Pursuit of Accurate Semantic Pixel Labeling via Markov Random Field Optimization

Le Lu
University of California, San Diego
l2lu @ ucsd.edu

Abstract

Semantic pixel labeling, which partitions a digital image into multiple semantic segments, is important for computationally understanding the real-world environment. Once it matured, this capability would enable applications such as safe autonomous driving and self-navigation of robots; therefore, it is wanted to improve the accuracy of such automated annotation. Recent advancement of deep convolutional neural networks (CNN) shows promising results by automatically learning discriminative features. This research aims at further improving the accuracy of semantic pixel labeling by applying Markov Random Field optimization on top of the CNN output.

1. Introduction

Pixel labeling, a process of partitioning a digital into multiple segments and labeling each segment as shown below, is commonly used to simplify an image for easier understanding and analysis [1][2]. It could be used to efficiently locate tumors for medical use [3][4] and to detect objects on the road for driving safety.

Recent researches on pixel labeling have shown the increasing attention to this topic in computer vision. Many researchers have proposed novel designs to promote practicality of pixel labeling. For example, Li et al. [5] described that existing methods for pixel labeling tasks usually lead to predictions that are visually implausible and need to be improved. Bertasius el al. [6] introduced a Boundary Neural Field (BNF) to integrate FCN predictions with boundary cues in order to raise the performance of existing methods. Xie et al. [7] then lifted the semantic instance labeling task from 2D into 3D in order to tackle the problem that existing methods are labor-intensive at very large scale and only little labeled data is available. In the research, I used SegNet [8], a pixel-labeling architecture designed by Badrinarayanan et al. to perform pixel labeling on road scene images.

SegNet is made as a fully convolutional neural network architecture. In machine learning, a convolutional neural network (CNN) [9] is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field. With 13 convolutional layers [8], SegNet is also a small yet fast deep learning model, which utilizes a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for pattern classification and for unsupervised feature learning [10]. Also, SegNet, as such a deep learning model, is particularly efficient for road scene understanding [8].

Taking a road scene image as input, SegNet outputs a new image where each semantic component is labeled in a different color. The SegNet model I used in research is trained with more than 3500 images [8] and it categorizes components into 11 different classes including sky, building, pole, road, pavement, tree, sign, fence, car, pedestrian and bicyclist [8]. After being trained, the SegNet model is able to process images similar to those from the training dataset.

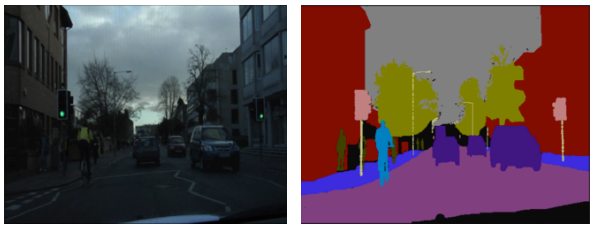

Figure 1: pixel labeling example, with a road scene image on the left and an expected output on the right.

2. Problem

Despite that SegNet has achieved promising results through deep CNN, it is not yet in the category of perfection. In order to quantify the prediction accuracy SegNet can achieve, I measured the global accuracy, class average accuracy [8] and mean intersection over union accuracy [11] with the results as 88.104%, 82.531% and 61.717% respectively. One obvious problem is the existence of extraneous noises on the prediction images. As shown in Figure 2, there are many irrelevant tiny segments scattered over the image; also, some boundaries between different components are ruggedly depicted. This possible disturbance on application and analysis of pixel labeling outputs deserves extra attention, since computers, without being intentionally trained, can hardly distinguish between actual noises and semantic components of small size.

