SEMANTIC SEGMENTATION AS IMAGE REPRESENTATION FOR SCENE RECOGNITION

Ahmed Bassiouny, Motaz El-Saban

Microsoft Advanced Technology Labs, Cairo, Egypt

ABSTRACT

We introduce a novel approach towards scene recognition using semantic segmentation maps as image representation. Given a set of images and a list of possible categories for each image, our goal is to assign a category from that list to each image. Our approach is based on representing an image by its semantic segmentation map, which is a mapping from each pixel to a pre-defined set of labels. Among similar high-level approaches, ours has the capability of not only representing what semantic labels the scene contains, but also their shapes, sizes and locations. We obtain state-of-the-art results over Siftflow and Msrc datasets.

Index Terms**—** Scene Recognition, Semantic Segmentation

1. INTRODUCTION

Scene recognition is a not an easy task, due to high variability even within a single scene. Much recent literature have dedicated their effort to approaching it [3] [4] [5] [6]. A typical approach consists of three steps: extracting suggested image representation, encoding the representation in a feature vector, and finally classifying images according to their feature vectors.

As with many computer vision problems, image representations could be divided into: Low level, mid-level, and high level-based methods. Low-level features typically encode physical properties [7] such as color, texture, … etc. Many of the low-level feature vector representations are imported originally from other computer vision applications, e.g. HOG [8] and Sift [9]. [3] for example uses a multitude of low-level feature vectors including Gist, Sift, LBP, texton, besides many others for Scene Recognition. Mid-level features concern extracting spatial and shape relations, nonetheless without regard to semantics [7]. Approaches using mid-level representations generally search for interest points in a picture and tries to match them with similar ones to discover a scene template representing its common interest points to be looked for. [5] for example uses an Exemplar SVM to learn distinctive parts in a scene.

In this paper, we mainly deal with high-level image representation for scene recognition. High-level image representation indicates a mapping from visual representation to meaning [7]. A common approach thus far has been to use pre-trained detectors for objects and background elements in order to find objects in a scene [4] [10] [11]. For example, [10] divides each image into a 10x10 grid and finds objects within each grid cell using 9 pre-trained detectors. Object Bank [4] generalizes to using more detectors and replaced image grid representation with a spatial pyramid. Object Bank used sparse response from object detectors' as representative feature vector. These methods, however, are only capable of indicating objects' existence in image, ignoring other important semantic information about their interrelated sizes, locations or shapes. Similarly, [6] has used weakly supervised classifier to predict objects within an image, but it suffers from the same disadvantage.

We are going into details over Semantic Segmentation techniques, as it is not our target problem. However, using segmentation for scene recognition has been used before, as in [1] where they model images based on their unsupervised segmentation as a GMM model. Additionally, [12] have borrowed NLP's LSA to generate vocabulary representing each image using SIFT descriptor [9]. However, they both obtain no semantic meaning for parts they segment. We argue that supervision on semantic segmentation level lead to a more compact and distinctive image descriptor. Resulting feature vector enforces scene recognition classifier to predict scene category based on combinations of semantically coherent regions, rather than regions with similar feature descriptors.

We adopt the view that high-level visual tasks such as scene recognition require semantic information concerning various scene parts and spatial relations between them [1]. To our knowledge, this is the first work to use pixel-level semantic segmentation as image representation for scene recognition. In the rest of the paper, we first go into more details of our approach, then we describe our experiments and discuss the results.

2. APPROACH

Our approach has two major steps: semantic segmentation, followed by scene classification. In order to avoid confusion, one distinction we would like the reader to distinguish beforehand, is that the list of classes we select from during semantic segmentation is denoted "Labels", whereas the list of classes we select during scene recognition are denoted “Categories”. Our pipeline is show in figure 3.

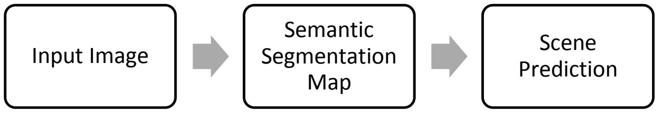


Figure 1: Sketch of our pipeline

3.1 Semantic Segmentation

Semantic Segmentation aims at assigning each pixel in an image to one of a pre-listed set of labels, such as water, person, etc. Formally, given a set of k labels: and a picture represented as a set of n pixel: a semantic segmentation algorithm seeks**.**

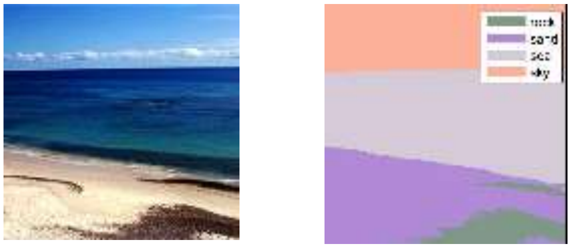
In all our experiments, we used Image Segmentation provided by Ladicky et all [13] – particularly for their well-documented open access code, and state-of-the-art results they achieve on challenging datasets.

