# BOG: an extension of HOG by interpreting it as bag of features

Zhouxin Yang
Department of information engineering
Hiroshima University
M113204@hiroshima-u.ac.jp

Takio Kurita
Department of information engineering
Hiroshima University
tkurita@hiroshima-u.ac.jp

## Abstract

*Histogram of orientated gradient (HOG) is widely used as a local feature descriptor in bag of features (BOF) method, whereas, few studies are conducted to discover the relationship between them. In this paper, we exploit this relationship and reveal that the construction method of descriptor in blocks in HOG can be treated as a variant of BOF method. Based on this interpretation, we propose a new descriptor termed as bag of gradient (BOG), which can be viewed as an extension of HOG, by incorporating principles used in BOF, such as the preservation of locality. Experiment results show that BOG significantly reduces the error rate in comparison to HOG in pedestrian detection.*

## 1 Introduction

Recent advances in the bag of features (BOF) approach have significantly contributed to the progress of BOF-based image classification systems. These advances focus on the different phases of BOF, such as the extraction of features [1], the coding of features [2, 3], the pooling of features [4, 5], and the spatial information preservation [6].

The histogram structure of the coefficient descriptor over certain vocabulary generated by BOF is similar to that of many local feature descriptors, such as HOG [7] and SIFT [8]. This resemblance tempts us to bridge those descriptors and BOF and leverage recent advances in BOF. Nonetheless, few studies are contributed to this work so far, and HOG and SIFT are used merely as local feature descriptor in the feature extraction phase of BOF.

In this paper, we build the relationship between HOG and BOF by pointing out that the method used to construct the descriptor of blocks in HOG is similar to that used to construct the coefficient descriptor in BOF. On the basis of this interpretation, we propose an extended local feature descriptor of HOG, called bag of gradient (BOG). As compared to HOG in the experiment of pedestrian detection, our proposed descriptor, which leverages the recent advances in BOF, reduces the error rate from 19.22% to 7.98% when $FPPW = 10^{-3}$ (Fig. 1).

Because BOG extends HOG by combing principles of BOF, it can be easily embedded into current frameworks using HOG. Moreover, further advances in HOG and BOF can be applied to boost the performance of BOG.

The rest of this paper is organized as follows. In section 2, we show how the method for constructing the descriptor of blocks in HOG can be viewed as a variant of those used to construct the coefficient descriptor in BOF. In Section 3, we introduce recent principles proposed for BOF into the construction of descriptor of blocks and propose a new local feature descriptor termed as BOG. Section 4 shows experiments in pedestrian detection to evaluate the performance of BOG. Section 5 gives the conclusion of this work.

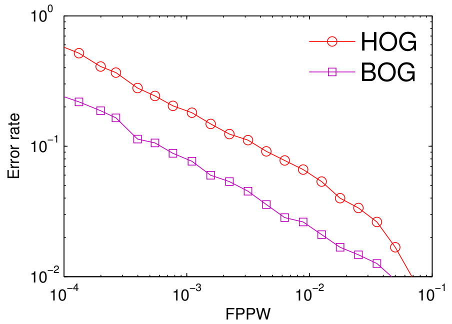

Figure 1. BOG remarkably reduces the error rate in the pedestrian detection task in comparison with HOG.

## 2 The relationship between HOG and BOF in descriptor construction

In this section, we show how the method used to construct the descriptor of blocks in HOG follows that used in the construction of coefficient descriptor in BOF.

### 2.1 Feature extraction

Let $\mathbf{p}_i \in \Re^n$ denote a feature extracted from some location, $\mathbf{B} \in \Re^{r\times n}$ a vocabulary with $r$ visual words, $\mathbf{b}_j$ the $j$th visual word in this vocabulary. In this paper, the feature is the gradient extracted from pixels, therefore, $n = 2$. Let $(p_i^\theta, p_i^m)$ and $(b_j^\theta, b_j^m)$ denote the orientation value and magnitude value of $\mathbf{p}_i$ and $\mathbf{b}_j$ respectively. If we scale the range of orientation dimension and magnitude dimension into $[0, \pi)$ and $[0, L]$, then $p_i^\theta, b_j^\theta \in [0, \pi)$ and $p_i^m, b_j^m \in [0, L]$. Let $u_{ij}$ denote the coefficient of $\mathbf{p}_i$ to $\mathbf{b}_j$.

