.1 - A.3. Each adjacency table is a symmetric matrix of size $N * N$, where N is the total number of landforms within that super-group. The entries in any adjacency matrix are:

L_i - Landform number with higher precedence among the two adjacent landforms.

N/A - Invalid (not possible).

Both - If both landform occur with equal precedence.

Knowledge of geoscientists is encoded in the table. Experts opinion is conseiderd to form the adjacency matrix.

Table A.3: Adjacency table for fluvial landforms.

	Ox-bow (L_1)	**Active channel** (L_2)	**Bar** (L_3)	**Flood plain** (L_4)	**Alluvial plain** (L_5)
Ox-bow (L_1)	-	NA	NA	L_1	L_1
Active channel (L_2)	NA	-	L_3	L_2	L_2
Bar (L_3)	NA	L_3	-	L_3	L_3
Flood plain (L_4)	L_1	L_2	L_3	-	L_4
Alluvial plain (L_5)	L_1	L_2	L_3	L_4	-

A.0.3 Experimental Results

To verify the effectiveness of the proposed method, experiments were performed on several test images of size 300x300. The SVM used for super group classification was trained using 180 training samples (60 for each class) and tested using 600 samples (200 each class). We obtained 99.2% of classfication accuracy, with a SVM using polynomial kernel of degree 2. Figs. A.7(a)-(c) show examples of correctly classified images of desertic, coastal and fluvial terrians, respectively at stage 1 (supergroup classification). Figs. A.7(d)-(f) show examples of a rann of kutch, coastal, fluvial terrians misclassfied as coastal, fluvial, coastal terrians, respectively at stage 1 (supergroup classification).

Results obtained at stages 2 and 3 using our proposed methodology are shown in Figs. A.8 - A.10. Fig. A.8(a) shows input image of a desertic/rann of kutch area. The corresponding landfroms obtained after classification are shown in: (b) dunes/sandy plains; (c) rocky exposure; and (d) saltflats/playas. Result obtained after fusing the individual outputs is shown in Fig. A.8(e). Fig. A.9(a) shows input image of a coastal area. The corresponding landforms obtained after classification are shown in: (b) coastal bar; (c) forested swamp; (d) swamp; (e) beach;

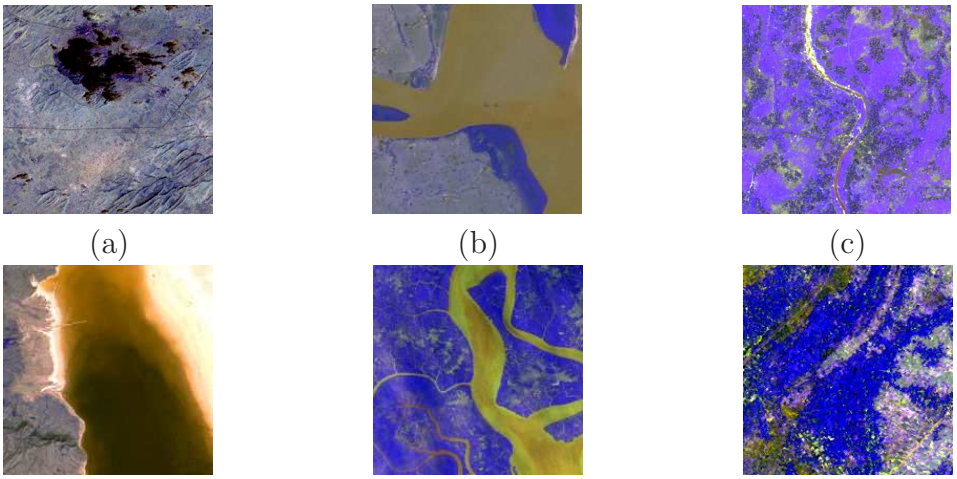


Fig. A.7: Examples of classified ((a)-(c)) and missclassified ((d)-(f)) images at stage 1 (supergroup classification): (a) Desertic Image; (b) Coastal Image; (c) Fluvial Image; (d) Rann of kutch image misclassified as coastal; (e) Coastal image misclassified as fluvial; (f) Fluvial image misclassified as coastal.

