# Semantic Image Labeling with Histograms of Oriented Gradient and Gray Level Co-occurrence Matrix

Jian-She Ma , Tong Liu , Xiu-Tian Huang , Ping Su
Division of Advanced Manufacturing, Graduate School at Shenzhen, Tsinghua University
Shenzhen, China
Email: 709794457@qq.com

***Abstract*-In this paper, we propose a new approach for semantic image labeling by incorporating texture, gradient and color information. In our paper, the texture information is extracted by Gray Level Co-occurrence Matrix (GLCM). The gradient information is obtained by Histograms of Oriented Gradients (HOG). We apply the HOG, GLCM descriptors with color information simultaneously to enrich the image features of different information. To utilizing these features more effectively, we use the Approximate Nearest Neighbors (ANN) algorithm for clustering. After obtaining these information, the Joint Boost algorithm is applied to give an effective classifier by training many weak learner classifiers. At the end, a set of experiments with one descriptor or several descriptors combined are made to evaluating the performance of our method.**

***Keywords-semantic image labeling; Histograms of Oriented Gradients; Gray Level Co-occurrence Matrix; accuracy; Approximate Nearest Neighbors.***

## I INTRODUCTION

Semantic image labeling is a fundamental and challenging problems popularly used for image understanding, searching and robotic visions, etc. The objective of semantic image labeling is to categorize every pixel of a given image into one of several predefined classes. For example, given the indoor scene labeling, we might label each pixel as either ‘desk’, ‘ground’, ‘chair’ or ‘bed’. The result is both a segmentation of the image and a recognition of each segment as a given object class. Figure 1 shows two examples of image labeling. In Figure 1, an image in *a* is a set of pixels $P$ with observed intensities $I_p$ for each $p \in P$ . A labeling $L$ shown in *b* assigns some label $L_p \in \{0,1,2\}$ to each pixel $p \in P$ . Such labels can represent depth (in stereo), object index (in segmentation), original intensity (in image restoration), or other pixel properties. The second example is shown in image *c* and *d*. Image in *c* is the input image and *d* is the result of image labeling of *c*. We use some different colors to represent object classes. For example, the red color in (b) represents ‘building’, and yellow is ‘tree’. The challenge of image labeling is to model the visual variability of a large number of both structured and unstructured object classes, to be invariant to viewpoint and illumination, and to be robust to occlusion [1]. Accuracy and efficiency are two important goals when dealing with large image groups.

The popular conditional random field (CRF) models [3] have been widely used in recent years with two components formulated in an energy function: (a) A local data term encoding pixels-based or superpixel-based classification results [1, 4]. (b) Some pairwise relation terms expressing local or long-range context between labels such as co-occurrence [4-6]. However, in this paper we just consider the local data term (also called unary term) for semantic image labeling. In fact, having obtained the unary term we can apply it to conditional random fields to improve the performance.

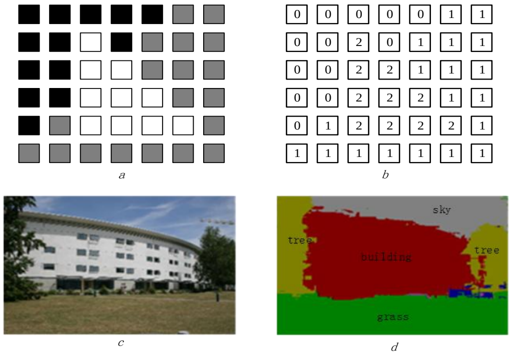

Figure 1. Two examples of image labeling.

The above two figures is one example.[2] And the below two is another example. (a) is an input image and (b) the labeling result of (a). (c) is an input image and (d) the labeling result of (c).

The inspiration of this work comes from two sources. The first source is the idea of using appearance information, whose importance in object detection has been widely discussed such as in [1, 7, 8], and specially the idea of combining cues with the HOG descriptor, e.g., co-occurrence HOG [9], color HOG [10], etc. The second source is the popularity of using GLCM on texture image segmentation. By combining both of them, we can enrich the descriptors for semantic image labeling to enhance the accuracy without decrease the efficiency largely.

