

# **Project Report**

## **(Multiscale Conditional Random Fields in Image Labeling)**

Navjot Singh  
UBC # 20560041  
nsingh@cs.ubc.ca

### **Abstract**

This report presents work done on the use of CRFs for image labeling. The project is based on the paper titled 'Multiscale Conditional Random Fields in Image Labeling' by Xuming he et al [1]. Contextual information can significantly improve image classification. The paper proposes an approach to include contextual features for labeling images. The features defined in the framework encode information occurring at different scales in the image. The outputs of various features are then combined probabilistically to yield an image labeling. The contrastive divergence algorithm is used to learn features from labeled data and inference is done using the maximum posterior marginals (MPM) criterion. The advantages and limitations of this approach are discussed. The report also presents a brief survey of recent literature on using context in vision.

# 1. Introduction

The term image labeling is used to mean a classification of every pixel of an image into one of a set of predefined classes. A labeled image is thus a segmentation of the image into regions of interest. Such a classification can also be used to identify object classes in the image.

**Context in Vision:** Images contain information at different levels or scales. At the local scale, color and texture can be obtained from pixels in a small locality. Color, for instance, can be used to distinguish the ‘sky’ and ‘vegetation’ classes. However, most real world scenes have many different objects and local information is not enough to identify a pixel’s object class. The following images from the paper by Xuming et al [1] (henceforth referred to as *the paper*) illustrate how local information is insufficient.

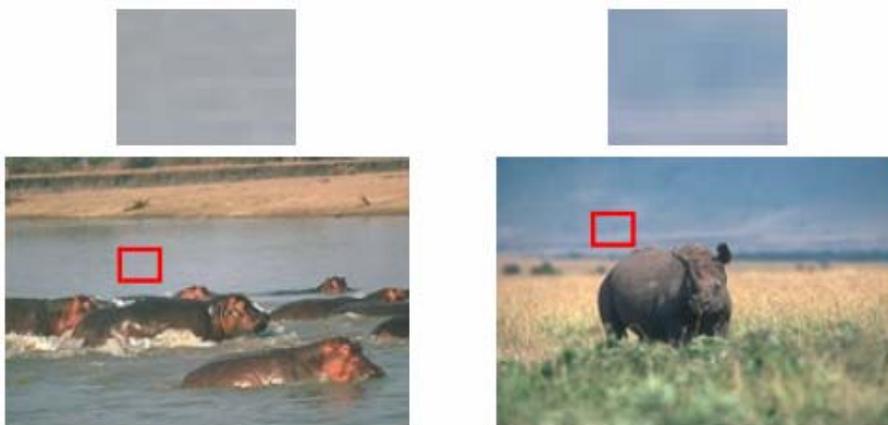


Figure 1: Top: 2 small image patches that are difficult to label.  
Bottom: Context makes it clear that one image patch is water while the other is sky.

Clearly, context can be useful in disambiguating local information. Methods of capturing contextual information from an image are an area of active research. Conditional Random Fields (CRFs) [2] are well suited to capturing context for image classification. Since a CRF can model the posterior probability of labels given the images, it needs smaller training samples than generative models like MRFs that model the joint distribution over labels and images. Context in an image occurs at different scales ranging from the local to the global. The paper uses regional and global features to capture these long range interactions between pixels. Neural nets are used to extract the local information. Probabilistic information from the different features and the neural net is then composed multiplicatively to yield a single labeling. Section 7 gives a general discussion of the problem of capturing context and various solutions proposed thus far.

## 2. Code Developed

The implementation is complete. It comprises 23 files and more than 1100 lines of Matlab code. The manual accompanying the project lists the Matlab code files that comprise the implementation. It also contains step by step instructions for configuring the codes for use. For training data, the MIT-CSAIL [3] dataset has been chosen. Potential users may note that the codes have not been cleaned up and therefore it is important to closely follow the steps described in the manual to set the project up.

**Parameter Estimation:** The contrastive divergence [4] algorithm is used to maximize the conditional likelihood of labels given the feature variables. Additionally, since the model is a Restricted Boltzmann Machine [5], block Gibbs sampling [6] can be implemented. Thus the hidden variables can all be updated in parallel given the label nodes and vice versa. The algorithm requires the sampling to be performed for three steps in the Markov chain. When it can converge, the one step learning algorithm requires significantly less computation. The paper by MacKay [7] discusses examples of a Markov chain is true likelihood is unimodal but the one-step algorithm does not converge to the maximum likelihood parameters.

