A Discriminative Model for Object Representation and Detection via Sparse Features

Xi Song[1,2] Ping Luo[2] Liang Lin[2] Yunde Jia[1]
[1]*Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PRC*
[2]*Lotus Hill Research Institute, Wuhan 430074, PRC*
songxi@bit.edu.cn, {pluo.lhi, sailalone}@gmail.com, jiayunde@bit.edu.cn

***Abstract*—This paper proposes a discriminative model that represents an object category with a batch of boosted image patches, motivated by detecting and localizing objects with sparse features. Instead of designing features carefully and category-specifically as in previous work, we extract a massive number of local image patches from the positive object instances and quantize them as weak classifiers. Then we extend the Adaboost algorithm for learning the patch-based model integrating object appearance and structure information. With the learned model, a few features are activated to localize instances in the testing images. In the experiments, we apply the proposed method with several public datasets and achieve advancing performance.**

***Keywords*-discriminative model; object detection; sparse features;**

I. Introduction

In the research of object detection and recognition, there are two open and critical problems, particularly for the object categories with large intra-variance:

- How many exemplars need to be collected for modeling an object category?
- How many features need to be activated for localizing an object instance?

Addressing these problems, this paper presents a discriminative model that implicitly represents an object category with a batch of boosted image patches, as illustrated in Fig.1.

In the literature, there are two categories of learning-based methods related to our work. Firstly, the bag-of-feature models [8], [10] achieve great success on natural scene classification and object category recognition, which often construct the dictionary of visual words by detecting and clustering key features from positive samples, and learn the generative latent topic models, such as LDA [1], PLSA [7], through the EM-type algorithms. Cao et al. [2] extend to solve segmentation and categorization simultaneously by integrating the spatial coherency of words. However, with these models, it is difficult to encode the structure information of objects and scenes due to ignoring the location and scale of visual words. In addition, the number of visual words is often decided empirically without an analytical solution. Secondly, the boosting model [5] and its variations provide an alternative way to represent object categories by a set of selected features (weak classifiers), and lead

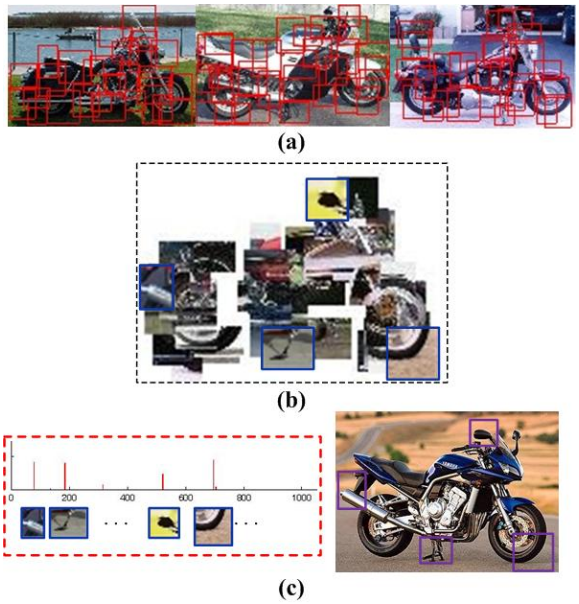

Figure 1: (a) Massive image patches are extracted from positive object samples to form a set of weak classifiers. (b) An object model is learned that implicitly represents an object category with a batch of boosted feature patches. This model naturally integrates object appearance and structure. (c) An object instance is localized in the testing image where few and sparse patches are activated.

the state-of-art performance on many public datasets. These methods, nevertheless, need to design feature carefully and category-specifically, which leads to the inconvenience for many applications.

In our method, we first extract a massive number of local patches from a set of positive object instances, (see Fig.1 a), and quantize them by a distinctive descriptor based on gradient orientation in transformed color space. Plus a small fluctuation denoted by δ, each quantized feature patch is then defined as a binary weak classifier that explains objects locally and compactly. In the training stage, we extend the Adaboost algorithm by introducing a default random guesser for calculating error rates. Given the strong classifier, a validation procedure is proposed to find the threshold of detection, which decides how many features should be activated for one shot testing. In the test stage, a simple and efficient strategy of sliding windows is adopted to search

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.754

objects with multiple scales in the testing images.