The paper is organized as follows. Section 2 and Section 3 introduce the HOG and GLCM descriptors respectively. Section 4 is the algorithm description. The experiment result is discussed in section 5 and Section 6 is our conclusion.

## II HOG FOR IMAGE LABELING

The Histograms of Oriented Gradient (HOG) is a feature descriptor used popularly for the purpose of object detection in computer vision. The technique is to count the occurrences of gradient orientation in localized portions of an image. Comparing with other descriptors such as edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, the difference is in that it is computed over a dense grid of uniformly spaced cells and uses overlapping local contrast normalization so that it can improve accuracy significantly. Navneet Dalal and Bill Triggs [14] first proposed HOG descriptors in 2005. In their work, this descriptor is applied in pedestrian detection over a set of static images.

Following is the detailed explanation of the steps in computing HOG descriptor in our paper.

Firstly, we compute the per pixel gradient and store its orientation and magnitude as shown in Equation (1) and (2).

$$M(x,y) = \sqrt{I_x + I_y} \tag{1}$$

$$\theta(x,y) = \tan^{-1}\frac{I_y}{I_x} \in [0,360^{\circ})\text{or} \in [0,180^{\circ}) \tag{2}$$

where $I_x$ and $I_y$ denote the gradient values of horizontal and vertical direction, $M(x,y)$ is the corresponding magnitude, and $\theta(x,y)$ is the orientation of pixel gradient.

To increase the efficiency only one bin (channel) of the image color is considered, here in our experiment, we only consider the L bin of image (the experimental images are all transformed from RGB to LAB).

Secondly, the cell histogram is generated. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel according to the gradient computation. Here, the histogram channels are spread over 0 to 180 degrees, assuming the gradient is “unsigned”. We divide the orientation into 9 histogram channels and calculate the relevant weight according to the angle by using linear interpolation.

Thirdly, to remove the influence of illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then derived as the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks can be overlapped so that each cell may contribute more than once to the final descriptor. The kind of our block is rectangular R-HOG blocks which are generally square grids and represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. In our experiment, we set $6 \times 6$ pixels per cell with 9 histogram channels and $3 \times 3$ cells per block according to [14]. After calculating the HOG features in our experiment, we will get $9 \times 3 \times 3$ (81) dimension responses for all training pixels. Getting the final HOG descriptor, we normalize the histogram by using L2-norm defined as following so that the HOG features are invariant to geometric and photometric transformations

$$\vec{f} = \frac{\vec{v}}{\sqrt{\|\vec{v}\|_2^2 + e^2}} \tag{3}$$

where $\vec{v}$ is the non-normalized vectors containing all histograms in the block and e is some small constant, e.g., e=0.0001. In fact, $\vec{v}$ is HOG descriptor, which can be described as (4).

$$\vec{v} = (x_1, x_2, \dots x_{81}) \tag{4}$$

where $x_i$ is the value of i-th dimension in the corresponding block.

## III GLCM FOR IMAGE LABELING

Texture property, like the color property, is an important low-level feature descriptor to describe an image. By using Gray Level Co-occurrence Matrix (GLCM), as many as fourteen texture features can be extracted simultaneously. To extract so many features simultaneously is usually very time-consuming. However, by using the approximate nearest neighbor(ANN) algorithm as demonstrated in Section 5 and careful choice of only four features from the fourteen features, the speed of labeling with GLCM can be largely enhanced. In addition, GLCM is very easy to be implemented and has a strong adaptability to apply in different sceneries.

The detail steps of calculating GLCM in our paper are as follows:

1) Transform the original images (RGB) into gray images and let gray level of the image compress to 8 which means the parameter L of GLCM is 8.

