# Analysis of Markov Random Fields for Semantic Segmentation of Indoor Scenes

Cristóbal Silva
Department of Electrical Engineering
Universidad de Chile
Email: crsilva@ing.uchile.cl

***Abstract*—With the coming of newer and better hardware and software, Deep Learning has taken considerable territory in Computer Vision tasks because of how they automatically learn rich image features only by training with a large number of images. Nonetheless, there are occasions when this is not always possible to do, such as when we don't have enough samples to train such networks, or if we want more control and better understanding of the features that will be used to train the model. For this, the classical supervised learning pipeline is a good option, but it lacks the use of structured information to make better decisions. To this end, we analyze an existing post-processing scheme where a Markov Random Field models interactions between labeled pixels to correct and optimize classification accuracy. We prove that setting certain constraints on the graph structure and choosing good hyper-parameters can lead to an increase in average accuracy for all classes without additional information. On the downside, adding a post-processing step adds extra computational cost that cannot be ignored, and depending on the graph size, this becomes non-optimal for real time applications. All in all, this study formalizes the concept of MRF and how it can be applied to any supervised classification problem where there is structure in the data.**

## I. Introduction

This work mainly focuses on the problem of *semantic segmentation*, a Computer Vision concept in which an input image is given (in this case, an RGB-D image) and an output image is returned containing each pixel labeled with a class. This is a particularly difficult problem because it requires not only characterizing relevant features of many objects, but it also needs to be robust against light conditions, occlusions, rotations, deformations, etc. Usually this is formulated as a supervised classification problem, in which we take the input image, extract relevant features and train a classifier using labeled examples as a reference.

For this particular case, we evaluate a model that is an extension of Ren et. al.'s original work on scene labeling [1], in which he implements a typical classification pipeline (i.e., pre-processing, feature extraction and classifying) while also adding an additional optimization step using *Markov Random Fields* (MRF) and segmentation trees to improve initial results. Since the original source code does not include the MRF optimization and neither does the study go in-depth about its relevance on the final results, the objective is to finish the implementation and prove that the results can be replicated or even improved once a deeper understanding of the graphical model is achieved.

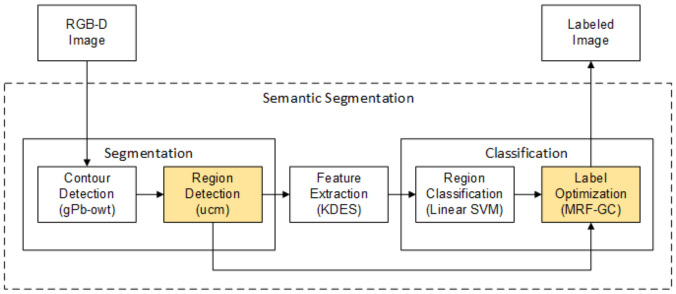

RGB-D Image
Labeled Image
Semantic Segmentation
Segmentation
Classification
Contour Detection (gPb-owt)
Region Detection (ucm)
Feature Extraction (KDES)
Region Classification (Linear SVM)
Label Optimization (MRF-GC)

Fig. 1: Overview of the pipeline. Colored blocks represent intervened or additional code created over the original source.

## II. Methodology

The proposed architecture consists of five steps and can be seen on Figure 1. Segmentation steps (gPb-owt-ucm [2]) are in charge of partitioning the image into groups of pixels that are somehow related between them, these regions are called *segments*; after that, a feature extractor (KDES [3]) transforms these segments into region descriptors that will be used to train a classifier (Linear SVM). Finally, the classifier's output is modeled as an MRF network that will check the labels and verify that they are consistent with the labels of its neighbors, overwriting them if needed.

Inference on MRFs is usually formulated as an energy minimization problem, and for semantic segmentation we can define the class label energy of an image as:

$$E(y_1, ..., y_{|S|}) = \sum_{s \in \mathcal{S}} D_s(y_s) + \sum_{s,r \in \mathcal{N}} V(y_s, y_r) \tag{1}$$

Both terms represent relationships of affinity within the MRF graph. Once the classifier outputs a score for each class and for each segment, we can check all possible configurations and find the one that gives the lowest energy, which is equivalent to maximizing the posterior probability in MRFs.

The *unary term* $D_s(y_s)$ represents the relation between a segment's class label and the observed data. For this particular case we write it as:

$$D_s(y_s) = f(s) \cdot A \tag{2}$$

Where $f(s)$ is the Linear SVM score for segment $s$ and $A$ is the area in pixels. The *pairwise term* $V(y_i, y_j)$ represents the relation between a segment's class label and the label of a neighbor segment. We define this expression as:

$$V(y_s, y_r) = \beta \exp(ucm(s, r)) \cdot c_{sr} \cdot \mathbb{1}_{y_s \neq y_r} \tag{3}$$

Where $ucm(s, r) \in \{0, 1\}$ is a dissimilarity measure between two segments $s$ and $r$ obtained in the segmentation step [4], $c_{sr}$ is the separation length between such segments and $\beta$ is a hyper-parameter. The constraint $\mathbb{1}_{y_s \neq y_r}$ ensures that we penalize with additional energy all configurations that contain contradictions between neighboring segments with different labels but low dissimilarity (i.e. they should have the same label instead). This is sometimes called a smoothing term because the constraint favors same-class neighbors. In MRF literature this is also known as the Potts Model and has a long story of use in MRF for segmentation problems [5].