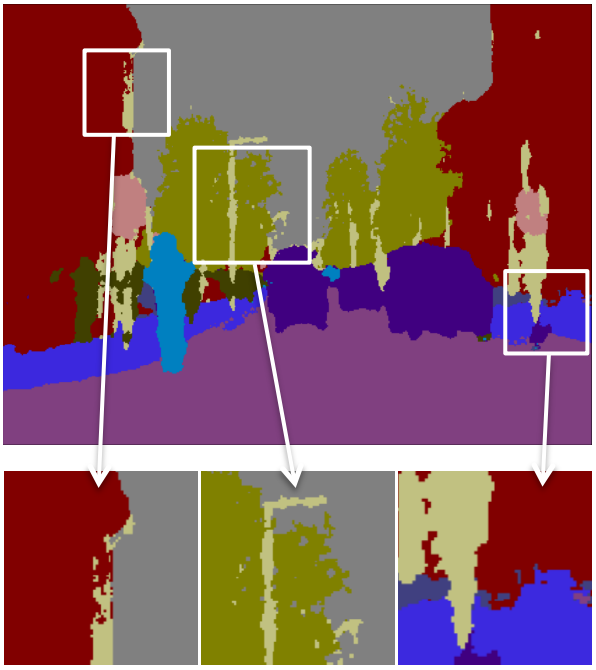

Figure 2: SegNet output without MRF optimization.

3. Purpose of Research

With the consideration of the problem caused by noises on the prediction images, I was inspired to discover a simple and efficient approach in order to improve the accuracy of pixel labeling. The effectiveness of the approach would be evaluated through both visual improvements and statistical improvements.

4. Methods

In order to optimize the existing method, I applied pairwise Markov Random Field (MRF) on top of the CNN outputs. This process is powered by a graph cut optimization tool [12] and a python wrapper [13] for it.

4.1 Pairwise MRF models

Based off the order of interaction between variables, MRF models can be classified into *pairwise models* and *higher-order models* [14]. Considering the scale of the project and the pursuit of simplicity, *pairwise models* were chosen.

Widely used in computer vision, pairwise MRF has the concept that every node depends and only depends on its neighbors. The application of pairwise MRF to pixel labeling is straightforward: the pixel values in an image depend most strongly on those in the immediate vicinity while only having weak correlations with those further away [15].

The pairwise MRF models have two potentials functions working together: *unary potentials* and *pairwise potentials*. Their sum is contributed to the overall energy of pairwise MRF models [14]. More specifically, the pairwise MRF energy has the following form:

$$E(x) = \sum_{i \in V} \theta_i(x_i) + \sum_{(ij) \in E} \theta_{ij}(x_{ij}). \qquad [16]$$

In the formula, E(x) is the overall energy of image x; θ is the potentials value; V is the set of all nodes and E is the set of all pairs of adjacent nodes.

Along with the formula above, every unary node is associated with a potentials value selected from a set of possible ones; also, every pair of adjacent nodes is associated with a pre-calculated potentials value. Pairwise MRF pursues the minimal energy (sum of potentials) for the whole graph.

4.2 Unary Potentials

Unary potentials, also called *singleton potentials* [14], is defined on single variables. In its application to pixel labeling, every single variable is represented by a pixel. The general idea is that for each pixel on the image, we assign it with a potential based off its certainty of correct prediction. The more certain the pixel is labeled correctly, the less potential it will be assigned with.

4.3 Pairwise Potentials

Pairwise potentials, is defined on a pair of variables. Similar to unary potentials, for pairwise potentials, we assign every pair of adjacent pixels with a potential based off its certainty of correct prediction. The more certain the pair is labeled correctly, the less potential it will be assigned with.

4.4 Optimizations

A problem comes with the use of pairwise MRF in pixel labeling: how to effectively quantify the potentials assigned to each pixel and each pair of adjacent pixels.

4.4.1 Optimization of Unary Potentials

Fortunately, to determine the values of unary potentials, SegNet offers a very good tool: *Bayesian SegNet* [17], which is a probabilistic extension to SegNet that outputs a measure of model uncertainty as shown below (Figure 3). On the image, the bright (white) areas are certain areas where the model feels confident about its prediction; the dark (black) areas are uncertain areas where the correctness of prediction is under doubt. This uncertainty distribution eases the quantification of unary potentials.

