Improving object detection with boosted histograms

Ivan Laptev*

INRIA Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France

ABSTRACT

We address the problem of visual object class recognition and localization in natural images. Building upon recent progress in the field we show how histogram-based image descriptors can be combined with a boosting classifier to provide a state of the art object detector. Among the improvements we introduce a weak learner for multi-valued histogram features and show how to overcome problems of limited training sets. We also analyze different choices of image features and address computational aspects of the method. Validation of the method on recent benchmarks for object recognition shows its superior performance. In particular, using a single set of parameters our approach outperforms all the methods reported in VOC05 Challenge for seven out of eight detection tasks and four object classes while providing close to real-time performance.

Keywords:
Object recognition
Machine learning
Histogram image features

1. Introduction

Among the vast variety of existing approaches to object recognition there is a remarkable success of methods using histogram-based image descriptors. An influential work by Swain and Ballard [24,18,2] proposed color histograms as a simple and efficient image descriptor for object recognition. The idea was further developed by Schiele and Crowley [23] who recognised objects using histograms of local filter responses. Histograms of Textons were proposed by Leung and Malik [14] as well as by Varma and Zisserman [27] for texture recognition. Schneiderman and Kanade [24] computed histograms of wavelet coefficients over localized object parts and were among the first to address object class detection in images of natural scenes. In a similar spirit the well-known SIFT descriptors [18] and Shape Context [1] as well as more recent HOG descriptor [2] and Spatial Pyramid representations [13] make an effective use of position-dependent histograms to describe local and global image content.

Histograms represent distributions of spatially unordered image measurements in a region and provide relative invariance to several variations of object appearance. The invariance and the descriptive power of histograms, however, crucially depend on (i) the type of local image measurements and (ii) the image regions used to accumulate histograms. Regarding the type of measurements, different alternatives have been proposed and investigated that may have better performance depending on the task.

As a general purpose image descriptor, the choice of Histograms of Oriented Gradients (HOG) is well supported by successful applications of SIFT descriptor [18,21] and other related methods [2].

Besides the question what to measure, the question where to measure obviously has a large impact on recognition performance. Global histograms [26,23] have recently achieved impressive performance for scene categorization [13,31]. Object recognition and localization, however, is currently better addressed by local methods [24,18,2] computing histograms over local image regions. As illustrated in Fig. 1, different regions of an object may have different descriptive power and, hence, different impact on the learning and recognition. In the previous work histogram regions were often selected either a-priori using fixed grids [24,2] or by applying region detectors of different kinds [18,3,19]. None of these two alternatives, however, guarantees an optimal choice of histogram regions for subsequent recognition. An arguably more attractive approach proposed by Levi and Weiss [15] and confirmed in [12,32] consists of learning class-specific histogram regions from the training data. We follow this approach and note its conceptual similarity to other methods making attempt to discover discriminative object parts for visual recognition [5]. In this work, similar to [15], we select the position and the shape of histogram features to minimize the training error for a given recognition task. During training, we consider an exhaustive set of rectangular regions in the normalized object window and compute histogram descriptors for each of them efficiently using integral histograms [22]. We then apply AdaBoost [8,29] to select histogram features and to learn an object classifier. As a part of our contribution to object learning, we adapt the boosting framework to vector-valued histogram features and design a weak learner based on Weighted Fischer Linear Discriminant (WFLD). We in addition deploy position-dependent histogram features and artificially enlarge the size of the training set by adding spatial noise to the annotation. These extensions demonstrate a substantial improvement with respect to [15].
To validate the proposed method, we test it on the task of object detection in natural images and evaluate the performance on PASCAL Visual Object Category datasets VOC 2005 and VOC 2006 [7,6]. Using a single set of parameters we demonstrate our approach to outperform all methods reported in the competition [7] for seven out of eight detection tasks and four object classes. Among the advantages of the method we emphasise (i) its ability to learn from a small number of samples, (ii) stable performance for different object classes, and (iii) close to real-time performance.

We further investigate the framework by comparing performance of alternative histogram features and feature selection mechanisms. Evaluation on several object classes confirms the high performance of HOG descriptors, however, the best performance is demonstrated by the combination of HOG features with other histogram descriptors in terms of second-order image derivatives and color. Given the popularity of interest point features in recognition problems, we also compare regions selected by our method with Harris-Affine regions [20]. Notably, we find Harris-Affine regions to perform no better than random regions in our framework tested on three different object classes. We finally investigate computational aspects of the method and evaluate its precision-speed tradeoff.