The Matlab files are as follows:

*RegionalPEstimation.m*, *GlobalPEstimation.m*: Parameter Estimation for Regional and Global parameters respectively

*UpdateHiddenVariables.m*, *UpdateLabels.m*: Block gibbs sampling of the regional hidden variables and associated label nodes respectively

*UpdateHiddenVariablesG.m*, *UpdateLabelsG.m*: Block gibbs sampling of global hidden variables and associated label nodes respectively

*multinomial.m*: Draws a sample from a multinomial distribution

*GlobalPatchToVector.m*, *RegionToVector.m*: Helper files for converting indices from features to label sites

*CalculateLabelPr.m*, *CalculateLabelPrG.m*: Calculates the posterior over labels given the features

### **LocalClassifier**

*MLP.m* / *MLPBatch.m*: Create and train a multi-layer perceptron

**Inference:** The MPM criterion is used to minimize the number of mislabeled sites. Again, Gibbs sampling is used. The sampling is begun from the labeling produced by the neural net.

*GibbsInferenceMPM.m*, *GlobalGibbsInferenceMPM.m*: Inference using the regional and global features respectively

*runLocalClassifier.m* : inference using the neural net

*PoE1.m* : Combines the experts multiplicatively and uses the MPM criterion

**Image Statistics:** The inputs to the neural net are in the form of raw image statistics. There are 30 statistics calculated for each image pixel. These include color, difference-of-Gaussian at 3 scales and quadrature pairs at 4 orientations and 3 scales.

*CalculateImageStatistics.m*: Calculates image statistics.

*CreateStatData.m*: Parses a folder and its subfolders for all images and calls *CalculateImageStatistics.m*. The folder structure is replicated in a different folder to store the statistics.

*Quadrature.m*: For the quadrature pairs

*getInitialLabelingAndStatData.m* : Converts ground truth labeling from the CSAIL annotations to a more convenient format. Additionally, does the same work as *CreateStatData.m*.

### **Other**

*createImagesFolderStructure.m*, *getObjectSegment.m*,

## **3. Background Information**

**Conditional Random Fields:** Representation is a key issue in labeling image data. For images, a generative model that represents a joint distribution over labels and observations is undesirable. Conditional Random Fields (CRFs) [2],[8] can be used, instead, to specify a conditional distribution over labels given an observed image. Inference is performed by choosing the labeling which maximizes this conditional probability for the input image. The entire observation space need not be modeled. Also, arbitrary attributes of the observation data can be captured whereas a joint distribution would require the model to make unnecessary independence assumptions about the data.

**Restricted Boltzmann Machines and Products of Experts:** Products of Experts (PoE) [9] combine many individual expert models by multiplying their probability distributions and renormalizing. PoEs can produce much sharper distributions than the individual expert models. A restricted Boltzmann machine (RBM) [5] is a PoE with

one expert per hidden unit. When the hidden unit of an expert is off, each visible unit is equally likely to be in any of its states. When a hidden unit is on, the weights on its connections to the visible units specify the preferences for the various states of the visible units. Hidden units are conditionally independent given the visible units and vice versa.

## 4. The Multiscale CRF framework

$$\mathbf{X} = \{x_i\}; i \in S$$

$\mathbf{X}$  = observed data from input image

$S$  = set of image sites to be labeled

\* Pixels refer to elements of the image and sites refer to elements of the label field

$x_i$  = local observation at site  $I$

$l_i$  = labeling for site  $i$  from a set of labels or classes  $\mathcal{L}$

**Features:** Label Features are used to encode patterns within label variables. These patterns capture the geometric relationships between objects in an image. The following figures taken from [1] give examples of regional and global label features.