In HOG, the orientation value $p_i^\theta$ of the gradient of a pixel in the block is treated as the feature ($n = 1$) extracted from this pixel. The extraction of those features is done densely for every pixel in this block.

### 2.2 The building of vocabulary

The vocabulary $\mathbf{B}$ in HOG is manually defined, which differs from other BOF methods that create the vocabulary by some clustering techniques such as k-means to obtain data-driven vocabulary or by dictionary-learning methods to obtain structured vocabulary [9]. Visual words in $\mathbf{B}$ are defined to be evenly distributed along the orientation dimension. Hence, a visual word $b_j^\theta$ in $\mathbf{B}$ can be simply calculated by

$$b_j^\theta = \frac{2j+1}{2r}\pi,\ j \in [0, r-1]. \quad (1)$$

Since these visual words are evenly distributed, the distance between any two adjacent visual words is the same. If we denote this distance as $dis_\theta$, then $dis_\theta = \frac{1}{r}\pi$.

### 2.3 The coding of features

The assignment of $p_i^\theta$ of a pixel to its corresponding bins in HOG is similar to the coding of $p_i^\theta$ into visual words $b_j^\theta$ using the local soft-assignment scheme [4]. The local soft-assignment scheme assumes that $p_i^\theta$ only contributes to its locally nearest visual words. In HOG, the number of these locally nearest visual words is set to 2. Due to the even distribution of visual words, the $u_{ij}$ can be simply calculated as

$$u_{ij} = \begin{cases} 1 - \frac{|p_i^\theta - b_j^\theta|}{dis_\theta} & b_j^\theta \in \mathbf{b}_{p_i^\theta}^N \\ 0 & b_j^\theta \notin \mathbf{b}_{p_i^\theta}^N \end{cases} \quad j \in [0, r-1] \quad (2)$$

where $\mathbf{b}_{p_i^\theta}^N$ is the set of locally nearest visual words of $p_i^\theta$.

### 2.4 The pooling of features

Max pooling is empirically asserted to have better performance than sum pooling and average pooling [10]. However, because in HOG a weight is required, sum pooling is applied for all $p_i^\theta$ to their corresponding $b_j^\theta$ with a weight $w_{ij}$, which is calculated as below.

$$w_{ij} = u_{ij} \times p_i^m \times s_i, \quad (3)$$

Where $s_i$ is calculated in terms of the location of the pixel where $p_i^\theta$ is extracted in the block (see section 3.2 for more details).

## 3 Introducing principles of BOF into HOG

In this section, we introduce principles proposed for BOF into the construction of descriptor of blocks in HOG. Those principles applied here are the preservation of locality, the data-driven vocabulary, and the preservation of spatial information.

### 3.1 The preservation of locality

The preservation of locality of features helps to discriminate features which are far away from each other in feature space, this can be achieved by coding and pooling the feature in the entire feature space.

In HOG, only the $p_i^\theta$ is used as the feature and gradients of close $p_i^\theta$ while of divergent $p_i^m$ are assigned to the same visual word in the coding phase. Whereas, gradients with different magnitudes usually carry different information which is useful for discrimination. For instance, gradients in the edge usually have large $p_i^m$ while gradients of small $p_i^m$ often come from background regions. In order to utilize this discriminative information, we extend the feature space to the entire gradient space by using $\mathbf{p}_i$ as the feature rather than merely $p_i^\theta$. This extension makes the coding and pooling of features more smoothly and locally[5].

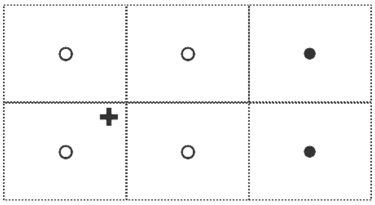

Figure 2. The gradient space is evenly divided into 6 regions (dashed rectangles) with $S_\theta = 3$ and $S_m = 2$. The visual words (circles) are in the center of each region. A feature (cross) is assigned to its 4 nearest visual words (hollow circles).