2) Calculate and normalize the gray level matrices of each pixel in two directions of $0^{\circ}$ and $90^{\circ}$. Note that the distance of our GLCM is 3 which means the size of calculating GLCM of each pixel is a matrix of $7 \times 7$, and $(i,j)$ is the center of its corresponding matrix. If some pixels of the $7 \times 7$ matrix are out of boundary in the image, we just simply process them with mirror symmetry.

3) Calculate the statics of gray level co-occurrence matrices as texture features, then combine them as feature vectors.

## IV ALGORITHM DESCRIPTION

In [1], a set of features are used which they call texture-layout filters. These features are capable of jointly capturing texture, spatial layout, and textural context. To make the process of boosting learning efficient, the training images are convolved with a 17-dimensional (17D) filter-bank in [1]. In the remaining part of our paper, we use Filterbank to represent the method of this convolution for extracting image features in [1]. Note that the Filterbank method in our paper only concerns the color information instead of combining color with some other features such as location[1]. In our method, the information of responses is enriched by HOG and GLCM descriptors. The detailed procedure is presented in Figure 2.

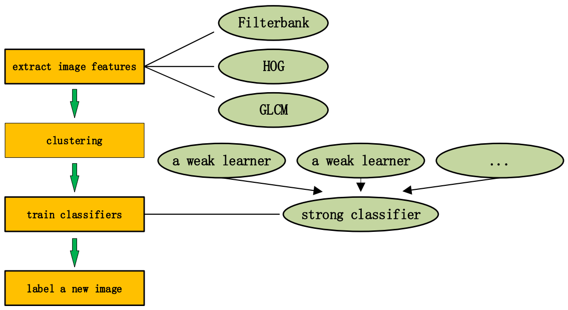

Figure 2. The procedure of our method

By combining the method of Filterbank in [1] with GLCM and HOG, we will get 102D (4+81+17) responses for training pixels. However, because the dimension is 102D, we propose to use the ANN algorithm instead of the Euclidean-distance K -means clustering algorithm. The reason of choosing ANN is that computing exact nearest neighbors in dimensions much higher than 8 seems to be a very difficult task while ANN usually achieve significantly faster running times with relatively small actual errors. Note that we use the well-known Euclidean distance in ANN.

Using the cluster centers obtained as above, the texton map is produced. We denote the texton map as T where pixel i has value $T_i \epsilon \{1,2,\dots,K\}$. Figure 3 is the process of image textonization. We set K as 50 in these experiments.

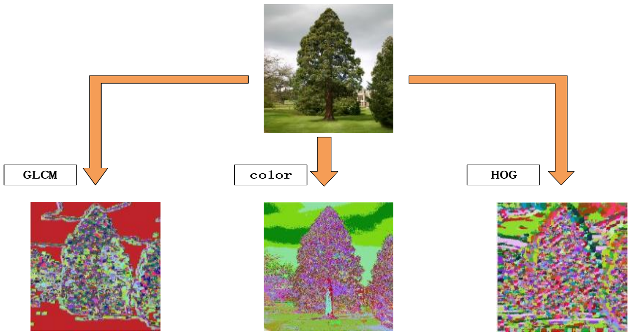

Figure 3. The process of image textonization.

An input is processed with GLCM color and HOG descriptor. The responses for all pixels in training images are then clustered.

After getting the texton map, we use a boosted learning of these features[1]. Based on the texton map, texture-layout filters are built. Each filter is a pair $(r, t)$ of an image region r, and a texton t. Region r is defined in coordinates relative to the pixel i classified. We only investigate rectangular regions for efficiency. The feature (filter) response at location i is the proportion of pixels under the offset region $r + i$ that have texton index t

$$v_{[r,t]}(i) = \frac{1}{\mathrm{area}(r)} \sum\nolimits_{j\epsilon(r+i)} [T_j = t] \qquad (5)$$

The filter responses can be efficiently computed over a whole image with integral images [8]. After calculating the filter responses, we use the Joint Boost algorithm[12] to combine weak learners into a strong classifier. Then we will label a new input image using the strong classifier.