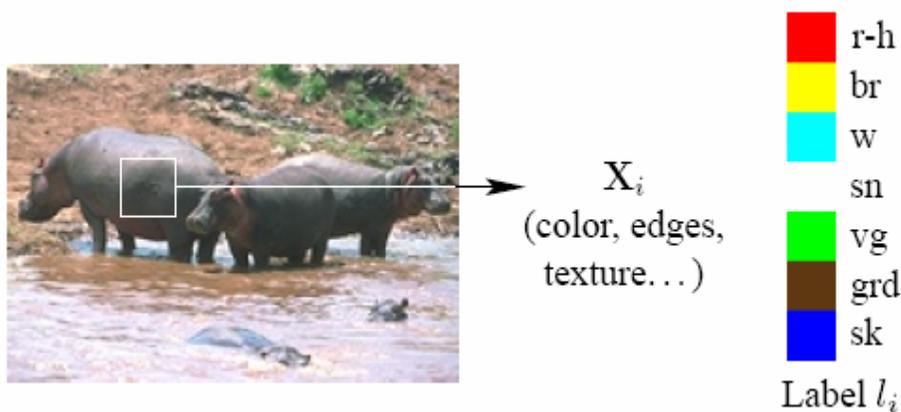


Figure 2: The aim is to associate one label class with each image site  $i$  corresponding to a patch in the image. The column on the right shows 7 different label classes into which a site may be classified.

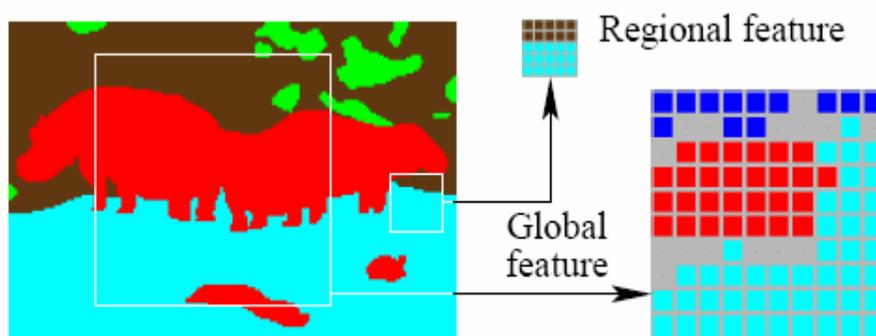


Figure 3: Global and Regional Features

The regional feature shown above encodes the pattern ‘ground above water’. The global feature acts over a larger area and encodes a coarser pattern such as ‘rhino/hippo in water with sky above’. Global features specify common values for a patch of pixels and can also specify ‘don’t care’ cells which are shown as grey here. A don’t care cell specifies a uniform probability distribution over all the label classes.

**The Model as an RBM:** The model can be visualized as an undirected graph with hidden variables and nodes corresponding to image pixels (called label nodes). Corresponding to every label feature, there is a hidden variable. There are no intra-layer connections between labels or between the hidden variables. Therefore, these can be separated into a two layer network called the RBM (see figure 4 taken from the paper). In section 4, we discussed how inference and learning are simplified in such a structure. It is important to note that label nodes are connected to each other via the hidden variables. This allows the model to capture long term interactions between pixels to extract context.

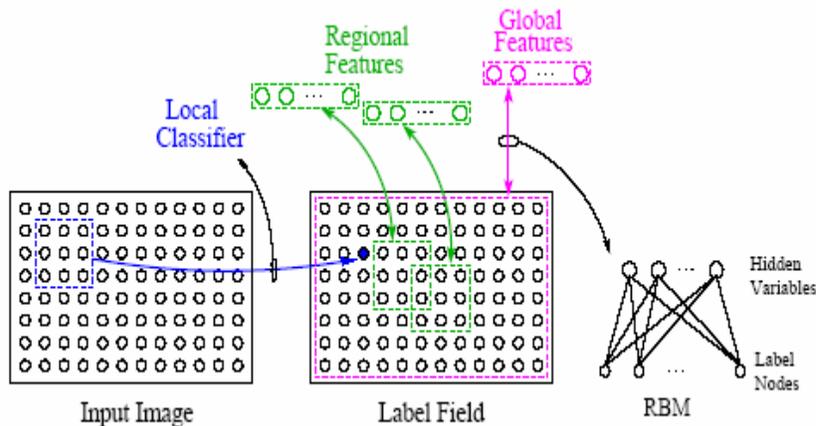


Figure 4: Graphical Model representation

Each of these features specifies a conditional distribution on the label field albeit at different scales. The Multiscale CRF combines the different distributions multiplicatively as follows

$$P(\mathbf{L} / \mathbf{X}) = 1/Z \prod_s P_s(\mathbf{L}/\mathbf{X})$$

$$\text{Where } Z = \sum_{\mathbf{L}} \prod_s P_s(\mathbf{L}/\mathbf{X})$$

This makes the model similar to the Product-of-Experts model with the difference that the component experts here are conditionals on the image  $\mathbf{X}$  and the hidden variables. The different label features produce multiple predictions for every pixel by drawing on information drawn from different scales in the image. These predictions or probability distributions are combined multiplicatively. An important thing to note here is that the label features need not specify the distribution for every label site within a region. A don't care condition can be specified by a uniform distribution over the label classes. The other convenient feature of a product-of-experts is that a number of weak predictions can agree to produce a strong prediction. A more detailed discussion is given by the Hinton [4],[9].