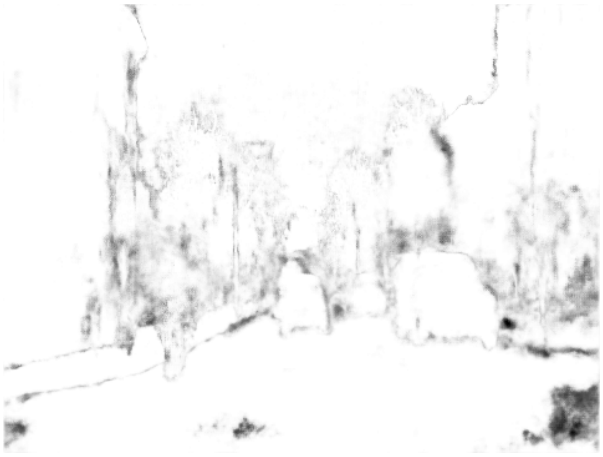

Figure 3: measure of model uncertainty (grayscaled) for the example input.

Specifically, the unary potentials is defined in the following form:

$$E(i) = C_1 + C_2\mu.$$

In the formula, C1 and C2 are two constants (As experiments went, both C1 and C2 were adjusted to be -10.0) and μ is the uncertainty weight.

The uncertainty weight (μ) of a pixel *i* is determined by the darkness of the pixel in the same position on the uncertainty image (Figure 3).

Specifically, the uncertainty weight has the following form:

$$\mu = GI / 255.0.$$

In the formula, GI (grayscale intensity), which ranges from 0 to 255 as darkest to brightest, is the pixel intensity on the grayscaled measure of model uncertainty (Figure 3).

	road	building	pole	sky	pavement	tree	signSymbol	fence	car	pedestrian	Bicyclist
road	**1.0000**	0.0000	0.0031	0.0000	0.0093	0.0002	0.0002	0.0001	0.0047	0.0031	0.0084
building	0.0000	**1.0000**	0.0526	0.0033	0.0033	0.0029	0.0187	0.0032	0.0029	0.0213	0.0048
pole	0.0031	0.0526	**1.0000**	0.0224	0.0101	0.0247	0.0100	0.0027	0.0031	0.0023	0.0006
sky	0.0000	0.0033	0.0224	**1.0000**	0.0000	0.0196	0.0007	0.0000	0.0001	0.0000	0.0001
pavement	0.0093	0.0033	0.0101	0.0000	**1.0000**	0.0008	0.0006	0.0027	0.0008	0.0090	0.0045
tree	0.0002	0.0029	0.0247	0.0196	0.0008	**1.0000**	0.0059	0.0032	0.0013	0.0011	0.0016
signSymbol	0.0002	0.0187	0.0100	0.0007	0.0006	0.0059	**1.0000**	0.0005	0.0008	0.0002	0.0002
fence	0.0001	0.0032	0.0027	0.0000	0.0027	0.0032	0.0005	**1.0000**	0.0013	0.0024	0.0006
car	0.0047	0.0029	0.0031	0.0001	0.0008	0.0013	0.0008	0.0013	**1.0000**	0.0033	0.0028
pedestrian	0.0031	0.0213	0.0023	0.0000	0.0090	0.0011	0.0002	0.0024	0.0033	**1.0000**	0.0006
bicyclist	0.0084	0.0048	0.0006	0.0001	0.0045	0.0016	0.0002	0.0006	0.0028	0.0006	**1.0000**

Table 1: Pairwise Potentials weights (*μp*) table, where the titles of row and column specify the pixel labels. This table is symmetrical since each pair is counted twice (eg. Pair (road-pole) & Pair (pole-road)).

4.4.2 Optimization of Pairwise Potentials

SegNet does not offer prepared information to quantify pairwise potentials. I therefore took advantage of the available training dataset by iterating through 367 ground truth images [8], counting all possible pairs of pixels and recording occurrences in a table. Afterwards, I generated another two tables by normalizing the occurrence table on each row and on each column, respectively, with the potential for the most frequent pair normalized to 1.0. Specifically, the normalized potential is obtained in the following form:

$$V_1(i,j) = N(i,j) / N(j,j).$$
$$V_2(i,j) = N(i,j) / N(i,i).$$

In the formula, V(i, j) is the potential for pair *i-j* in table normalized on each row (V1) and column (V2). N(i, j) is the number of occurrences of pair *i-j* .

By averaging corresponding cell values of the two normalized tables, I obtained the table shown above (Table 1), where each cell represents the pairwise potential weight (*μp*).