## III. Results

Table I shows the results for NYUDv1, a dataset containing 2284 RGB-D images of indoor scenes labeled with 13 classes [6]. We can see that once parameters have been tuned correctly (via cross-validation), classification accuracy increases up to 4% compared to the non-optimized case.

| | Segmentation threshold value | | | | |
|---|---|---|---|---|---|
| Avg. Accuracy (%) | 0.06 | 0.08 | 0.12 | 0.17 | 0.21 |
| No MRF | 70.8 | 71.57 | 71.5 | 68.02 | 63.37 |
| MRF ($\beta = 30$) | 75.17 | **75.27** | 73.56 | 69.45 | 64.26% |
| MRF ($\beta = 1000$) | 70.8 | 21.74 | 20.6 | 21.0 | 20.23 |

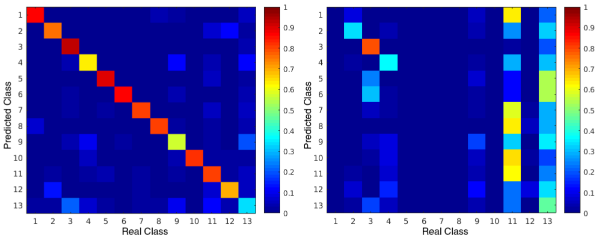
TABLE I: Average Accuracy for three different setups. *No MRF* corresponds to base case and is equivalent to having $\beta = 0$ in the MRF model. The segmentation threshold value indicates the size of the pixel regions from the segmentation step. Lower threshold means smaller segments and higher threshold indicates larger segments.

In terms of computational costs, Table II shows the additional time required for evaluating the entire dataset.

| | Segmentation threshold value | | | | |
|---|---|---|---|---|---|
| Average time (secs) | 0.06 | 0.08 | 0.12 | 0.17 | 0.21 |
| Classification | 40 | 28 | 9 | 5 | 3 |
| MRF optimization | 355 | 240 | 125 | 72 | 54 |
| **Total** | 395 | 268 | 134 | 77 | 57 |

TABLE II: Average computation times for classification (Linear SVM) and optimization (MRF). Notice the increasing time for lower thresholds, corresponding to smaller segments.

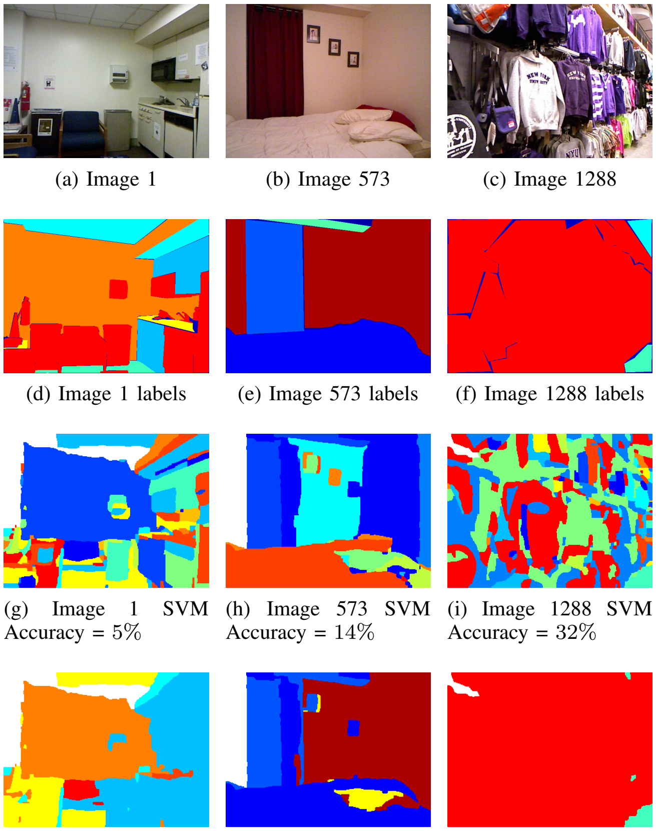
Of course, we are also interested in seeing how this numbers translate to individual class labels when using good and bad hyper-parameters. Figure 2 contains confusion matrices for both $\beta = 30$ and $\beta = 1000$. We can see that if we penalize too much, the exponential in the pairwise term becomes irrelevant and all interactions are heavily penalized, taking the most common class in the image and propagating the label as much as it can, which in this case corresponds to *wall* (11) and *background* (13).