The rest of the paper is organised as follows. In Section 2 we recall AdaBoost algorithm and develop a weak learner for vector-valued features. Section 3 defines histogram features and integrates them with the boosting framework. In Section 4 we evaluate and compare the method on the task of object detection. Sections 5 and 6 investigate alternative image features and computational aspects of the method, respectively. Section 7 concludes the paper.

2. AdaBoost learning

AdaBoost [8] is a popular machine learning method combining properties of an efficient classifier and feature selection. The discrete version of AdaBoost defines a strong binary classifier $H$

$$H(z) = \text{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(z)\right)$$

using a weighted combination of $T$ weak learners $h_t$ with weights $\alpha_t$. At each new round $t$, AdaBoost selects a new hypothesis $h_t$ that best classifies training samples with high classification error in the previous rounds. Each weak learner

$$h_t(z) = \begin{cases} 
1 & \text{if } g_t(f(z)) > \text{threshold} \\
-1 & \text{otherwise}
\end{cases}$$

may explore any feature $f$ of the data $z$. In the context of visual object recognition it is attractive to define $f$ in terms of local image properties over image regions $r$ and then use AdaBoost for selecting features maximizing the classification performance. This idea was first explored by Viola and Jones [29] who used AdaBoost to train an efficient face detector by selecting a discriminative set of local Haar features. Here similar to [15], we will define $f$ in terms of histograms computed for rectangular image regions on the object.

2.1. Weak learner

The performance of AdaBoost crucially depends on the choice of weak learners $h$. While effective weak learners will increase the performance of the final classifier $H$, the potentially large number of features $f$ prohibits the use of complex classifiers such as Support Vector Machines or Neural Networks. For one-dimensional features $f \in \mathbb{R}$ such as Haar features in [29], an efficient classifier for $n$ training samples can be found by selecting an optimal decision threshold in $O(n \log n)$ time. For vector-valued features $f \in \mathbb{R}^m$ such as histograms, however, finding an optimal linear discriminant would require unreasonably long $O\left(\binom{n}{m}\right)$ time.

One approach to deal with multi-dimensional features used in [15] is to project $f$ onto a pre-defined set of one-dimensional manifolds using a fixed set of functions $g_i : \mathbb{R}^m \rightarrow \mathbb{R}$. A weak learner can then be constructed for each combination of basis functions $g_i$ and features $f$. Although efficient, such an approach can be suboptimal if a chosen set of functions $g_i$ is not well suited for a given classification problem. As an example of inefficient AdaBoost classifier consider the problem of separating two diagonal distributions of points in $\mathbb{R}^2$ illustrated in Fig. 2(left). Using axis-parallel linear basis functions $g_1(f) = (1 0)f$ and $g_2(f) = (0 1)f$, the resulting AdaBoost classifier has poor generalization and requires $T \approx 50$ weak hypotheses for separating $n = 200$ training samples.

An alternative and efficient choice for a multi-dimensional classifier is Fisher Linear Discriminant (FLD) [4]. FLD has been used as a weak learner in the context of AdaBoost in [30]. FLD guarantees optimal classification of normally distributed samples of two classes using a linear projection function

$$g = w^T f$$

defined by the class means $\mu^{(1)}$, $\mu^{(2)}$ and the class covariance matrices $S^{(1)}$, $S^{(2)}$. Illustration of FLD classification in Fig. 2(right) clearly indicates its advantage in this example compared to the classifier in Fig. 2(left). A particular advantage of using FLD as a weak learner is the possibility of re-formulating FLD to minimize a weighted classification error as required by AdaBoost. Given the weights $d_i$ corresponding to samples $z_i$, the Weighted Fisher Linear Discriminant (WF LD) can be obtained using a function $g$ in (2) with the means $\mu$ and covariance matrices $S$ substituted by the weighted means $\mu_d$ and the weighted covariance matrices $S_d$ defined as

$$\mu_d = \frac{1}{n} \sum_{i=1}^{n} d_i f(z_i), S_d = \frac{1}{(n-1) \sum_{i=1}^{n} d_i} \sum_{i=1}^{n} d_i^2 (f(z_i) - \mu_d) (f(z_i) - \mu_d)^T$$

Fig. 1. Rectangles on the left and right image are examples of possible regions for histogram features. Stable appearance in A, B and C on both images makes corresponding features to be good candidates for a motorbike classifier. On the contrary, regions D are unlikely to contribute for the classification due to the large variation in appearance.
Using WFLD as an AdaBoost weak learner eliminates the need of re-sampling training data required by classifiers that do not make use of sample weights.