Following the operation in HOG, we manually define $\mathbf{B}$ to be evenly distributed in the gradient space. This amounts to dividing the gradient space into $r$ regions of identical area and the gradient in the center of each region is a visual word of $\mathbf{B}$ as depicted in Fig. 2. Let $S_\theta$ ($S_m$) denotes the number of regions that the orientation dimension (magnitude dimension) is divided into, $\mathbf{b}_{jk}$ ($j \in [0, S_\theta - 1], k \in [0, S_m - 1]$) a visual word. Then, $\mathbf{b}_{jk} = (b_{jk}^\theta, b_{jk}^m)$, and $b_{jk}^\theta$ and $b_{jk}^m$ are calculated as below,

$$b_{jk}^\theta = \frac{2j+1}{2S_\theta}\pi,\ b_{jk}^m = \frac{2k+1}{2S_m}L. \quad (4)$$

Also the coefficient $u_{ijk}$ is calculated as below,

$$u_{ijk} = \begin{cases} (1 - \frac{|p_i^\theta - b_{jk}^\theta|}{dis_\theta}) \times (1 - \frac{|p_i^m - b_{jk}^m|}{dis_m}) & \mathbf{b}_{jk} \in \mathbf{b}_{\mathbf{p}_i}^N \\ 0 & \mathbf{b}_{jk} \notin \mathbf{b}_{\mathbf{p}_i}^N \end{cases}, \quad (5)$$

where $dis_\theta$ and $dis_m$ are the lengths of the region in the orientation dimension and magnitude dimension respectively. Since $\mathbf{B}$ is evenly distributed, the $dis_\theta$ and $dis_m$ of all regions are identical. We limit the number of nearest visual words to 4 (see Fig. 2 for an example).

We term this extended descriptor of HOG BOG_GRAD, where GRAD means using entire gradient as feature.

### 3.2 The data-driven vocabulary

Manually defined vocabulary $\mathbf{B}$ is a generic vocabulary, which fails to capture the distribution of the local features. To incorporate this information, we use the clustering technique k-means, as what conventional BOF method does, to generate visual words for the vocabulary.

We cluster $p_i^\theta$ and $p_i^m$ individually to get $S_\theta$ cluster centers in the orientation dimension and $S_m$ cluster centers in the magnitude dimension, then combine them together to build $\mathbf{B}$. In this case, $r = S_\theta \times S_m$

and the coefficient $u_{ijk}$ is calculated as below,

$$\begin{cases} u_{ijk} = \dfrac{e^{-\beta|p_i^\theta - b_{jk}^\theta| - \beta|p_i^m - b_{jk}^m|}}{\sum_{jk} e^{-\beta|p_i^\theta - b_{jk}^\theta| - \beta|p_i^m - b_{jk}^m|}} & \mathbf{b}_{jk} \in \mathbf{b}_{\mathbf{p}_i}^N \\ 0 & \mathbf{b}_{jk} \notin \mathbf{b}_{\mathbf{p}_i}^N, \end{cases} \tag{6}$$

where $\beta$ is a parameter adjusting the softness of the exponential function.

We term this extended descriptor of BOG_GRAD BOG_GRAD_DV, where DV means data-driven vocabulary. Unlike BOG_GRAD, which uses the same vocabulary for all blocks, each block in BOG_GRAD_DV has its own vocabulary generated by k-means.

### 3.3 The preservation of spatial information

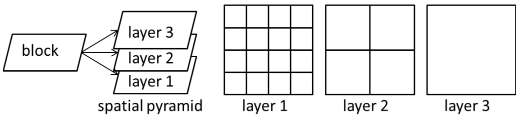

Figure 3. An example of spatial pyramid of $T = 3$.

The spatial information of the feature works as an important role in discriminating the class of this feature with other classes. The spatial pyramid matching technique introduced in [6] significantly mines this information and boost the performance of BOF method. In HOG, this spatial information is also exploited by dividing the block into 4 cells, extracting the descriptor from each cell and then concatenating these 4 cell descriptors to form the block descriptor. As well as this, a weight $s_i$ for $p_i^\theta$ (or $\mathbf{p}_i$ in BOG), which is calculated based on the distance between the pixel where $p_i^\theta$ is extracted and the center of the block, along with the distance between this pixel and the centers of 4 cells, is used to further record the location information of $p_i^\theta$.