Specifically, the pairwise potentials function is defined as following:

$$E(i,j) = \begin{cases} -C & \text{if } i = j. \\ -C\mu_p & \text{if } i \neq j. \end{cases}$$

In the formula, *μp* is the weight and C is a constant that was adjusted to 10 after experiments.

As an exploration of alternatives to optimize pairwise potentials, I examined road scene images from test dataset and compared corresponding pairs of pixels on road scene and output images. The idea behind this approach is that RGB values on road scene images may help correct some wrongly-predicted pixels on output images. For example, a pair of adjacent pixels on a road scene image having the same RGB values should mostly be labeled the same on the output image.

Specifically, I recalculated the pairwise potentials weights table by comparing road scene images with corresponding output images and (a) only counting pairs whose labels reflect either their identical or distinct RGB values / (b) removing pairs whose labels are different and RGB values on road scene images are also different.

However, these two different approaches (a) and (b) did not result in obvious accuracy improvements. Table 6 and 7 at the end of the report show the recalculated pairwise potentials

weight matrices (some weights in Table 7 are negative because data is obtained by subtracting counts from data in Table 1). Although many weights have been changed, the overall affect of these changes is very marginal since pairs of pixels with the same label are still dominating other pairs.

5. Results

As mentioned earlier, the effectiveness and validity of the optimization are evaluated through visual improvements and statistical improvements.

5.1 Visual Improvements

The output after MRF optimization shows significant improvements in smoothness and cleanness: almost all "noise spots" are eliminated; also, boundaries between different components are more smoothly identified. Additionally, pairwise MRF empowers the discriminative denoising: only actual noises are deleted; semantic segments of small size are remained on the image.

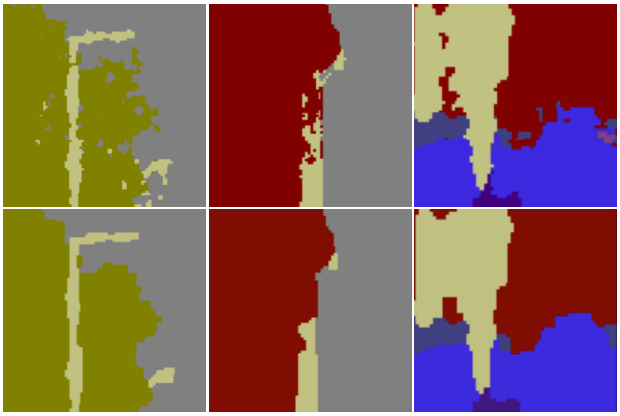

Figure 4: comparison of example segments before and after MRF optimization, with unoptimized segments on the upper side and optimized segments on the lower side.

Additional examples are displayed as Figure 6 at the end of the report.

5.2 Statistical Improvements

Compared to visual improvements, statistical improvements are not as significant but still acceptable. I again measured prediction accuracy in terms of global accuracy, class average accuracy and mean intersection over union (MIOU) accuracy. After multiple attempts at adjusting parameters for unary potentials and pairwise potentials, I achieved the following results while considering the balance between both visual and statistical improvements:

	Global	Class Avg.	MIOU
Before	88.104%	82.531%	61.717%
After	88.235%	82.546%	61.960%
Improvement	**0.131%**	**0.015%**	**0.243%**

Table 2: Statistical Improvements table, statistics are for 233 test images [8] from test dataset.

The SegNet 3.5K model model is not powered with the measure of model uncertainty as the Bayesian SegNet model does; therefore the results above are obtained through MRF with only pairwise potentials optimized. The publicly available Bayesian SegNet model is trained with a smaller dataset and performance improvements of it with optimized unary potentials does not succeed performance improvements of the SegNet 3.5K model with unoptimized unary potentials. Table 4 and 5 display the performance improvements of the available Bayesian SegNet model with and without unary potentials optimized.

Apart from unary potentials and pairwise potentials themselves, there is another ratio parameter that balances the effect of these two terms in calculating the overall energy E(x). Specifically, this parameter is the ratio of weights of unary potentials and pairwise potentials. Table 3 at the end of the report and Figure 5 below show the performance improvements of SegNet 3.5K model with different ratio parameters. I decided to set the ratio to -0.8 since it came with the most possible improvements using experimental data.