In practice, the distribution of image features \( f(x_i) \) will mostly be non-Gaussian and multi-modal. Given a large set of features \( f \), however, we can assume that the distribution of samples at least for some features will be close to Gaussians yielding the good performance of the resulting WFLD classifier. Experimental validation of this assumption and the advantage of WFLD will be demonstrated in Section 4 on real classification problems. In this work we use WFLD to find one-dimensional projections of histogram features according to (2) and then determine an optimal classification threshold as in [29].

### 3. Image features

During training we assume a rough alignment of object samples within a rectangular window (see Fig. 5). Under this assumption we rely on the correspondence of object parts and learn the appearance of parts from corresponding image regions. To avoid a heuristic selection of such regions, we initially consider an exhaustive set of rectangular sub-windows \( r \) on the object for AdaBoost learning as illustrated in Fig. 3(left).

#### 3.1. Histogram features

We represent each feature by a histogram of local image measurements within a region \( r \). Following previous work [18,2], we initially adopt Histograms of Oriented Gradients (HOG) features and consider histograms of alternative image measurements such as color and second-order image derivatives later in Section 5. To construct HOG features, we compute orientation \( \gamma \) of local image gradient at each point \((x,y)\in r\):

\[
\gamma(x,y) = \arctan \frac{L_x(x,y)}{L_y(x,y)} = I \frac{\partial}{\partial z} \left( \frac{1}{2\pi\sigma^2} e^{-((x^2+y^2)/2\sigma^2)} \right) |_{z=xy}
\]

using Gaussian derivatives \( L_x, L_y \) [16] of image \( I \) computed for scale parameter \( \sigma \). We discretize \( \gamma \) into \( m=4 \) equal orientation bins and increment histograms by the values of the gradient magnitude \( ||(L_x,L_y)||_2 \). The histograms are normalized to the unit norm.

To preserve rough location of image measurements within a region, we sub-divide regions into parts as illustrated in Fig. 3(right) and compute histograms separately for each part. Four types of image features \( f_k(I) \) with spatial grids \( k = (1x1,1x2,2x1,2x2) \) are then computed for each region \( r \) by concatenating part-histograms into feature vectors of dimensions \( m,2m,2m \) and \( 4m \), respectively. We use integral histograms [15,22] for efficient computation of histogram features.

#### 3.2. Feature selection

At the training we compute features \( f_k(I) \) for normalized training images and apply AdaBoost to select a set of features \( f_k \) and the corresponding weak classifiers \( h(f_k) \) optimizing classification performance. A few features selected for the motorbike class at first rounds of AdaBoost are shown in Fig. 4(left). Superposition of all selected features in Fig. 4(middle) illustrates the emphasis of the final strong classifier on image regions with prominent appearance such as the regions of the front wheel and of the seat. Fig. 4(right) illustrates the high number of selected features with \( 2 \times 2 \) spatial grids and indicates the preference of position-dependent histograms for classification.

### 4. Evaluation

We evaluate the described classifier on the problem of object detection in natural images. To train the classifier for a particular object class, we use positive training set with scale and position-normalized images of objects in similar views. We obtain new negative training samples for each training cascade by collecting false positive detections from training images. For the detection we use the standard window scanning technique and apply the classifier to the large number of image sub-windows with densely sampled...
positions and sizes. To suppress multiple detections we cluster detected image windows with respect to their positions and sizes in the image and use the size of resulting clusters as a confidence measure of detections.

To overcome the frequently limited number of positive training samples, we found it particularly useful to artificially enlarge the positive training set as follows. Given annotation rectangles for objects in training images, we generate similar rectangles for each annotation by adding noise to the position and the size of original rectangles. We use noisy annotation to generate new positive image samples and in this way enlarge the positive training set. The procedure is illustrated in Fig. 5.


Our method differs from the one proposed by Levi and Weiss [15] in three main respects: (i) we introduce WFLD weak learner for vector-valued features, (ii) we use position-dependent histogram features and (iii) we artificially enlarge the positive training set. To evaluate these extensions we compare our method with

![Fig. 4. Selected features for the motorbike class. (Left) Regions of the first three selected and most discriminative features; (middle) all selected features superimposed using transparent color. Bright areas correspond to the high feature density; (right) relative frequency of features with different spatial grids.](image1)

![Fig. 5. (Left) Positive training samples are obtained using crop-and-resize procedure applied to training images with rectangular objects annotations. (Right) The same procedure is applied to training images using the large number of automatically generated noisy annotations. Note how annotation noise adds simulated affine deformations to the novel training samples.](image2)