The paper describes components at 3 different scales: a local classifier, regional features and global features (fig. 4).

### 1. Local Classifier

A 3 layer perceptron network is used for a local classifier. The hidden layers have sigmoid transfer functions. There are  $|\mathcal{L}|$  outputs and a softmax activation function. The network outputs can thus be interpreted as a posterior distribution over the label classes. For each image pixel, a 3 x 3 window of neighboring pixel statistics is used as an input to the MLP. The scaled conjugate gradient algorithm is used to train the network to minimize the cross entropy for multiple classes.

The image statistics for each pixel includes color, edge and texture information. For color, the RGB values are changed to CIE Lab\* color space. For edge information, the difference-of-Gaussian filter is applied to the image at three different scales. Finally, for texture, quadrature filter pairs at 4 orientations (0,  $\pi/4$ ,  $\pi/2$ ,  $3\pi/4$ ) and 3 scales are used. For each pixel we get 30 raw image statistics which are used by the neural network as inputs.

The distribution given by the classifier can be written as  $P_C(\mathbf{L} | \mathbf{X}) = \prod_i P_C(l_i / \mathbf{x}_i)$

For each label site  $i$ , the term within the product gives the probability of the label class  $l_i$  for that site.

## 2. Regional Label Features

Regional features represent local relationships between objects such as edges and corners. The image is divided into a series of overlapping regions. The feature for each region has one hidden variable but shares the probability distribution with other regions. The mathematical formulation for these features as an RBM follows.

Let  $r$  index regions,  $a$  index the different regional features within each region, and  $j = \{1, \dots, J\}$  index the label sites within the region  $r$ .

$f(r, a)$ : hidden regional variable

$l(r, j)$ : a vector with  $|\mathcal{L}|$  elements.  $l(r, j, v) = 1$  if node belongs to class  $v$  and 0 otherwise

$w(a, j)$ : parameter connecting  $f(r, a)$  with  $l(r, j)$ . It is a vector with  $|\mathcal{L}|$  elements.  $w(a, j, v)$  represents the preference for class  $v$  for label node  $j$  specified by the feature  $a$

The posterior distribution over the label variables given the hidden variables is written as

$$P_R(l_i = v / \mathbf{f}) = \exp[\sum_{a, (r, j) = i} f(r, a) w(a, j, v)] / \sum_{v'} \exp[\sum_{a, (r, j) = i} f(r, a) w(a, j, v')]$$

In the formula above,  $i$  indexes the sites and the summation is over all regions that contain the site  $i$ .

## 3. Global Label Features

The global features are defined over regions that span the entire image. The image is divided into non-overlapping patches. The parameters for all the label sites within a patch are tied together. This is done to encourage the global features to represent coarser features. Of course, it is also necessary to reduce the number of parameters.

Let  $m$  index patches,  $b$  index the different global features and  $j = \{1, \dots, J\}$  index the label sites within the patch  $m$ .

$g(b)$ : hidden regional variable

$l(m, j)$ : a vector with  $|\mathcal{L}|$  elements.  $l(m, j, v) = 1$  if node belongs to class  $v$  and 0 otherwise

$u(b, m)$ : parameter connecting  $g(b)$  with the entire  $p(m)$ . It is a vector with  $|\mathcal{L}|$  elements.  $u(b, m, v)$  represents the preference for class  $v$  for patch  $m$  specified by the feature  $b$

The posterior distribution over the label variables given the hidden variables is written as