Compared to the previous work, the contribution of the proposed approach is as follows. (1) We automatically generate features (boosted image patches) to capture the variability of object categories rather than designing global features carefully and category-specifically. (2) Our models naturally integrate the appearance and structure information for objects. (3) The activated features for object localizing are few and sparse, which increase the detection robustness against the occlusion.

II. Features via proto image patches

Instead of designing global features, we first extract massive image patches from a set of positive object samples at a small range of size and aspect ratio. Then we quantize these patches to generate features that evolve into weak classifiers. We define the extracted patches, namely "proto patches", as

$$\mathbf{S}^{proto} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}, \tag{1}$$

where an instance of proto patch, $\mathbf{p}_i = (u_i, v_i, \Lambda_i)$, includes the width and height (u_i, v_i), and the image domain Λ_i, (both u_i and $v_i \in [15, 25]$ pixels in our experiments and $M \approx 10^4$). Note that our proto patches are location sensitive, thus the structure information is well captured.

A. HOG with transformed color space

Recently, the HOG (Histogram of Oriented Gradients) descriptor [3] demonstrates good performance in describing inhomogeneous texture properties. In this work, following [9], we compute the HOG descriptor in the transformed color space for each patches, which is proved more robust and invariant to illumination change. Our feature $\mathbf{Hist}(\mathbf{p})$ is generated from a proto patch by computing color transformed HOG. To counteract illumination change, we calculate histogram of oriented gradients in three color channels R, G, B respectively with pixel value distributions normalized as

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix}, \tag{2}$$

where μ and σ denote the mean and standard deviation of the distribution in each channel. Then the histogram of oriented gradients is pooled over the image domain in 3 transformed color channels, which are discretized into 24×3 bins. And the distance measure $\mathbf{d}(\mathbf{p}, \mathbf{q})$ between proto patch $\mathbf{p}$ and $\mathbf{q}$ is defined as

$$\mathbf{d}(\mathbf{p}, \mathbf{q}) = \mathcal{K}(\mathbf{Hist}(\mathbf{p}) \| \mathbf{Hist}(\mathbf{q})), \tag{3}$$

where $\mathcal{K}(\cdot)$ is the Kullback-Leibler divergence.

B. Weak classifiers

Given the massive quantized proto patches, we further define the weak classifiers by introducing a small fluctuation δ, as

$$\mathbf{h}_z(\mathbf{O}_k) = \begin{cases} 1, & \mathbf{d}(\mathbf{p}_i, \mathbf{q}_j) \leq \delta, \exists \mathbf{q}_j \text{ s.t. } X_i \approx X_j \\ -1, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{O}_k$ is an object instance with its label $l_k \in \{1, -1\}$, and $\mathbf{q}_j$ is a patch from $\mathbf{O}_k$ at locationX_j. X_i and X_j indicate the relative location of patch $\mathbf{q}$ and $\mathbf{p}$ with respect to their respective object center. $X_i \approx X_j$ denotes that $\mathbf{q}_j$ and $\mathbf{p}_i$ are almost at the same location with respect to the object center. Each weak classifier can be viewed as a ball in the quantized metric space with the proto patch viewed as its center. These balls can explain the images locally and independently.

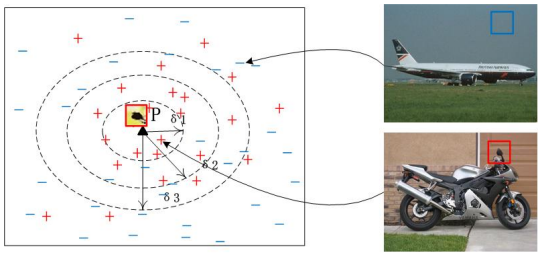

Figure 2: A subspace can grow based on a proto patch in the quantized metric space with a certain threshold δ given. Then samples that have similar structure at almost the same location fall into the subspace. Given different δ, we can obtain weak classifiers with different discrimination.

Now we discuss the computing of threshold δ for each weak classifier. First we define a big similarity matrix as

$$\mathbf{D} = [\mathbf{d}(\mathbf{p_i}, \mathbf{p_j})], \mathbf{d}(\mathbf{p_i}, \mathbf{p_i}) = \mathbf{0}, \tag{5}$$

where we can compute the neighborhood connections between patches. We propose an empirical method to compute δ efficiently as showed in Fig. 2. For each proto patch $\mathbf{p}$, we generate a small number $K(K = 3$ in our experiments) of δ according to the number of neighboring patches falling into the weak classifier. In the experiments, the discretized value of δ is computed by the weak classifier containing 0.1%, 0.2% and 0.4% amount of total proto patches, and thus we have $3 \times M$ weak classifiers in total.