(f) sea/creek and (g) alluvial plain. Result obtained after fusing the individual outputs is shown in Fig. A.9(h). Fig. A.10(a) shows input image of a fluvial area. The corresponding landforms obtained after classification are shown in: (b) active channel; (c) flood plain; (d) bar; (e) ox-bow; and (f) alluvial plain Result obtained after fusing the individual outputs is shown in Fig. A.10(g). Although active channel is not a landform but it is shown because other landforms are associated with the active channel. It can be observed from Figs.A.8-A.10, that each landform has been identified correctly in the final output.

A.0.4 Conclusion

A hierarchical approach for landform classification has been proposed in the paper. The proposed hierarchical framework enables us to consider large number of landform classes for segmentation of satellite images. The proposed methodology has been tested on a large number of images. Results show that all major landforms

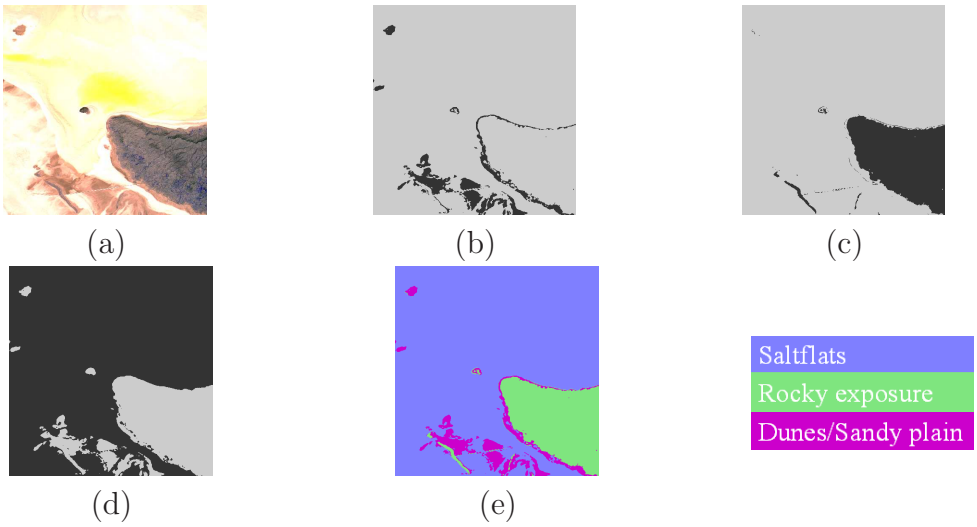


Fig. A.8: (a) Input image consists of desertic landforms (b) Dunes/Sandy plains; (c) Inselburg/rocky exposure; (d) Saltflats/playa; (e) Fused Result.