Motivated by SPM, we extract the block descriptor from a spatial pyramid of the block. A spatial pyramid has $T$ layers and each layer is a copy of the block. The $t$th ($t \in [1, T]$) layer is divided into $4^{T-t}$ cells (Fig. 3 gives an example of spatial pyramid of $T = 3$.). Cell descriptors are concatenated into layer descriptors and layer descriptors are finally concatenated into block descriptors. The method used in HOG dividing block into 4 cells can be viewed as a simplified version of this method. This spatial pyramid extraction scheme is applied on the block, and thus is different from the multi-level HOG defined in [11], which applies the spatial pyramid extraction for the whole image. Moreover, the spatial weight $s_i$ is removed from multi-level HOG.

We call this extended descriptor of BOG_GRAD_DV using spatial pyramid extraction BOG.

## 4 Experiment

We performed three experiments in pedestrian detection task to evaluate the performance of BOG. Experiment 1 compares the performance of BOG_GRAD with that of HOG and examine how its performance is affected by different values of $S_m$. Experiment 2 compares the performance of BOG_GRAD_DV with that of BOG_GRAD. Experiment 3 compares the performance of BOG with that of BOG_GRAD_DV and examines the impact of different $T$ to the performance of BOG.

### 4.1 Dataset

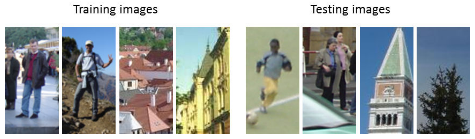

Figure 4. Some examples of image from the training and testing dataset.

We used the INRIA pedestrian dataset [7] to conduct the evaluation experiments. The positive and negative training dataset all contain 1000 images of size 64 × 128, which are generated by cropping positive and negative images of INRIA for training. The positive testing dataset contains 476 positive images with their reflections of size 64 × 128. We randomly selected 45300 patches of size 64 × 128 from negative images for testing in INRIA to form the negative testing dataset. Several examples of image from these datasets are displayed in Fig. 4.

### 4.2 Setting the parameters

For both HOG and BOG, the block size is set to 16×16. Unless otherwise noted, we use $S_a = 9$, $S_m = 3$, $T = 2$ and $\beta = 10$. Moreover, for both HOG and BOG, we used the L2-norm normalization rule $\mathbf{H} = \mathbf{H}/\sqrt{\|\mathbf{H}\|_2 + \epsilon}$, where $\epsilon$ is a regulating parameter and is fixed to $10^{-3}$ in this paper. For HOG, $\mathbf{H}$ is the block descriptor, for BOG, $\mathbf{H}$ is the layer descriptor. For purpose of capturing the discriminant information of the pedestrian, the clustering is performed merely on positive training dataset.

We trained the linear SVM classifier using Libsvm [12] for both HOG and BOG.

Experimental results are depicted in the detection error tradeoff (DET) curve, where the horizontal axis displays the logarithmic value of the false positives per window (FPPW) and the vertical axis displays the logarithmic value of the corresponding error rate $(\frac{false\ negative}{true\ negtive + false\ positive})$.

### 4.3 Experiment 1

In this experiment, we used $S_m = 2, 3, 4$ and compared the performances of them with HOG. The result is shown in Fig.5. The performance of BOG_GRAD is comparative to HOG when $S_m = 2$. With $S_m$ increasing to 3 and 4, the performance of BOG comes to excel HOG. Although the performance is continually raised when $S_m$ steps from 3 to 4, this promotion is not as significant as when $S_m$ rises from 2 to 3.

### 4.4 Experiment 2

In this experiment, we compared the performance of BOG_GRAD, which uses manually defined vocabulary, with BOG_GRAD_DV using data-driven vocabulary generated by k-means. The result is shown in

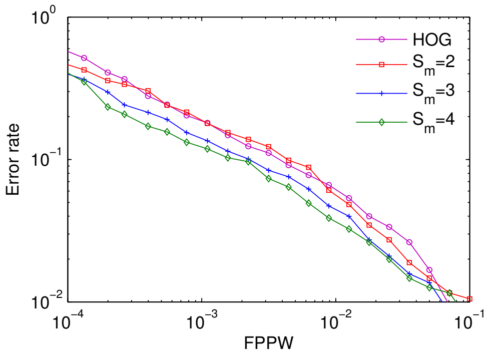

Figure 5. Result of experiment 1.