![Fig. 6. Detection results for motorbikes on VOC 2005 validation set in terms of precision-recall curves and average precision (AP) values. (Left) Evaluation of improvements introduced in this paper compared to “1-bin” method [15]; (right) evaluation of artificially enlarged training sets.](image3)
Fig. 7. PR-curves for eight object detection tasks in PASCAL VOC 2005 Challenge. The proposed method (boosted histograms) is compared to the best performing methods reported in [7]. (Better viewed in color.)
Fig. 8 shows examples of detection results for motorbikes and people. In Fig. 8(top) the gradual decrease of detection confidence is consistent with the increasing complexity of detected motorbikes. The frequent presence of bicycles within false positives is also intuitive. The detection performance for people is lower in Fig. 8(bottom), however, many high confident false detections (red rectangles) overlap with people in test images. These detections are classified as false positives due to the insufficient overlap with ground truth (green rectangles) or due to the missing annotation.

Table 1

<table>
<thead>
<tr>
<th>Method</th>
<th>Motorbikes</th>
<th>Bicycles</th>
<th>People</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosted Histograms</td>
<td>0.896</td>
<td>0.370</td>
<td>0.250</td>
<td>0.663</td>
</tr>
<tr>
<td>TU-Darmstadt</td>
<td>0.886</td>
<td>–</td>
<td>–</td>
<td>0.489</td>
</tr>
<tr>
<td>Edinburgh</td>
<td>0.453</td>
<td>0.119</td>
<td>0.002</td>
<td>0.000</td>
</tr>
<tr>
<td>INRIA-Dalal</td>
<td>0.490</td>
<td>–</td>
<td>0.013</td>
<td>0.613</td>
</tr>
</tbody>
</table>

Bold values correspond to methods with best performance for particular object classes.

Table 2

<table>
<thead>
<tr>
<th>Method</th>
<th>Motorbikes</th>
<th>Bicycles</th>
<th>People</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosted Histograms</td>
<td>0.400</td>
<td>0.279</td>
<td>0.230</td>
<td>0.267</td>
</tr>
<tr>
<td>TU-Darmstadt</td>
<td>0.341</td>
<td>–</td>
<td>–</td>
<td>0.181</td>
</tr>
<tr>
<td>Edinburgh</td>
<td>0.116</td>
<td>0.113</td>
<td>0.000</td>
<td>0.028</td>
</tr>
<tr>
<td>INRIA-Dalal</td>
<td>0.124</td>
<td>–</td>
<td>0.021</td>
<td>0.304</td>
</tr>
</tbody>
</table>

5. Alternative image features

In this section we consider alternative histogram features and feature selection mechanisms and evaluate our method augment with such extensions on object detection tasks.

5.1. Histograms of color and second-order image derivatives

We investigate whether the histograms of alternative image properties can provide better or complementary performance with respect to HOG features. For this purpose in addition to HOGs we introduce three histogram descriptors defined by local image measurements in terms of (i) multi-scale Laplacian responses, (ii) second-order jet responses and (iii) color as illustrated in Fig. 10(left). The choice of Laplacian features is motivated by their rotation invariance and scale selection property [17,18]. Second-order jets [11] capture local second-order differential image structure while color is discriminative e.g. for certain animal classes. To construct histograms, we maximize responses over associated filters at every image point and increment corresponding histogram bins. The training and the detection then follows the same procedure as described for HOG features in previous sections.

Relative performance of different histogram features in Fig. 10(right) illustrates superior performance of HOG compared to other gray-scale descriptors. Color histograms outperform HOG for horses but result in poor training convergence for other two object classes. The best performance for all tested classes is achieved by the combination of all features. The combination was achieved by clustering multiple responses of alternative detectors trained separately for each type of features.

5.2. Alternative feature selection

Interest point features have been a popular choice of local image descriptors in many recognition methods [9,19,25,31]. We investigate if these descriptors bear similarity with histogram features selected by our method. For this purpose we choose Harris-Affine features [20] as an example of a popular region detector illustrated in Fig. 11(left). We then compare the values of Harris function [10] computed for Harris-Affine features, boosted regions and random regions on motorbike images. Distributions of Harris values for these three types of regions are illustrated in Fig. 11(right). As expected, the responses of Harris function are higher for Harris-Affine features compared to random regions. Notably, boosted regions show low responses for Harris function and, hence, bear low similarity to Harris-Affine features.