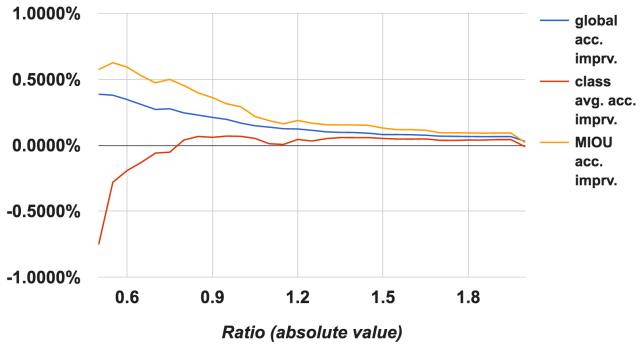

Figure 5: relation between accuracy improvements and the ratio parameter.

6. Conclusion

Although the statistical improvements are not as significant as visual improvements, I believe that it does not deteriorate the effectiveness of the optimization. Firstly, image denoising does not necessarily change many pixels. Rather, it prioritizes the overall cleanness and smoothness, which could be more important than statistical prediction accuracy, depending on how images are utilized or analyzed. Secondly, Despite that SegNet has outstanding performance, it does not promise flawlessness. While the optimization works well with

segments predicted mostly correct, it does not usually lead to positive alternations on incorrectly predicted segments.

Therefore, I believe that my optimization has achieved initial success while there is surely space for further improvements.

7. Acknowledgement

I would like to thank my advisor Professor Yasuyuki Matsushita, for the patient guidance and continuous support of my research. It is my honor to have an advisor who have such immense knowledge and cared so much about my work.

My sincere thanks also go to all members in Matsushita Lab at Osaka University for helping me with my research, especially Assistant Professor Masaki Samejima, for giving me many advices; Hiroaki Santo-senpai, for helping me to set up the environment and Iku Shimizu-senpai, for sharing her SegNet experience with me.

8. References

[1] Linda G. Shapiro and George C. Stockman (2001): "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[2] Barghout, Lauren, and Lawrence W. Lee. "Perceptual information processing system." Paravue Inc. U.S. Patent Application 10/618,543, filed July 11, 2003.

[3] W. Wu, A. Y. C. Chen, L. Zhao and J. J. Corso (2014): "Brain Tumor detection and segmentation in a CRF framework with pixel-pairwise affinity and super pixel-level features", International Journal of Computer Aided Radiology and Surgery, pp. 241-253, Vol. 9.

[4] E. B. George and M. Karnan (2012): "MR Brain image segmentation using Bacteria Foraging Optimization Algorithm", International Journal of Engineering and Technology, Vol. 4.

[5] Ke Li, Bharath Hariharan, Jitendra Malik: "Iterative Instance Segmentation", arXiv:1511.08498 (2016). pp.1

[6] Gedas Bertasius, Jianbo Shi, Lorenzo Torresani: "Semantic Segmentation with Boundary Neural Fields", arXiv:1511.02674 (2016). pp.1

[7] Jun Xie, Martin Kiefel, Ming-Ting Sun, Andreas Geiger: "Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer", arXiv:1511.03240 (2016). pp.1

[8]Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." arXiv preprint arXiv:1511.00561 (2015).

[9] "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.

[10] Li Deng: "An Overview of Deep-Structured Learning for Information Processing", APSIPA ASC 2011 Xi'an (2011). pp.1

[11] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," International Journal of Computer Vision, vol. 111, no. 1, pp. 98–136.

[12] gco-v3.0: Optimizes multi-label energies via the α-expansion and α-β-swap algorithms. http://vision.csd.uwo.ca/code/

[13] pygco: A python wrapper for gco. http://peekaboo-vision.blogspot.jp/2012/05/graphcuts-for-python-pygco.html

[14] Chaohui Wang, Nikos Komodakis and Nikos Paragios, "Markov Random Field Modeling, Inference & Learning in Computer Vision & Image Understanding: A Survey." pp. 10.

[15] Peter Orchard, "Markov Random Field Optimisation". http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0809/ORCHARD/

[16] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006.