III. Learning patch-based model

We propose a discriminative learning algorithm with the weak classifiers. A validation process is presented to calculate a threshold for the output strong classifier. As the detection result, the bounding box of each object is predicted based on our model.

Training process. In the training process, we use the AdaBoost algorithm [5] to select a subset from the total set of the various weak classifiers we obtain. AdaBoost is a classical algorithm to train strong classifier by boosting weak

classifiers and improve the performance efficiently. A strong classifier is a combination of a number of weak classifiers,

$$\mathbf{H}(x) = sign(\sum_{i=1}^{T} \lambda_i \ h_i(x)), \tag{6}$$

Default random guesser: Being different from conventional AdaBoost algorithm, the threshold of each weak classifier is not picked automatically. Unlike the discriminative boundary just divide the training samples without deep meaning, our manner to compute the threshold δ (see Sec.2.2) makes samples falling to a weak classifier look like the corresponding proto patch as much as possible. Thus only a small number of positive training samples fall into each classifier. While for most negative samples, most weak classifiers consider them to be negative. This would result in strong correlation between features, and the weighted error of each weak classifier would increase in the next iteration. To avoid strong correlation between all features, we introduce a default random guesser outside each weak classifier. The label of samples inside the weak classifier is still 1, but the guesser separates samples outside into two parts randomly and averagely, and assigns samples in one part label 1. In other words, we assume that all weak classifiers own the same error rate 0.5 outside. Therefore, the error rate of each weak classifier is calculated as

$$\begin{aligned} \mathbf{Err} &= \sum_{i=1}^{N} D_t(i) 1(y_i \neq h_t(x_i)) 1(h_t(x_i) = 1) \\ &+ \frac{1}{2} \sum_{i=1}^{N} D_t(i) 1(h_t(x_i) = -1), \end{aligned} \tag{7}$$

where $D_t(i)$ denotes the distribution over the training samples. The error rate of each weak classifier is composed of two parts, while error rate outside is fixed on 0.5 as a result of default random guess.

Actually, from the view of statistics, given thousands of weak classifiers, a certain sample is considered to be positive by half of random guessers. That is to say, the introducing of default random guesser would not affect the training process. Then in each round the learning algorithm selects the weak classifier with the lowest weighted error rate on the training samples. The output of the learning algorithm is a strong classifier which is a combination of the selected weak classifiers. The training process is summarized in Algorithm 1.

Validating process. The situation caused by default random guesser outside weak classifiers can not be predicted in the testing set. Since the effect of random guesser can be ignored, the weak classifier that selected for testing can be redefined as

$$\mathbf{h}_z(\mathbf{O}_k) = \begin{cases} 1, \mathbf{d}(\mathbf{p}_j, \mathbf{q}_i) \leq \delta, \exists \mathbf{q}_i \text{ s.t. } X_i \approx X_j \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{8}$$

Algorithm 1: Training process

Input: A set of proto image patches $\mathbf{S}^{proto}$; A training set $\mathbf{S}^{Train}$ contains N labeled training samples (x_i, y_i) with $y_i \in$ {-1,1} and $x_i \in \mathbf{S}^{Train}$; An initial distribution $D_1(x_i)$ over the samples.
Output: A strong classifier
1. Train weak classifiers based on $\mathbf{S}^{Proto}$;
   (1) Extract features from $\mathbf{S}^{Proto}$;
   (2) Computing a δ for each classifier;
2. Generate random guesser for each weak classifier;
3. Select features by weighted error and update data weights like Adaboost;
4. Output the strong classifier.

i.e. it is only the inside part of weak classifiers enabled in the testing procedure. Thus a validation process is necessary. We obtain a threshold of the strong classifier according to the scores our detector give on the validation set. The validation set consisting of a certain number of normalized positive samples and negative samples has no intersect with the training set. Given the validation scores, we adjust the threshold of the strong classifier from $+\infty$ to $-\infty$ to create a Receiver Operating Characteristic (ROC) curve, then choose a threshold according to the ROC curve.

Testing process. The goal of our object detection system is to predict the bounding box of objects in the testing images. In the testing process, the detection window is scanned across the testing image at multiple scales densely. And the image region within the window is normalized to a certain scale according to the training samples. Each window that achieves higher score than the validation threshold is used to predict a bounding box of an object. After the scanning process, we have a set of detection results for a certain image. Then we adopt the Non-Maximum Suppression method proposed by [4] to prune those results, that have too much overlap with others but lower score.