3.2 Scene Recognition

With the semantic labeling of an image at pixel level, as in Figure 1, we are now equipped with a semantic map of the image. We propose a number of feature representations encoding the information in the semantic image map, namely: SegCounts, Spatial Pyramid over Seg and Seg + Centrist.

SegCounts Feature Vector: The simplest form of a feature vector considers the ratio of each label in the picture, through the equation:

Hence, the length of our feature vector is the number of labels we are initially choosing from.



Original Image

Possible semantic segmentation

Figure 2: Possible segmentation result. Original image taken from Siftflow dataset [14].

Spatial Pyramid over Seg: In order to equip our Seg feature vector with a sense of locality we used Spatial Pyramid [15]. Essentially, Spatial Pyramid divides an image into many sub-regions over various scales then computes over each histogram of a feature vector, SegCounts in our case. Finally, it compares these histograms using the following weight equation. Specifically, Spatial Pyramid replaces SegCounts’ standard Bag of Features representation with a more sophisticated manner expressing spatial relations between labels.

SegCounts + Centrist: Centrist [16] has the advantage of expressing structural features of an image. Hence, utilizing specific label shapes. Centrist uses the Census Transform, comparing each bit with its 8 local neighbors for having greater or less intensity, as show in Figure 2 in an image from Siftflow dataset. This results in encoding strong constraints of image’s global structure, as Wu et al. argues. A spatial pyramid is then applied to retain locations. We apply Centrist over semantically segmented images, resulting in a feature vector that represents labels’ outer shapes.

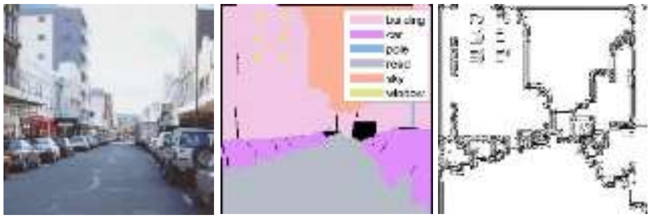


Figure 3: Centrist over an image from Siftflow dataset. Left to right: original image, semantic segmentation result, Centrist result.

4. EXPERIMENTS

We conducted experiments with two datasets: MSRC [17], Siftflow [14]. Below we list the configuration results for the two datasets. Then, we discuss our results and try to answer how changing our experiment parameters, including the semantic segmentation training algorithm training and number of labels, affect our results.

The MSCR dataset has 591 images, covering 23 labels and 21 categories. Siftflow has 2886 images, covering 33 labels and 8 categories. For the two datasets, we used test and train image splits as specified in their original papers [17] and [14] respectively.

We follow the same pipeline for all the experiments. As detailed in section 3, there are two main stages in our pipeline: semantic segmentation followed by scene recognition. First, we train [13]’s classifier using 50% of random images from the dataset. The resulting classifier that is trained over the available labels in the dataset is like a black box that we could use to semantically segment any image. We use that black box against the whole dataset. The resulting semantically segmented images are divided into train and test sets upon which different feature vectors described earlier are applied. For scene classification, we experimented using two different classifiers: a random forest and a linear SVM. We report the better performing classifier out of these two through this section of the paper. Moreover, a spatial pyramid of 3 levels is utilized. Throughout the paper, accuracy reported is the average over the confusion matrix diagonal.

	SegCounts	Spatial Pyramid	Seg + Centrist	Gist	HOG	[1]	[2]
Siftflow	83.5%	86.00%	88.50%	81.50%	80.50%	88.31%	-
Msrc	81.25%	81.64%	79.68%			-	80.60%

Table 1: Experiment results over Siftflow and Msrc datasets.

In table 1 we show our features results for scene recognition over Siftflow and Msrc datasets. We used HOG [8] and Gist [18] feature vectors as baselines. Three essential issues are there to discuss.

4.1 Effect of varying Segmentation Accuracy

First issue is the relation between segmentation accuracy and scene recognition. In order to study this, we used another segmentation algorithm provided by [19], that usually yielded worse results than ALE. In Table 2, we compare the accuracy effect of semantic segmentation on scene recognition. As the table shows, segmentation accuracy plays a very important rule in the resulting scene recognition. SegCounts is clearly proportional to Semantic Segmentation accuracy. This is expected since an incorrect segmentation could result in a total different description for the scene, for example, if a wall is segmented as a mountain.