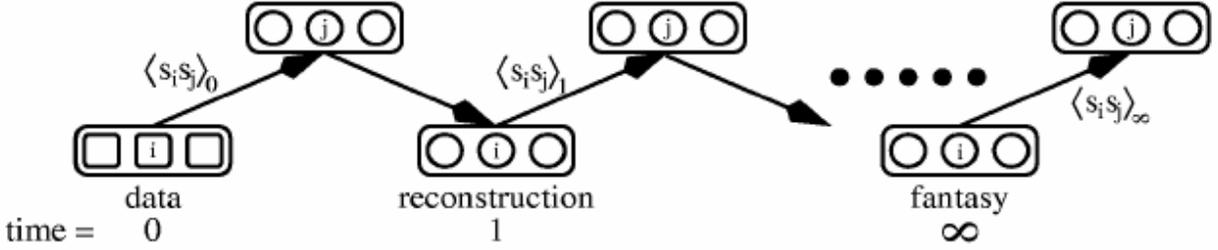
$$P_G(l_i = v / \mathbf{g}) = \exp[\sum_{b, (m, j) = i} g(b) u(b, m, v)] / \sum_{v'} \exp[\sum_{b, (m, j) = i} g(b) u(b, m, v')]$$

Again, the summation is over all patches that contain the site  $i$ .

The three components are combined multiplicatively. All components need not model the entire label field. We would expect that the components would learn to model different aspects of images. The final labeling got by the product maximally satisfies the predictions at every site.

## 5. Parameter Estimation

The Contrastive Divergence algorithm [4] is used to learn the parameters  $\theta = \{\lambda, \{w_a\}, \{u_b\}, \gamma\}$ .  $\lambda$  are the local classifier's parameters and  $\gamma$  is a tradeoff factor to prevent confident but in-correct local classifier predictions from dominating. The training is performed sequentially. The local classifier is trained first, followed by the regional and global features. A joint training of the models is likely to produce a more optimal training but training them sequentially is more efficient. Block Gibbs sampling is used since the model is an RBM. The formulation is changed to maximize the approximate conditional likelihood.



$i$  indexes the data or the label variables and  $j$  indexes the hidden variables. Each arrow corresponds to a block Gibbs's sampling. Hinton's paper [4] describes the algorithm in detail. For our model the formulation can be expressed as

$$\Delta w(i, j) \approx \langle s_i s_j \rangle_0 - \langle s_i s_j \rangle_1$$

## 6. Inference

The maximum posterior marginals (MPM) are used to infer the optimal label configuration  $\mathbf{L}$  given an image  $\mathbf{X}$ . The MPM criterion effectively minimizes the number of mislabeled sites. Again, the posteriors are evaluated by performing block Gibbs's sampling [6]. The sampling is begun at the initial configuration output by the local classifier.

$$l_i^* = \arg \max(l_i) P(L_i | \mathbf{X})$$

## 7. Discussion

**Related work:** Context in an image can be extracted from nearby image data, scene information, and the presence and locations of other objects. The human visual system uses context extensively to identify objects. Part of an object's description can be had from its function. For instance, cars run on ground, birds fly. So an object not on the ground is unlikely to be a car. The context of an object can give us clues about its function. Scene context can also aid object identification. A car would normally be on a street scene and a bird is unlikely to be in the water. Finally, neighboring objects can give important clues too. For instance, a keyboard and a computer screen are likely to occur together.

Neighbor based contextual constraints, as in MRFs, have been shown to be useful even when training data is not fully labeled. Carbonetto et al [10] learn models of objects from captioned images. Kumar et al [11] demonstrated that using nearby labels and nearby data is better than using just the data at the label site. The multiscale CRF framework proposed by Xuming et al captures local site data through the local classifier. The regional label features incorporate the relationships with nearby objects and the global features capture scene context.

Scene based contextual constraints were sought to be captured using a gist representation to focus attention on a particular object class or determine the likelihood of an object's presence [12]. The gist is used as a low-dimensional representation of the entire image that can be used to evaluate the likely objects. Contextual features are used to define context based priors on object classes and help disambiguate information from local features. Murphy et al [13] combine object detection and scene context to resolve local ambiguities. Object location/scale information from the global context is combined with the information from a detector to yield the object probability at a location/state.

Object based context, as captured through mutual boosting by Fink and Perona [14], demonstrates that using the location of other objects in the scene can improve results. Furthermore, using BRFs, Torralba et al [15] constructed contextual models that search for objects in a cascaded fashion. As the easier objects are identified, information

about their location is used to identify other objects in the scene. The approach uses boosting to build progressively stronger classifiers and combines this with a CRF's ability to model the correlation between labels given an input. Labeled data is used to derive template features. Boosting selects from amongst these features to build the graph structure.

The approaches mentioned above try to extract context from neighboring data, the entire scene or neighboring objects. A better understanding of what contextual information may be extracted from a scene and an integrated framework to extract, represent and use this contextual information in conjunction with image data will significantly enhance our ability to process images.