IV. Experiments

We evaluate our method on three object categories from two challenging datasets: motorbike and aeroplane from the Caltech [6] dataset both have about 800 images; pedestrian from the INRIA [3] dataset contains 1805 images. For each category, we randomly divide the image set into 4 subsets: the first for feature generation, the second for training, the third for validation and the last for testing. The ground truth in learning stage is roughly annotated. For proto patch extraction, we resize the objects into 150×150 pixels. The patch size is in a range, $(15, 15) \sim (25, 25)$. We consider that the flatness patches has less information, therefore they are simply removed. For each category, the number of proto patches is more than 2×10^4. In the testing stage, we adopt

Methods	Motorbike	Aeroplane	Pedestrian
SVM+HOG	67.25%	75.39%	41.81%
Our method	**76.41%**	**83.81%**	**58.63%**

Table I: Average precision compared to SVM+HOG.

Figure 3: A few results of object localization. For each category, the first frame shows a model comprised of a batch of proto patches. The next two frames demonstrate an instance detected and the activated proto patches of the model.

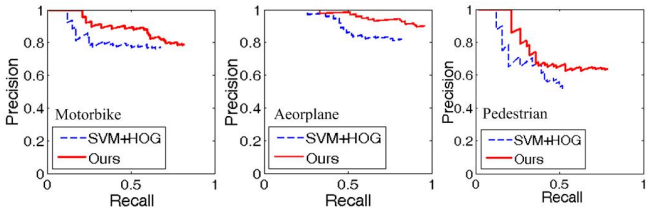

Figure 4: PR curve obtained by our method and SVM+HoG on Caltech dataset and INRIA dataset.

windows in a set of scales with a factor of 1.25 apart.

A few example images and results are summarized in Fig.3. We compare the performance with the SVM-classifier with the HOG feature [11]. Fig.4 shows the PR (precision recall) curves obtained by the two methods. And the average precision (AP) of the two methods on the three categories are summarized in Table I.

V. Conclusion

We present a discriminative model to represent an object category by quantizing a massive number of proto image patches. We extend the Adaboost algorithm by introducing a default random guesser for calculating error rates. Then a validation process is proposed to find the threshold of detection. In the experiments, we show the advancing performance of our method on several typical categories. We intend to improve our work by combining hierarchical models in the future work.

Acknowledgment

This work was partially supported by the Chinese High-Tech Program under Grant No.2008AA01Z126 and No.2009AA01Z323, and the Natural Science Foundation of China under Grant No.60875005.

References

[1] D. M. Blei, A. Y. Ng and M. I. Jordan, *Latent Dirichlet allocation, Journal of Machine Learning Research*, 3, 993-1022 ,2003.

[2] L. L. Cao and L. Fei-Fei, *Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes* IEEE Proc. Int'l Conf. Computer Vision, 2007.

[3] N. Dalal and B. Triggs, *Histograms of Oriented Gradients for Human Detection*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[4] P. F. Felzenszwalb, R. B. Girshick, D. M. Allester and D. Ramanan, *Object Detection with Discriminatively Trained Part Based Models*, TPAMI, 2009.

[5] J. Friedman, T. Hastie and R. Tibshirani, *Additive logistic regression: a statistical view of boosting*, Dept. of Stat., Stanford Univ. Tech. Rep. 1998.

[6] L. Fei-Fei, R. Fergus, and P. Perona, *Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories*, In CVPR Workshop on Generative-Model Based Vision, 2004.

[7] T. Hofmann, *Probabilistic Latent Semantic Indexing*, Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), 1999.

[8] L. Fei-Fei, P. Perona, *A Bayesian Hierarchical Model for Learning Natural Scene Categories*, CVPR, 2005.

[9] K. Sande, T. Gevers and G. M. Snoek, *Evaluation of Color Descriptors for Object and Scene Recognition*, CVPR, 2008.

[10] A. Torralba, K. P. Murphy and W. T. Freeman, *Sharing features: efficient boosting procedures for multiclass object detection*, CVPR. pp 762-769, 2004.

[11] J. Zhang, et al., *Local features and kernels for classification of texture and object categories: A comprehensive study*, IJCV, 73(2): 213-238, 2007.