(a)

(b)

Fig. 2: Confusion Matrices for best and worst average cases. (a) MRF for $\beta = 30$. (b) MRF for $\beta = 1000$



(a) Image 1

(b) Image 573

(c) Image 1288

(d) Image 1 labels

(e) Image 573 labels

(f) Image 1288 labels

(g) Image 1 SVM Accuracy = 5%

(h) Image 573 SVM Accuracy = 14%

(i) Image 1288 SVM Accuracy = 32%

(j) Image 1 MRF Accuracy = 44%

(k) Image 573 MRF Accuracy = 75%

(l) Image 1288 MRF Accuracy = 94%

Fig. 3: Comparison between SVM output vs MRF optimization with $\beta = 30$. Accuracy corresponds to rate between correctly detected pixels versus total pixels.

Finally, we analyzed how big is the visual impact in the labeling process before and after post-processing with MRF. Figure 3 shows some relevant samples where the algorithm recovers from wrong segment labels and achieves much better pixel-wise accuracy, as we see greater than 50% differences in some images. The reason behind these big increases is because SVM scores for one segment are probably quite similar for all 13 classes, meaning there is uncertainty behind the result, something that the MRF corrects by analyzing the neighbors.

## IV. Discussion

One of the core concepts we wanted to exploit is the contextual relation between pixels once they have been labeled by a Machine Learning algorithm. Do the initial output labels make sense? Can they be improved without additional information other than the classifier's output? Results evidence that accuracy can indeed be improved once we add local structure information, which in this case is provided by the pairwise term and the $ucm$ dissimilarity measure, though other measures can be used to compare segments (Wu et. al. [7]). This choice was arbitrary because of the original implementation and because it's a consequence of the segmentation step, making it available without doing additional computations.

In terms of labeling accuracy before and after post-processing, it is worth noting that the raw number is affected by many other variables since this is a large pipeline involving parts not detailed in this summary, such as the segmentation algorithm, feature descriptors and classifier hyper-parameters. All these choices will in the end have a greater influence on the results, and the MRF will just correct the most uncertain ones, thus, if there is little uncertainty, post-processing will not do much help.

Finally, we can see there is a trade-off between segment sizes and time/accuracy of results. Using a finer segmentation will increase computational complexity as the graph will be composed by a larger number of segments, making the inference difficult and not so effective since in this case the SVM decreases its accuracy too. Using a coarse segmentation on the other hand will make the graph easier to evaluate, but will lose generalization in the process, as the feature descriptors of big segments will introduce noise by including more than one class features in the area of pixels, something the MRF cannot correct since the segment is considered an atomic unit and is not altered.

## V. Conclusions

We evaluated MRFs on a semantic segmentation context and verified their capacity to improve results with no additional information given during the process. We successfully implemented Ren et. al.'s work and got slightly better results in some cases due to finer parameter tuning, which is in a sense, proof that the work can be replicated and possibly applied to any semantic segmentation output.

Another contribution of this work is the evaluation of time constraints when using graphical models, whose inference methods have a reputation of being slow or difficult due to sometimes being intractable if using the full probabilistic formulation, making them unusable in large scale graphs. Indeed, there is a significant computational cost in this post-processing step due to the number of comparisons needed between segments, but it may be possible to overcome this obstacle by using either low level languages or by parallelizing such comparisons.

One big problem that sometimes can't be solved by these models is overriding decisions in where no label makes sense, and thus, the choice is irrelevant. Examples of these problems were seen a lot because of the noisy dataset, whose main drawbacks included unlabeled pixels or contradictory labels, such as doors that in some frames are classified as windows, or bookshelves that are labeled as walls. This introduces erroneous information from the start and the system cannot recover from it unless more samples are given during training to increase generalization.

Finally, we must be careful with the results obtained, as MRF effectiveness is highly dependent on how graphical model is defined and tune its hyper-parameters accordingly. However, this weakness is also their greatest strength, as it gives full control on how to model the structured relations between data samples and make them as simple or as complex as they need to be. In an era where Deep Learning dominates many computer vision areas, sometimes it is also good to research other ideas or invent new ones that can complement the results of current methods or at least help to gain a better understanding of them.

**Acknowledgments:** This submission was possible thanks to the funding of the GAMES group from the Center for Mathematical Modeling (CMM) of Universidad de Chile, and to my undergraduate thesis advisor, Dr. Pablo Guerrero.

## References

[1] X. Ren and L. Bo and Fox, D. *Scene Labeling: Features and Algorithms* Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, p:2759-2766

[2] Arbelaez, P. and Maire, M. and Fowlkes, C. and Malik, J. *Contour Detection and Hierarchical Image Segmentation* IEEE Trans. Pattern Anal. Mach. Intell., May 2011, p:898-916

[3] L. Bo and X. Ren and Dieter Fox *Kernel Descriptors for Visual Recognition* Advances in Neural Information Processing Systems 23, 2010, p:244-252

[4] Arbelaez, P. *Boundary Extraction in Natural Images Using Ultrametric Contour Maps* Proceedings 5th IEEE Workshop on Perceptual Organization in Computer Vision (POCV), June 2006

[5] Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques* MIT Press, 2009

[6] N. Silberman and R. Fergus *Indoor Scene Segmentation using a Structured Light Sensor* Proceedings of the International Conference on Computer Vision, 2011

[7] C. Wu and I. Lenz and A. Saxena *Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception* Proceedings of Robotics: Science and Systems, 2014