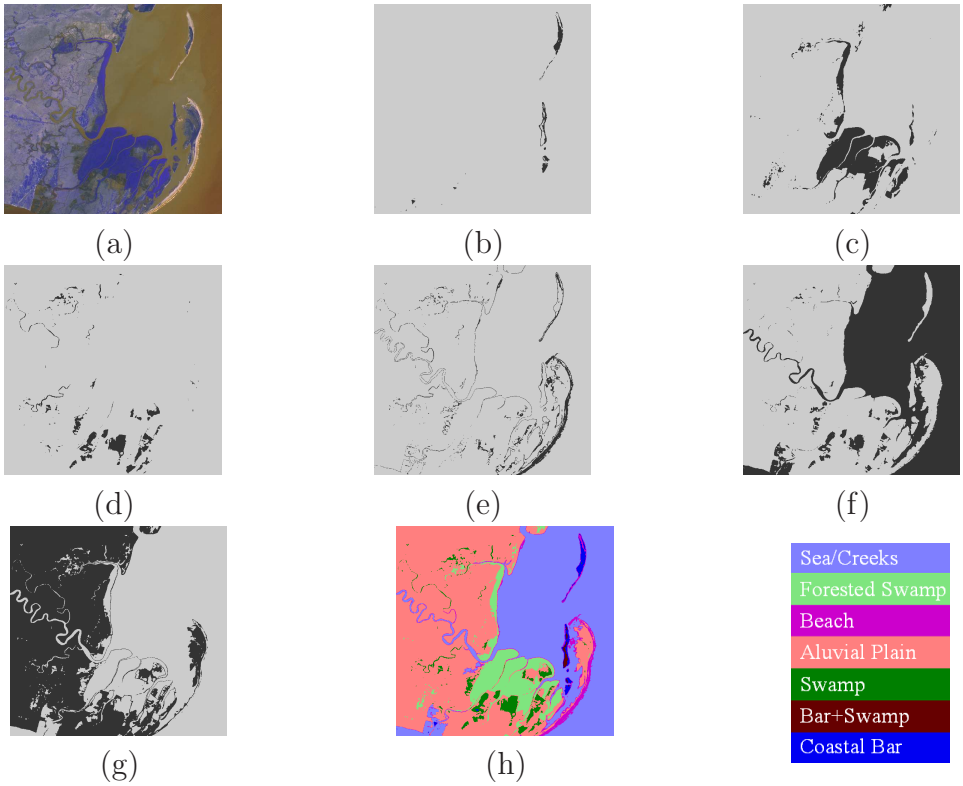


Fig. A.9: (a) Input image consists of coastal landforms; (b) Coastal bar; (c) Forested swamp; (d) Swamp; (e) Beach; (f) Creeks/sea; (g) Alluvial plain; (h) Fused result.

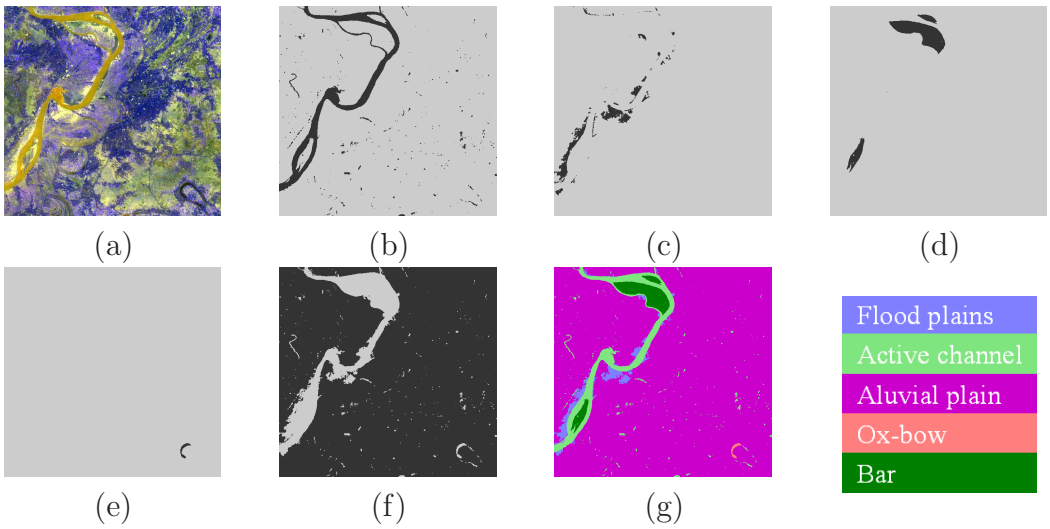


Fig. A.10: (a) Input image consists of fluvial landforms; (b) Active channel; (c) Flood plain; (d) Bar; (e) Ox-bow; (f) Alluvial plain; (g) Fused Result.

have been identified correctly. With the increase in the number of landforms the complexity of the adjacency table will also increase, as well as the super-classes in Fig. A. However the performance of the system has yet to be analyzed for such situations. Future work includes expanding the system to handle more set of landforms, for instance, a method to discriminate among different dunes.

REFERENCES

1. **Alt, F. L.** (1962). Digital pattern recognition by moments. *Journal of the Association for Computing Machinery*, **9**(2), 240–258.

2. **Barahona, F.** (1982). On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical, nuclear and general*, **15**, 3241–3253.

3. **Batlle, J.**, **A. Casals**, **J. Freixenet**, and **J. Marti** (2000). A review on strategies for recognizing natural objects in colour images of outdoor scenes. *Image and Vision Computing*, **18**(6-7), 515–530.

4. **Baxter, R. J.**, *Exactly solved models in statistical mechanics*. Academic Press, 1982.

5. **Berger, A. L.**, **S. D. Pietra**, and **V. J. D. Pietra** (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**(1), 39–71.

6. **Besag, J.** (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. Ser. B*, **36**, 192–236.