We next investigate whether the boosted histogram detector can be improved by using Harris interest regions for training. For this purpose we pre-select fractions of features using (a) random selection of regions and (b) selection of interest regions maximizing the Harris function. We train classifiers for three VOC 2005 object classes using different fractions of pre-selected features and different selection methods and evaluate the detection performance in Fig. 12(top). Notably, the performance of random features is similar or better compared to Harris features. At the same time the complexity of classifiers trained on Harris regions is higher compared to random regions according to Fig. 12(bottom).
This indicates that image features selected by the popular Harris function may not always be the best choice in a recognition system.

In Fig. 12 we also observe the very stable performance and complexity of detectors trained on 10% randomly selected regions only. This implies the opportunity to speed up the training procedure without penalizing the performance and the complexity of the detection.

6. Computational aspects

In their face detection method Viola and Jones [29] introduced integral images for the fast computation of rectangular gray-level features. This idea was further developed to integral histograms [15,22] to enable fast computation of histograms in rectangular image regions of arbitrary positions and sizes. A major difference to the original approach in [29] arises, however, when computing...
histograms of filter responses for multi-scale tasks such as for object detection at multiple image resolutions.

Filter responses such as the responses of Gaussian derivatives are known to change over image scales [16]. Hence, to enable unbiased computation of histograms at different scales, either the size of filter kernels or the image resolution has to be adapted to the scale parameter. This, however, implies additional computational cost due to a separate filtering step and the re-computation of integral histograms at each scale level.

Given the high correlation of filter responses at adjacent image scales, computation of integral histograms for a limited set of sparse scale levels is likely to imply a speed up at the cost of a limited decrease of performance. To investigate this issue in the context of our detection algorithm we introduce the following parameters. We denote the number of scale levels in octave by \( a \) implying a scale factor of \( 2^{1/a} \) between adjacent scale levels. We recompute integral histograms at each \( b \)th scale level \( c = n b, n \in \mathbb{Z} \) only. At other scale levels \( c_i \) we accommodate for scale changes by resizing rectangular

---

**Table 3**

<table>
<thead>
<tr>
<th>Method</th>
<th>Bicycle</th>
<th>Cow</th>
<th>Horse</th>
<th>Motorbike</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>INRIA Douze</td>
<td>0.414</td>
<td>0.212</td>
<td>–</td>
<td>0.390</td>
<td>0.164</td>
</tr>
<tr>
<td>INRIA Laptev</td>
<td>0.440</td>
<td>0.224</td>
<td>0.140</td>
<td>0.318</td>
<td>0.114</td>
</tr>
<tr>
<td>TKK</td>
<td>0.303</td>
<td>0.252</td>
<td>0.137</td>
<td>0.265</td>
<td>0.039</td>
</tr>
</tbody>
</table>
features of the object classifier similar to \[29\] while deriving histogram features from an integral histogram at scale level \(c = b_{ci}/b_{c}\).

To study the tradeoff between the speed and the accuracy of our detection method we perform a set of experiments using different values of parameters \(a\) and \(b\) while measuring average precision of detection on the VOC05 motorbike validation dataset. As illustrated in Fig. 13(left) the precision of detection remains stable for \(a = 5, 10\) and \(b = 1, 5, 10\) while the detection speed increases more than twice (see Fig. 13, right). Choosing two scale levels in octave \((a = 2)\) while recomputing integral histograms at each 5th scale level \((b = 5)\) seems to give a near optimal precision-speed tradeoff on this dataset. We have observed similar behaviour for detectors trained on other object classes. Our current implementation of object detection runs at about 10fps frame rate on 320 \(\times\) 240 images on a modest PC.

The implementation source code is available for download.2

7. Conclusion

We presented a method for object detection that combines AdaBoost learning with local histogram features. While being conceptually similar to \([15]\) our method provides a number of extensions that significantly improve the results of object detection. We evaluated the method on recent benchmarks for object recognition \([7,6]\) and demonstrated its competitive performance compared to the state-of-the-art. We also addressed computational aspects of the method by analyzing precision-seed tradeoff of detection.

Acknowledgements

The author would like to thank Patrick Pérez and Patrick Bouthemy for their helpful comments. Mark Everingham, Mario Fritz and Navneet Dalal were extremely helpful providing details and results of the VOC 2005 Challenge.

2 Family http://www.irisa.fr/vista/Equipe/People/Laptev/objectdetection.html.
References


[28] V. Viitaniemi, J. Laaksonen, Techniques for still image scene classification and object detection, in: 16th Int. Conf. on Artificial Neural Networks (ICANN), 2006.