[17] Kendall, Alex, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." arXiv preprint arXiv:1511.02680 (2015).

Ratio	global acc. imprv.	class avg. acc. imprv.	MIOU acc. imprv.
-0.50	0.3874%	-0.7531%	0.5751%
-0.55	0.3797%	-0.2797%	0.6262%
-0.60	0.3470%	-0.1923%	0.5928%
-0.65	0.3089%	-0.1299%	0.5269%
-0.70	0.2719%	-0.0591%	0.4738%
-0.75	0.2770%	-0.0525%	0.4997%
-0.80	**0.2457%**	**0.0402%**	**0.4533%**
-0.85	0.2291%	0.0667%	0.3972%
-0.90	0.2115%	0.0602%	0.3618%
-0.95	0.1966%	0.0695%	0.3151%
-1.00	0.1683%	0.0677%	0.2915%
-1.05	0.1478%	0.0521%	0.2184%
-1.10	0.1377%	0.0116%	0.1867%
-1.15	0.1255%	0.0063%	0.1626%
-1.20	0.1235%	0.0443%	0.1883%
-1.25	0.1135%	0.0323%	0.1678%
-1.30	0.1019%	0.0501%	0.1562%
-1.35	0.0979%	0.0586%	0.1542%
-1.40	0.0968%	0.0581%	0.1543%
-1.45	0.0912%	0.0579%	0.1514%
-1.50	0.0813%	0.0517%	0.1305%
-1.55	0.0813%	0.0473%	0.1183%
-1.60	0.0800%	0.0467%	0.1183%
-1.65	0.0762%	0.0483%	0.1141%
-1.70	0.0691%	0.0377%	0.0948%
-1.75	0.0679%	0.0372%	0.0945%
-1.80	0.0664%	0.0398%	0.0936%
-1.85	0.0656%	0.0395%	0.0919%
-1.90	0.0659%	0.0430%	0.0928%
-1.95	0.0659%	0.0430%	0.0928%
-2.00	0.0299%	-0.0111%	0.0213%

***Table 3: performance improvements of SegNet 3.5K model, without any potentials optimization.**

Ratio	global acc. imprv.	class avg. acc. imprv.	MIOU acc. imprv.
-0.50	-0.0619%	-1.8427%	-0.4947%
-0.55	-0.0090%	-1.5034%	-0.4140%
-0.60	-0.0028%	-1.4941%	-0.3577%
-0.65	0.0119%	-1.0602%	-0.2084%
-0.70	0.0071%	-0.6745%	-0.0911%
-0.75	0.0037%	-0.5130%	-0.0204%
-0.80	**-0.0078%**	**-0.4422%**	**-0.0015%**
-0.85	0.0043%	-0.3474%	0.0299%
-0.90	0.0056%	-0.2936%	0.0559%
-0.95	0.0025%	-0.2648%	0.0693%
-1.00	0.0057%	-0.2186%	0.1252%
-1.05	-0.0041%	-0.1328%	0.0978%
-1.10	-0.0032%	-0.1385%	0.0941%
-1.15	0.0189%	-0.0826%	0.1550%
-1.20	0.0122%	-0.1175%	0.1253%
-1.25	0.0110%	-0.0789%	0.1197%

***Table 4: performance improvements of the available Bayesian SegNet model trained with a small dataset, with unary potentials unoptimized.**

Ratio	global acc. imprv.	class avg. acc. imprv.	MIOU acc. imprv.
-0.50	0.0134%	-0.2893%	0.0594%
-0.55	0.0051%	-0.2143%	0.0751%
-0.60	0.0025%	-0.1498%	0.1078%
-0.65	0.0084%	-0.1063%	0.1255%
-0.70	0.0093%	-0.0896%	0.0939%
-0.75	0.0088%	-0.0953%	0.0798%
-0.80	**0.0076%**	**-0.0859%**	**0.0656%**
-0.85	0.0035%	-0.1013%	0.0445%
-0.90	0.0045%	-0.0847%	0.0583%
-0.95	0.0061%	-0.0586%	0.0733%
-1.00	0.0069%	-0.0525%	0.0745%
-1.05	0.0087%	-0.0554%	0.0723%
-1.10	0.0091%	-0.0255%	0.0796%
-1.15	0.0122%	0.0188%	0.0828%
-1.20	0.0119%	0.0367%	0.0686%
-1.25	0.0125%	0.0450%	0.0692%

***Table 4: performance improvements of the available Bayesian SegNet model trained with a small dataset, with unary potentials optimized.**

*red – negative improvement, gray – less than 0.1% improvement, yellow – less than 1% improvement. All statistics are for the same 5 images from test dataset.