## V EXPERIMENT RESULT

### A. *Image Databases*

The labeled image database used in our experiment is the Microsoft Research Cambridge (MSRC-9) database [1] which is composed of 240 photographs of 10 object classes which is composed of the following classes: grass, tree, cow, sheep, sky, building, airplane, face, car and bike. In our experiment, the database is divided randomly into roughly 45% training, 10% validation and 45% test sets. All experiments are computed on a Windows PC with 4GB memory and an Inter i3-2120 processor clocked at 3.30GHz.

### B. *The comparison of Different Methods on Global Accuracy*

Table 1 list the global accuracy and time consuming of different methods, note that the “F” in the table stands for the textonization used in [1], ‘H’ and ‘G’ represent the methods of using HOG and GLCM descriptors respectively. The number of boosting rounds and ‘weak learners’ in our experiments are 300 and 200 respectively. The last method in Table 1 is our experiment using the code given by [1] on our machine, which only use the texture-layout potentials. We will use ‘[1]-unary’ to denote this method in the remaining parts.

From Table 1, we find that combining ‘H’ and ‘F’ or combining ‘H’ and ‘G” gives a better result on global accuracy comparing to using them alone. Combining three of them has a best performance of global accuracy which is up to 79.90%. However, the consuming time of training is much longer compared to using these methods alone, while the consuming time of evaluating increase slightly. Compared to ‘[1]-unary’, ‘H+F+G’ improve the global accuracy by 10.23%.

TABLE I. THE GLOBAL ACCURACY (%) OF DIFFERENT METHODS

| method | training time per image (m) | evaluation time per image (s) | accuracy(%) |
|---|---|---|---|
| G | 3.55 | 13.18 | 71.13 |
| H | 9.35 | 13.22 | 72.56 |
| F | 4.09 | 13.49 | 77.73 |
| H+G | 9.83 | 14.35 | 79.07 |
| F+G | 4.41 | 14.72 | 74.77 |
| H+F | 10.46 | 14.63 | 78.35 |
| G+H+F | 10.72 | 15.58 | 79.90 |
| [1]-unary | 0.93 | 18.10 | 69.67 |

### C. *The comparison of different Number of Textons*

The graph in Figure 4 plots the resulting pixel-wise segmentation accuracy using 'G' of the boosted classifiers as a function of K, which is the number of textons. From it, we learn that too many textons do not give a better improvement because with too many textons the boosting algorithm starts to overfit. Moreover, the more textons, the more test time consumes.

### D. *Different Methods on Class Accuracy*

Table 2 shows class accuracy of different methods we have operated on MSRC-9 databases. The number of boosting rounds is set 300 and the number of textons is set to 50 in all these experiments. Accuracy values in the table are computed as the percentage of image pixels assigned to the correct class label, ignoring pixels labeled as void in the ground truth. 'H+F+G' make a better result of all these 10 classes comparing to the method of using '[1]-unary' especially in the classes of 'tree', 'cow', 'sheep', 'sky', 'building', 'face'.

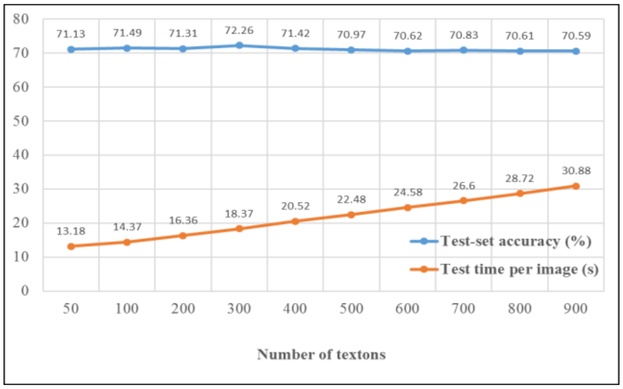

Figure 4. Performance of different number of textons using 'G' method.