7. **Bishop, C. M.**, *Pattern Recognition and Machine Learning.*. Springer, 2006.

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

9. **Burrough, P. A.**, **P. F. M. van Ganns**, and **R. A. MacMillan** (2000). High-resolution landform classification using fuzzy k-means. *Fuzzy sets and systems*, **113**, 37–52.

10. **Cross, G.** and **A. K. Jain** (1983). Markov random field texture models. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, **5**(1), 25–39.

11. **Darroch, J. N.** and **D. Ratcliff** (1972). Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, **43**, 1470–1480.

12. **Dietterich, T. G.**, **A. Ashenfelter**, and **Y. Bulatov**, Training conditional random fields via gradient tree boosting. *In Proceedings of the twenty-first International Conference on Machine learning*. ACM Press, New York, NY, USA, 2004.

13. **Felzenszwalb, P. F.** and **D. P. Huttenlocher** (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, **70**(1), 41–54.

14. **Feng, X.**, **C. K. I. Williams**, and **S. N. Felderhof** (2002). Combining belief networks and neural networks for scene segmentation. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, **24**(4), 467–483.

15. **Fergus, R.**, **P. Perona**, and **A. Zisserman**, Object class recognition by unsupervised scale-invariant learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 02. IEEE Computer Society, Los Alamitos, CA, USA, 2003.

16. **Geman, S.** and **D. Geman** (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, **6**(6), 721–741.

17. **Gentle, A.** (1997). The improved iterative scaling algorithm: A gentle introduction.

18. **Gorsevski, P. V.**, **P. E. Gessler**, and **P. Jankowski** (2003). Integrating a fuzzy k-means classification and a bayesian approach for spatial prediction of landslide hazard. *Journal of Geographical Systems*, **5**, 223–251.

19. **Hanson, A. R.** and **E. M. Riseman**, VISIONS: A computer system for interpreting scenes. *In Computer Vision Systems*. Academic Press, New York, 1978.

20. **Haralick**, **M. Robert**, and **G. L. Shapiro**, *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 1992.

21. **He, X.**, **R. S. Zemel**, and **M. A. Carreira-Perpinan**, Multiscale conditional random fields for image labeling. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 02. IEEE Computer Society, Los Alamitos, CA, USA, 2004.

22. **He, X.**, **R. S. Zemel**, and **D. Ray**, Learning and incorporating top-down cues in image segmentation. *In Proceedings of 9th European Conference on Computer Vision*. Austria, 2006.

23. **Hinton, G. E.** (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, **14**(8), 1771–1800.

24. **Jordan, M. I.** (2004). Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, **19**(1), 140–155.

25. **Kim, K. I.**, **K. Jung**, **S. H. Park**, and **H. J. Kim** (2002). Support vector machines for texture classification. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, **24**(11), 1542–1550.

26. **Kittler, J.** and **J. Föglein** (1984). Contextual classification of multispectral pixel data. *Image and Vision Computing*, **2**, 13–29.

27. **Kumar, S.** (2005). *Models for Learning Spatial Interactions in Natural Images for Context-Based Classification*. Ph.D. thesis, School of Computer Science, The Robotics Institute , Carnegie Mellon University Pittsburgh, PA 15213.

28. **Kumar, S.** and **M. Hebert**, Discriminative fields for modeling spatial dependencies in natural images. *In Advances in Neural Information Processing Systems*. MIT Press, 2003*a*.

29. **Kumar, S.** and **M. Hebert**, Man-made structure detection in natural images using a causal multiscale random field. *In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1. IEEE Computer Society, 2003*b*.

30. **Kumar, S.** and **M. Hebert**, A hierarchical field framework for unified context-based classification. *In Proceedings of International Conference on Computer Vision*. IEEE Computer Society, 2005.

31. **Lafferty, J. D.**, **A. McCallum**, and **F. C. N. Pereira**, Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of International Conference on Machine Learning*. Morgan Kaufmann, 2001.

32. **Li, S. Z.**, *Markov Random Field Modeling in Image Analysis.*. Comp. Sci. Workbench. Springer, 2001.

33. **Mallat, S.** (1996). Wavelets for a vision. *Proceedings of the IEEE*, **84**(4), 604–614.

34. **Nocedal, J.** and **S. J. Wright**, *Numerical Optimization*. Springer-Verlag, New York, 1999.