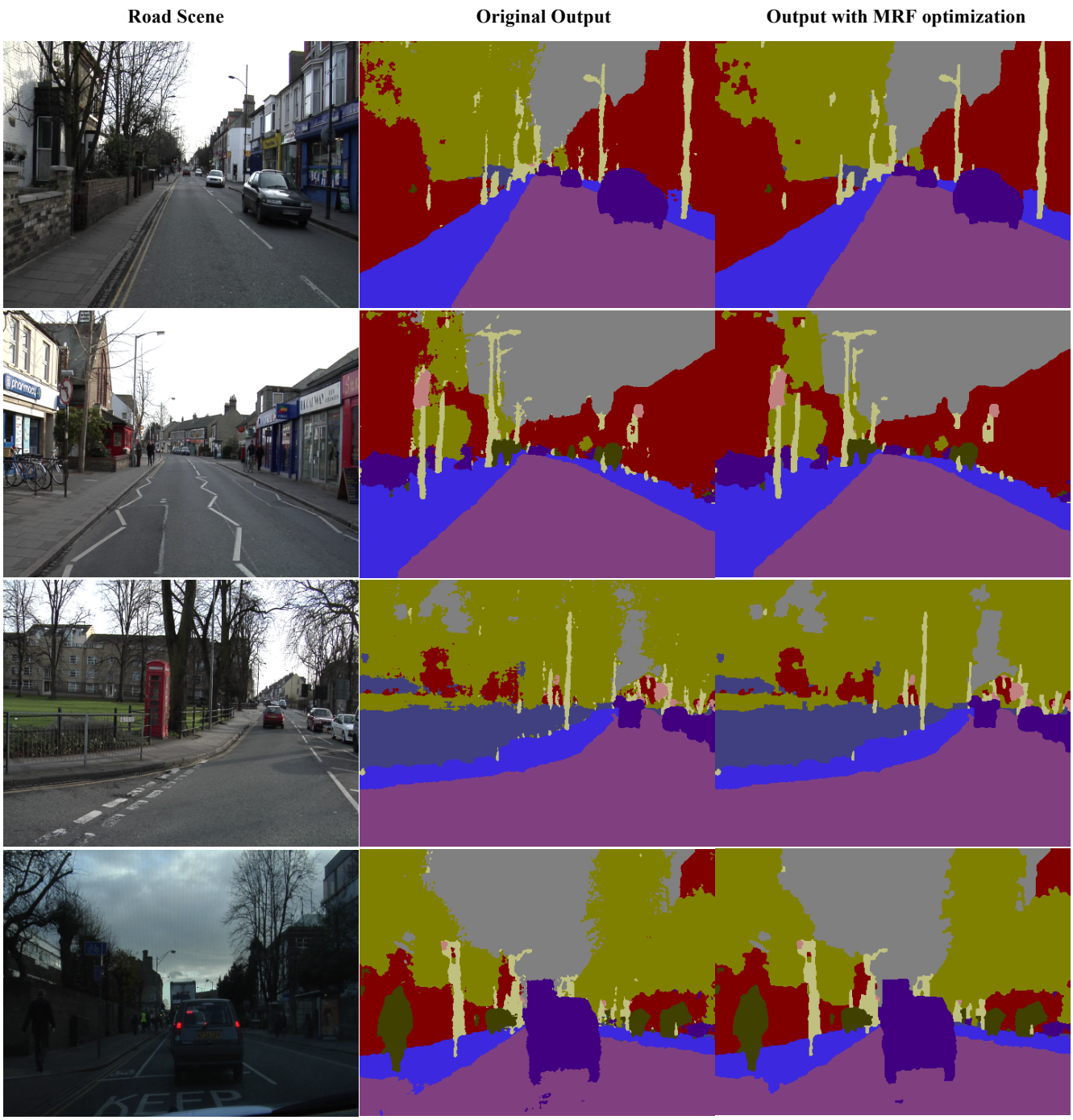

Figure 6: additional examples of MRF optimization. Road scenes (input) are on the leftmost; original outputs without MRF optimization are in the middle; outputs with MRF optimization are on the rightmost.