	Siftflow		Msrc	
	SS	SegCounts	SS	SegCounts
[13]	83.8%	83.5%	91.9%	81.25%
[19]	81.9%	75.8%	66.8%	61.50%

Table 2: Comparison between different performing semantic segmentation accuracy (SS) versus scene recognition result

4.2 Varying semantic segmentation training

In most real world applications, the image set used to train the segmentation algorithm will be different from that used to train the scene recognition algorithm. In order to discover how scene classification would react to using a poorly-trained segmentation classifier, we did the following experiment. In order to do scene classification over Msrc dataset, we trained the semantic segmentation classifier over another different dataset. We take in mind how the number of labels within each dataset could play a role too. Hence, we trained the semantic segmentation classifier over three datasets: Msrc, subMsrc, and Sowerby. Msrc is the usual Msrc dataset described above. We compiled a generic dataset; subMsrc, that is designed to contain the same images of Msrc, but with less number of labels, giving us an indication of the effect of varying the number of labels over the same dataset. With subMsrc, we merged Msrc's 23 labels into 11 labels by naming similar labels the same, e.g. grass and tree labels are both renamed vegetation. Sowerby is a tiny dataset that includes 7 labels. Table 3 shows Siftflow's scene recognition accuracy using different datasets for training its segmentation classifier. As a baseline, we used the results of using same dataset, Siftflow, to train the semantic segmentation classifier.

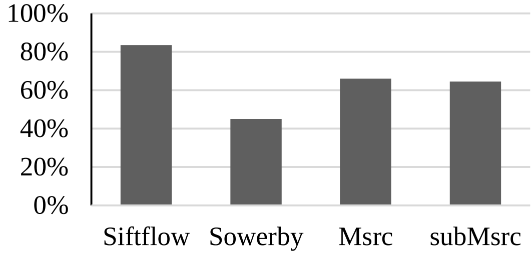


Figure 3: Siftflow's scene recognition results using different training datasets to train segmentation classifier

Interestingly, accuracies for Msrc and subMsrc were quite higher than expected. Also, they were similar despite the disparity in label number. We demonstrate the following explanation. There are two scenarios. First case is when the segmentation classifier is trained and tested on the same dataset. In that case, results are consistent with labels, i.e. a book is usually labeled as book and sea is always labeled as sea. Then, scene recognition classifier is able to draw connections from combination of these labels. The second case, shown in table 3, is when training and testing datasets are different. In that case, the segmentation classifier is trained over Sowerby for example. Then, it is merely exposed to animals, and suddenly it sees city pictures in Siftflow. In that case, it becomes blind and makes its own dictionary. For example, it encodes cars as cows. In the example above, with Msrc, mountain label is encoded as dog. As a result, scene recognition classifier is still capable of making correlations between labels and scenes, even if they do not hold correct semantic meaning. However, as Sowerby's result indicates, too few labels result in confusing the Scene Recognition classifier, for example both sky and sea could be encoded as the same label.

4.3 Optimum number of semantic labels

Although more label in semantic segmentation makes the scene classification classifier's job easier, it could negatively affect the semantic segmentation classifier's results. To explore this area, we did the following experiment. We varied the number of labels used over Siftflow dataset and tested how this affect scene recognition results. Our method for varying the number of labels is based over merging similar labels together and gathering semantically similar labels into one. Table 4 shows results. [Commentary over results – expected that there will be a specific optimal number of labels, we will claim that these labels represent the essential labels and smooth into next part. – ignore what's below for now].

[We consider experiments in in [2] where scene detection is done over two datasets: Siftflow with 33 labels and 2886 images, and Barcelona with about 170 labels and 14,871 images. As seen in table 1, although the average accuracy (diagonal average on confusion matrix) is low for Siftflow and drops drastically for Barcelona dataset, per-pixel accuracy remains at a good shape despite the large increase in the dataset size. The cause behind this consistency in overall accuracy despite low average per-class accuracy is easily seen through the confusion matrix, where the classes with highest availability in the dataset are performing highly versus less available classes with low performance.

	Per-pixel accuracy	Average accuracy
Siftflow	73.2%	29.1%
Barcelona	65.2%	8.7%

Table 5: segmentation results against varying datasets

On the other hand, it is crucial to realize that for Scene Recognition process to be successful, only specific labels need to be correctly labeled, we will denote these as the "Essential Labels". By definition, a combination of different densities of Essential Labels is what defines a Scene. For example, library could be defined as a high density of books and a possible low density of computers, while an office is defined as an average density of both books and computers.

Put together, we use these facts to justify numbers in table 2: the reason why high-level segmentation does not suffer much through the increase number of classes; that Essential Labels' accuracy remains high despite the increase in the number of classes leading to correct scene recognition. In other words, making the process of successful scene recognition is a matter of successful emphasis of Essential Labels in the training set for Semantic Segmentation, despite dataset size.