35. **Ohta, Y.**, A region-oriented image-analysis system by computer. *In Ph.D.*. 1980.

36. **Pearl, J.**, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

37. **Pennock, D. J.**, **B. J. Zebarth**, and **W. Dejong**, Landform classification and soil distribution in hummocky terrain, saskatchewan, canada. *In Proceedings of British Machine Vision Conference*. Norwich, UK, 1987.

38. **Pieczynski, W.** and **A.-N. Tebbache**, Pairwise markov random fields and its application in textured images segmentation. *In Procedings of the 4th Southwest Symposium on Image Analysis and Interpretation (SSIAI-00)*. IEEE Computer Society, Los Alamitos, CA, 2000.

39. **Quattoni, A.**, **M. Collins**, and **T. Darrell**, Conditional random fields for object recognition. *In Advances of Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

40. **Rabiner, L. R.** and **B.-H. Juang**, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

41. **Rao, A. R.** and **R. Jain** (1988). Knowledge representation and control in computer vision systems. *IEEE Expert*, **3**(1), 64–79.

42. **Rosenfeld, A.**, **R. A. Hummel**, and **S. W. Zucker** (1976). Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man and Cybernetics*, **6**, 420–433.

43. **Sarawagi, S.** and **W. W. Cohen**, Semi-markov conditional random fields for information extraction. *In Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

44. **Schneider, R. Z.** and **D. Fernandes**, Entropy concept for change detection in multitemporal sar images. *In European Conference on Synthetic Aperture Radar,*. Cologne, Germany, 2002.

45. **Sha, F.** and **F. C. N. Pereira**, Shallow parsing with conditional random fields. *In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, 2003.

46. **Singhal, A.**, **J. Luo**, and **W. Zhu**, Probabilistic spatial context models for scene content understanding. *In Proceedings of the IEEE International Conference of Computer Vision and Pattern Recognition*. IEEE Computer Society, 2003.

47. **Strat, T. M.**, *Natural Object Recognition*. Springer, 1992.

48. **Sudderth, E. B.**, **E. T. Ihler**, **W. T. Freeman**, and **A. S. Willsky**, Nonparametric belief propagation. *In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 2002.

49. **Sutton, C.** and **A. McCallum** (2005). Fast, piecewise training for discriminative finite-state and parsing models. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts,IR-403.

50. **Sutton, C.** and **A. McCallum**, *In Introduction to statistical relational learning*, chapter An introduction to conditional random fields for relational learning. MIT Press, 2006.

51. **Torralba, A. B.** and **P. Sinha**, Statistical context priming for object detection. *In Proceedings of the International Conference on Computer Vision*. Vancouver, Canada,, 2001.

52. **Vladimir N. Vapnik** (1999). An overview of statistical learning theory. *IEEE Trans. on Neural Networks*, **10**(5), 988–999.

53. **Wallach, H.** (2002). *Efficient Training of Conditional Random Fields*. Master's thesis, School of Cognitive Science,University of Edinburgh.

54. **Williams, C.** and **X. Feng**, Tree-structured belief networks as models of images. *In Proceedings of the 13th conference of Neural Information Processing Systems*. 1998.

55. **Winkler, G.**, *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods: A Mathematical Introduction*. Springer, 1995.

56. **Winston, P. H.** (1970). Learning structural descriptions from examples. Technical report, MIT, Cambridge, Massachusetts.

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

58. **ying Chen, Q.** and **Q. Yang**, Segmentation of images using support vector machines. *In Proceedings of the third International Conference on Machine Learning and Cybernetics*. Shanghai, 2004.

LIST OF PAPERS BASED ON THESIS

Papers Published:

- Pranjal Awasthi, Aakanksha Gagrani, B. Ravindran. "Image Modeling using Tree Structured Conditional Random Fields." in *International Joint Conference of Artificial Intelligence.* AAAI Press, 2007, pp. 2060-2065
- Aakanksha Gagrani, Lalit Gupta, B. Ravindran, Sukhendu Das, Pinaki. Roychowdhury and V.K Panchal."A Hierarchical approach to Landform Classification of Satellite Images using a Fusion Strategy." in *Indian Conference on Vision, Graphics and Image Processing.* Springer, 2006, pp. 140-151.