	road	building	pole	sky	pavement	tree	signSymbol	fence	car	pedestrian	Bicyclist
road	**1.0000**	0.0000	0.0003	0.0000	0.0066	0.0000	0.0000	0.0003	0.0033	0.0005	0.0026
building	0.0000	**1.0000**	0.0207	0.0048	0.0024	0.0034	0.0141	0.0044	0.0020	0.0104	0.0002
pole	0.0003	0.0207	**1.0000**	0.0086	0.0065	0.0094	0.0084	0.0036	0.0026	0.0091	0.0020
sky	0.0000	0.0048	0.0086	**1.0000**	0.0000	0.0079	0.0023	0.0001	0.0007	0.0000	0.0000
pavement	0.0066	0.0024	0.0065	0.0000	**1.0000**	0.0002	0.0000	0.0035	0.0011	0.0077	0.0039
tree	0.0000	0.0034	0.0094	0.0079	0.0002	**1.0000**	0.0028	0.0032	0.0013	0.0012	0.0021
signSymbol	0.0000	0.0141	0.0084	0.0023	0.0000	0.0028	**1.0000**	0.0000	0.0003	0.0003	0.0005
fence	0.0003	0.0044	0.0036	0.0001	0.0035	0.0032	0.0000	**1.0000**	0.0014	0.0019	0.0016
car	0.0033	0.0020	0.0026	0.0007	0.0011	0.0013	0.0003	0.0014	**1.0000**	0.0027	0.0051
pedestrian	0.0005	0.0104	0.0091	0.0000	0.0077	0.0012	0.0003	0.0019	0.0027	**1.0000**	0.0054
bicyclist	0.0026	0.0002	0.0020	0.0000	0.0039	0.0021	0.0005	0.0016	0.0051	0.0054	**1.0000**

Table 6: experimental pairwise potentials weights matrix (a).

	road	building	pole	sky	pavement	tree	signSymbol	fence	car	pedestrian	Bicyclist
road	**1.0000**	0.0000	0.0040	0.0000	0.0120	0.0002	0.0002	0.0001	0.0049	0.0043	0.0097
building	0.0000	**1.0000**	0.0682	-0.0061	0.0040	0.0029	0.0200	0.0031	0.0033	0.0270	0.0057
pole	0.0040	0.0682	**1.0000**	0.0086	0.0124	0.0316	0.0123	0.0028	0.0039	0.0021	0.0007
sky	0.0000	-0.0061	0.0086	**1.0000**	0.0000	-0.0183	-0.0002	0.0000	0.0000	0.0000	0.0001
pavement	0.0120	0.0040	0.0124	0.0000	**1.0000**	0.0010	0.0006	0.0029	0.0010	0.0116	0.0051
tree	0.0002	0.0029	0.0316	-0.0183	0.0010	**1.0000**	0.0063	0.0034	0.0014	0.0012	0.0017
signSymbol	0.0002	0.0200	0.0123	-0.0002	0.0006	0.0063	**1.0000**	0.0005	0.0009	0.0002	0.0002
fence	0.0001	0.0031	0.0028	0.0000	0.0029	0.0034	0.0005	**1.0000**	0.0013	0.0028	0.0005
car	0.0049	0.0033	0.0039	0.0000	0.0010	0.0014	0.0009	0.0013	**1.0000**	0.0042	0.0030
pedestrian	0.0043	0.0270	0.0021	0.0000	0.0116	0.0012	0.0002	0.0028	0.0042	**1.0000**	0.0004
bicyclist	0.0097	0.0057	0.0007	0.0001	0.0051	0.0017	0.0002	0.0005	0.0030	0.0004	**1.0000**

Table 7: experimental pairwise potentials weights matrix (b).