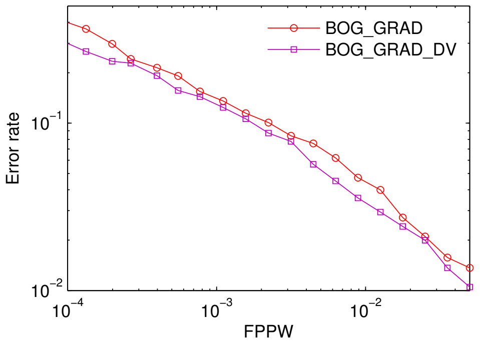

Figure 6. Result of experiment 2.

Fig.6. The performance of BOG_GRAD_DV continues to reduce the error rate in contrast to BOG_GRAD. This result consists with many experimental results of BOF method. Whereas, unlike typical BOF method, in which different classes explicitly have different cluster centers, the cluster centers of pedestrian may not diverge largely from those of the background, which makes the improvement of detection accuracy not quite significant in this experiment.

### 4.5 Experiment 3

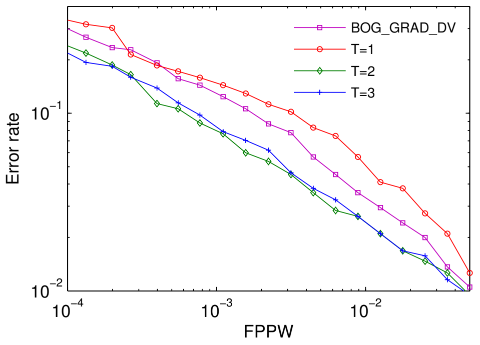

Figure 7. Result of experiment 3.

In this experiment, we used $T = 1, 2, 3$ and compared the performances of them with BOG_GRAD_DV. The result is shown in Fig.7. The usage of spatial pyramid extraction successfully exploits the spatial information of features and improve the performance of BOG_GRAD_DV. It seems like the adoption of $T = 2$ is sufficient to extract this spatial information. By increasing $T$ from 2 to 3, the performance conversely becomes worse. This deterioration is probably caused by the excessively fine division of the block into cells, which brings in redundant spatial information. Therefore, for block of certain size, it is necessary to conduct some experiments to select the suitable value of $T$.

## 5 Conclusion

We studied the relationship between HOG and BOF and showed that approach used to construct the block descriptor in HOG is similar to that used to build the coefficient descriptor in BOF method. Enlightened by this interpretation, we proposed a new descriptor, called BOG, which is a generalized version of HOG by embedding principles of BOF into HOG. The experimental results in pedestrian detection confirmed that BOG is more robust than HOG in capturing the discriminant information. Although this study is based on HOG, it is possible to be extended to the learning of other HOG-like local feature descriptors' relation with BOF and leverage advances of each other.

## References

[1] Y-Lan Boureau, et al.: "Learning mid-level features for recognition" *CVPR*, pp.2559–2566, 2010.

[2] Wright John, et al.: "Robust Face Recognition via Sparse Representation" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.31, nu.2, pp.210–227, 2009.

[3] Jinjun Wang, et al.: "Locality-constrained linear coding for image classification" *CVPR*, 2010.

[4] Lingqiao Liu, et al.: "In defense of soft-assignment coding" *ICCV*, pp.2486–2493, 2011.

[5] Y-Lan Boureau, et al.: "Ask the locals: multi-way local pooling for image recognition" *ICCV*, 2011.

[6] Svetlana Lazebnik, et al.: "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories" *CVPR*, vol.2, pp.2169–2178, 2006.

[7] Navneet Dalal, et al.: "Histograms of Oriented Gradients for Human Detection" *CVPR*, PP.886–893, 2005.

[8] David G Lowe, et al.: "Distinctive Image Features from Scale-Invariant Keypoints" *Int. J. Comput. Vision*, vol.60, nu.2, pp.91–110, 2004.

[9] Meng Yang, et al.: "Fisher Discrimination Dictionary Learning for sparse representation" *ICCV*, pp.543–550, 2011.

[10] Jianchao Yang, et al.: "Linear spatial pyramid matching using sparse coding for image classification" *CVPR*, vol.0, pp.1794–1801, 2009.

[11] Subhransu Maji, et al.: "Classification using intersection kernel support vector machines is efficient" *CVPR*, 2008.

[12] Chang, Chih-Chung, et al.: "LIBSVM: A library for support vector machines" *ACM Transactions on Intelligent Systems and Technology*, vol.2, pp.27:1–27:27, 2011.