TABLE II. THE CLASS ACCURACY (%) OF DIFFERENT METHODS

| method<br>class | H | F | G | H+F | H+G | F+G | F+G+H | [1]-unary |
|---|---|---|---|---|---|---|---|---|
| grass | 68.39 | 83.46 | 68.42 | 84.21 | 71.46 | 86.10 | 83.98 | 82.72 |
| tree | 76.88 | 81.22 | 77.12 | 82.47 | 79.32 | 82.06 | 83.42 | 68.14 |
| cow | 70.02 | 85.24 | 70.58 | 84.86 | 73.86 | 85.05 | 83.96 | 75.60 |
| sheep | 35.38 | 71.95 | 41.77 | 69.19 | 39.62 | 82.09 | 71.34 | 17.01 |
| sky | 83.05 | 86.62 | 83.08 | 86.42 | 87.23 | 86.67 | 88.56 | 77.17 |
| building | 75.12 | 74.91 | 66.94 | 77.58 | 76.90 | 71.36 | 76.95 | 50.70 |
| airplane | 72.85 | 76.73 | 71.04 | 77.41 | 70.44 | 79.22 | 77.75 | 61.55 |
| face | 61.52 | 63.43 | 63.33 | 60.15 | 57.22 | 60.39 | 60.98 | 49.44 |
| car | 77.30 | 62.41 | 74.69 | 67.98 | 76.28 | 64.06 | 72.60 | 70.27 |
| bike | 73.38 | 71.40 | 69.00 | 73.47 | 74.73 | 75.54 | 75.42 | 75.14 |

Figure 5 shows results for different combinations of these 3 methods. The percentage accuracies in Figure 5 (evaluated over the corresponding single dataset) show that each descriptor captures essential information. In this example, combining 'F' with 'H' or 'G' cannot improve the performance compared to using 'F' alone. However, combining three of them makes a better performance by 1.26% compared to using 'F' alone. Combinations of 'H' with 'G' improve the accuracy by 8.67% compared to using 'H' alone and 0.64% compared to using 'G' alone.

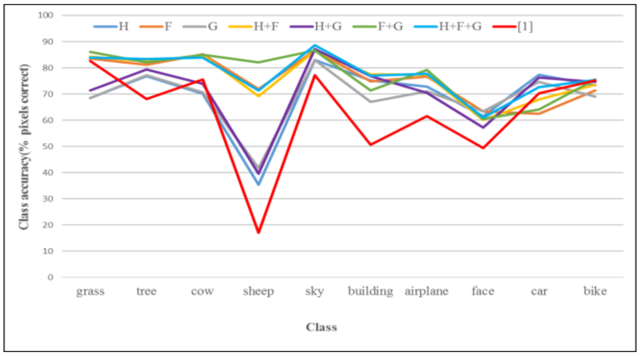

Figure 5. The class accuracy of image labeling with different methods. We also consider “sheep” in the MSRC-9 database.

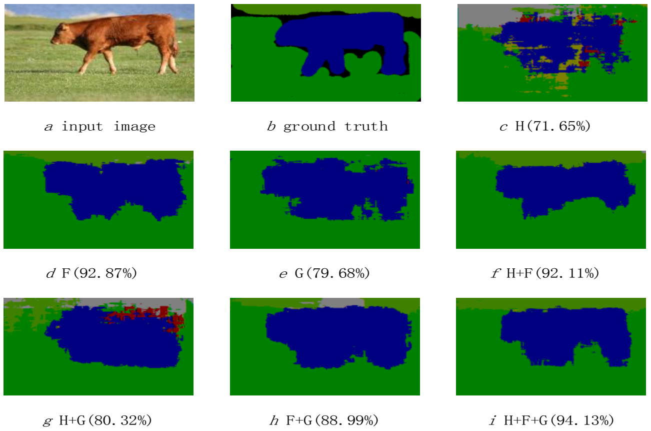
a input image
b ground truth
c H(71.65%)
d F(92.87%)
e G(79.68%)
f H+F(92.11%)
g H+G(80.32%)
h F+G(88.99%)
i H+F+G(94.13%)

Figure 6. Result of different methods.