Numbers in table (2) show state-of-the-art low level features HOG, GLCM, Color Histogram and their combination perform against two different datasets: Siftflow and Msrc. As detailed in the appendix, Siftflow dataset has 8 classes while Msrc dataset has 21 classes. Although low-level features' numbers are close to SegCounts's with the Siftflow dataset, numbers are distinctively less with Msrc dataset. We justify close SegCounts's performance using facts above, by the fact that Essential Labels, e.g. sand, sky and building for Siftflow, are well-emphasized, resulting in high segmentation accuracy for them and in turn successful scene classification.

We reach the conclusion that given a well-trained set of training images, Essential Labels become the labels with high availability, and hence are the ones kept with high segmentation accuracy even on huge datasets. This in turn allows the Scene Recognition classifier to find correct definitions for scenes using densities of Essential Labels.]

5. CONCLUSION AND FUTURE WORK

In this paper we have demonstrated a novel approach by expressing scenes in terms of their pixel-level semantic content. We have shown that state-of-the-art results are easily obtainable and improvable by using feature vector that makes usage of size, shape and location of shapes in an image.

Further testing over larger datasets such as SUN [3] or MIT-indoors [20] is essential. The major obstacle towards such tests would be the availability of enough annotated images for training semantic segmentation classifier over the varying set of labels available. Also, as the table 1 shows, segmentation accuracy plays a very important rule in the resulting scene recognition. One effort towards treating this has been to incorporate pixel-level segmentation accuracy into consideration. This experiment could not be implemented due to the difficulty involved with obtaining uncertainties with graph cut approaches [21]. However, this would also still depend over segmenter accuracy, hence this was not tested further.

5. REFERENCES

1. Akbas, E. and N. Ahuja. *Low-Level Image Segmentation Based Scene Classification*. in *Pattern Recognition (ICPR), 2010 20th International Conference on*. 2010.
2. Urtasun, R., *Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation*, in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, IEEE Computer Society. p. 702-709.
3. Jianxiong, X., et al. *SUN database: Large-scale scene recognition from abbey to zoo*. in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. 2010.
4. Li, L.-J., et al., *Object Bank: An Object-Level Image Representation for High-Level Visual Recognition.* International Journal of Computer Vision, 2013: p. 1-20.
5. Juneja, M., et al., *Blocks that Shout: Distinctive Parts for Scene Classification*, in *IEEE Conference on Computer Vision and Pattern Recognition*. 2013.
6. Kwitt, R., N. Vasconcelos, and N. Rasiwasia, *Scene recognition on the semantic manifold*, in *Proceedings of the 12th European conference on Computer Vision - Volume Part IV*. 2012, Springer-Verlag: Florence, Italy. p. 359-372.
7. Henderson, J.M. and A. Hollingworth, *High-level scene perception.* Annu Rev Psychol, 1999. **50**: p. 243-71.
8. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. in *Computer Vision*

and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. 2005.

9. Lowe, D.G. *Object recognition from local scale-invariant features*. in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. 1999.
10. Vogel, J. and B. Schiele, *Semantic Modeling of Natural Scenes for Content-Based Image Retrieval.* Int. J. Comput. Vision, 2007. **72**(2): p. 133-157.
11. Chunjie, Z., et al. *Beyond local image features: Scene calssification using supervised semantic representation*. in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. 2012.
12. Bosch, A., A. Zisserman, and X. Muñoz, *Scene Classification Via pLSA*, in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Editors. 2006, Springer Berlin Heidelberg. p. 517-530.
13. Ladicky, L., et al., *Graph Cut Based Inference with Co-occurrence Statistics*, in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Editors. 2010, Springer Berlin Heidelberg. p. 239-253.
14. Ce, L., J. Yuen, and A. Torralba, *Nonparametric Scene Parsing via Label Transfer.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2011. **33**(12): p. 2368-2382.
15. Lazebnik, S., C. Schmid, and J. Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. 2006.
16. Jianxin, W. and J.M. Rehg, *CENTRIST: A Visual Descriptor for Scene Categorization.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2011. **33**(8): p. 1489-1501.
17. Shotton, J., et al., *TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation*, in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Editors. 2006, Springer Berlin Heidelberg. p. 1-15.
18. Oliva, A. and A. Torralba, *Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope.* Int. J. Comput. Vision, 2001. **42**(3): p. 145-175.
19. Tighe, J. and S. Lazebnik, *SuperParsing: Scalable Nonparametric Image Parsing with Superpixels*, in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Editors. 2010, Springer Berlin Heidelberg. p. 352-365.
20. Quattoni, A. and A. Torralba. *Recognizing indoor scenes*. in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2009.
21. Tarlow, D. and R.P. Adams. *Revisiting uncertainty in graph cut solutions*. in *CVPR*. 2012. IEEE.