**Multiscale CRF: Advantages and limitations:** The preceding discussion intends to indicate how the approach used in the paper fits in within the larger scheme of research into the use of context in object detection and image labeling. The common motivation is to combine local classification with a model of label relationships. In this approach, the model comprises regional and global features. The structure of the model allows efficient training and inference. Besides, label features form a redundant specification of label predictions and can hence be combined multiplicatively. Due to the same reasons, the features tend to encode simple, geometric relationships between specific label classes. The paper [1] provides a good discussion of the strengths of the approach. In what follows, I discuss certain weaknesses.

The regional and global label features have no access to image statistics used by the local classifier. The model, thus implicitly assumes that the context is independent of local evidence. It will be interesting to train features that have access to local image statistics. The label features are defined at two different scales. In reality, there is a continuum from local to global context. One solution could be to spread out features over different scales. This is unlikely to be the optimal solution. If label features could be automatically learnt over optimal scales from labeled data, it might yield a better representation of context in an image. Fixing two different scales should affect the model's ability to classify objects that consistently occur at a small scale, such as a mouse or a keyboard. Let us consider how the approach will scale to a general purpose image labeling scenario with a large number of label classes. The features encode geometric relationships between object classes. As the number of object classes increases, the number of features required to reasonably capture most such relationships will explode. Experiments with the model and training data should reveal empirical estimates for how the number of features required scales with the number of object classes to be considered.

## 8. Conclusion and Future Plans

The implementation is complete. It has not been executed on the complete CSAIL database, and hence, results haven't been compiled. The codes were tested by using two images drawn from the CSAIL set and resized to 300 x 400 as against the original size of 1200 x 1600. Running the codes on the full sized CSAIL images and evaluating the results obtained is left as future work. The CSAIL database was not the best choice of image database for this project due to two reasons. First, the ground truth labels are very sparse. Hence, large parts of the image are classified into the unlabeled class. This will definitely affect training results. A different approach that was not tried was to train on only the labeled parts of the images. While overcoming the sparse labeling problem, this approach would not provide any contextual information to the model. It is unlikely to produce favorable results. Another problem encountered with CSAIL was the extremely large size of the images. While the images were of good quality, the large size meant that training and inference would take inordinate amounts of time. Resizing the images is not practical since the annotations would also have to be modified accordingly. This process has been automated so that users may choose to resize their images by changing the scaling parameter in the file *getInitialLabelingAndStatData.m*.

As part of the project, recent literature on the use of context in image labeling was surveyed. The previous section provides a brief discussion of this literature along with an analysis of the approach being implemented.

## 9. References

1. X. He, R. Zemel and M. Carreira-Perpiñán, “Multiscale Conditional Random Fields for Image Labeling,” *CVPR*, 2004.
2. J. Lafferty, A. McCallum and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” *ICML*, 2001.
3. The MIT-CSAIL Database of Objects and Scenes <http://web.mit.edu/torralba/www/database.html>.
4. G. E. Hinton: “Training products of experts by minimizing contrastive divergence,” *Neural Comp.* 14:1771–1800, 2002.
5. Y. Freund and D. Haussler: “Unsupervised learning of distributions on binary vectors using 2–layer networks,” *NIPS*, 1992.
6. B. Frey and N. Jovic: “A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models,” U of Toronto Technical Report, 2003.
7. D. MacKay: “Failures of the One-Step Learning Algorithm”, 2001
8. H. Wallach: “Conditional Random Fields: An Introduction”, U of Pennsylvania CIS Technical Report, 2004
9. G. Hinton: “Products of Experts”, [ICANN 99 Vol. 1 pages 1-6].
10. P. Carbonetto, N. Freitas and K. Barnard. “A Statistical Model for General Contextual Object Recognition,” *ECCV*, 2004
11. S. Kumar and M. Hebert, “Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification,” *ICCV*, 2003
12. A. Torralba, “Contextual Priming for Object Detection,” *IJCV*, 2003
13. K. Murphy, A. Torralba and W. Freeman, “Using the Forrest to See the Trees: A Graphical Model Relating Features, Object, and Scenes,” *NIPS*, 2003
14. M. Fink and P. Perona, “Mutual Boosting for Contextual Inference,” *NIPS*, 2003
15. A. Torralba, K. Murphy, and W. Freeman, “Contextual Models for Object Detection using Boosted Random Fields,” *AI Memo 2004-013*, 2004