The input image consists of 3 classes which are ‘cow’, ‘grass’ and ‘tree’. The accuracy is computed based on these 3 classes. Observe that combining all of these methods has a better performance, despite a seemingly small numerical improvement.

## VI CONCLUSION

This paper proposes a new method involving using GLCM and HOG descriptors to segment images for image labeling. In this paper, we have: (i) applied GLCM descriptor to image labeling; (ii) applied HOG descriptor to image labeling. This paper compares the global accuracy and class accuracy of the results with different method’s combinations. We also compare our experiments to the experiment proposed (texture-layout potentials only) in [1] on our machine.

The results of our experiments show that combining GLCM, HOG and filter bank method can improve the global accuracy of image labeling in the MSRC-9 database. This is because every descriptor can capture different information of an image, that is, GLCM obtains the texture information, HOG the gradient information and filter bank the color information. Although the database is composed of only 240 images with only 10 classes (include “sheep”), we believe our methods can also be applied to more classes.

## ACKNOWLEDGEMENT

This work was supported by the new research project of spending on subsidies of Graduate School at Shenzhen, Tsinghua University in 2016 named “The key technology research and industrialization of LED headlamps on smart cars” (JZ20160002).

## REFERENCES

[1] Shotton J, Winn J, Rother C, et al. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation[M]//LEONARDIS A, BISCHOF H, PINZ A. Lecture Notes in Computer Science. 2006:1-15.

[2] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision[M]//FIGUEIREDO M, ZERUBIA J, JAIN A K. Lecture Notes in Computer Science. 2001:359-374.

[3] Lafferty J, McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data: ICML, 2001[C].

[4] Zhuowen T, Xiang B. Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010,32(10):1744-1757.

[5] Gould S, Rodgers J, Cohen D, et al. Multi-class segmentation with relative location prior[J]. INTERNATIONAL JOURNAL OF COMPUTER VISION, 2008,80(3):300-316.

[6] Galleguillos C, Rabinovich A, Belongie S. Object categorization using co-occurrence, location and appearance[M]//2008:8.

[7] Wang X, Han T X, Yan S. An HOG-LBP Human Detector with Partial Occlusion Handling[M]//IEEE International Conference on Computer Vision. 2009:32-39.

[8] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[M]//2001:511-518.

[9] Watanabe T, Ito S, Yokoi K. Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection[M]//WADA T, HUANG F, LIN S. Lecture Notes in Computer Science. 2009:37-47.

[10] Anwer Rao M, Vazquez D, Lopez A M. Color Contribution to Part-Based Person Detection in Different Types of Scenarios[M]//REAL P, DIAZPERNIL D, MOLINAABRIL H, et al. Lecture Notes in Computer Science. 2011:463-470.

[11] S. Arya, D. Mount, N. S. Netanyahu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions[J]. JOURNAL OF THE ACM, 1998,45(6):891-923.

[12] Torralba A, Murphy K P, Freeman W T. Sharing visual features for multiclass and multiview object detection[J]. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2007,29(5):854-869.

[13] Zhou Q, Yan C, Zhu Y, et al. Image labeling by multiple segmentation[M]//IEEE International Conference on Image Processing ICIP. 2011.

[14] Dalal N, Triggs B. Histograms of oriented gradients for human detection[M]// SCHMID C, SOATTO S, TOMASI C. PROCEEDINGS-IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. 2005:886-893.

[15] Jue W. Discriminative Gaussian mixtures for interactive image segmentation[M]//2007:601-604.

[16] Blake A, Rother C, Brown M, et al. Interactive image segmentation using an adaptive GMMRF model[M]//PAJDLA T, MATAS J. LECTURE NOTES IN COMPUTER SCIENCE. 2004:428-441.