

# Shared Random Ferns for Efficient Detection of Multiple Categories

Michael Villamizar<sup>1,2</sup>, Francesc Moreno-Noguer<sup>1</sup>, Juan Andrade-Cetto<sup>1</sup> and Alberto Sanfeliu<sup>1,2</sup>

<sup>1</sup>*Institut de Robòtica i Informàtica Industrial, CSIC-UPC*

<sup>2</sup>*Department of Automatic Control, UPC*

{mvillami, fmoreno, cetto, sanfeliu}@iri.upc.edu

## Abstract

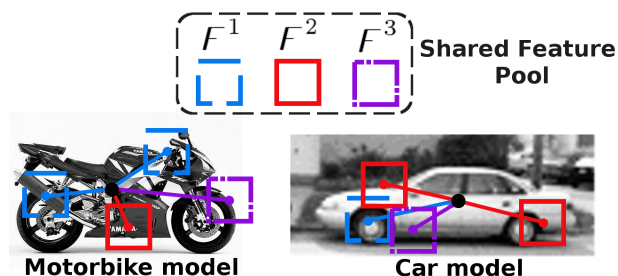
We propose a new algorithm for detecting multiple object categories that exploits the fact that different categories may share common features but with different geometric distributions. This yields an efficient detector which, in contrast to existing approaches, considerably reduces the computation cost at runtime, where the feature computation step is traditionally the most expensive. More specifically, at the learning stage we compute common features by applying the same Random Ferns over the Histograms of Oriented Gradients on the training images. We then apply a boosting step to build discriminative weak classifiers, and learn the specific geometric distribution of the Random Ferns for each class. At runtime, only a few Random Ferns have to be densely computed over each input image, and their geometric distribution allows performing the detection.

The proposed method has been validated in public datasets achieving competitive detection results, which are comparable with state-of-the-art methods that use specific features per class.

## 1. Introduction

The problem of object category detection from images has been shown to be a challenging one, because of the large intra-class variations that an object may suffer, due to changes in pose, illumination and cluttered backgrounds. Many recent methods have shown remarkable success for single class detection, by using intensive machine learning techniques such as Boosting [7, 14], or Support Vector Machines [2, 3, 6, 10, 12], to learn the different appearances of an object. As shown in [17], among the previous methods, those based on

This work has been partially funded by the Spanish Ministry of Science and Innovation under projects UbROB DPI2007-61452, PAU DPI2008-06022, and MIPRCV Consolider Ingenio 2010 CSD2007-00018; and by the EU project GARNICS FP7-247947. The first author is funded by the Technical University of Catalonia (UPC).

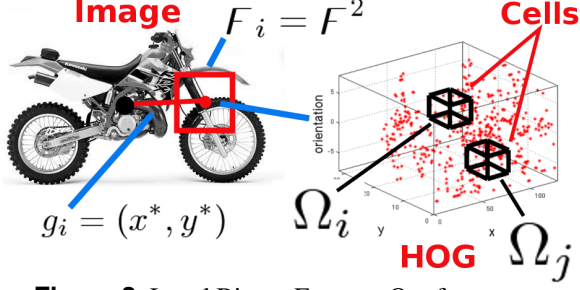


**Figure 1.** Shared Feature Pool. Car and motorbike categories share the features computed using Random Ferns.

Histograms of Oriented Gradients (HOG) have demonstrated to outperform classical methods based on Haar-like intensity features.

In order to go a step further and deal with multiple categories, several classifiers or detectors have to be learnt offline and tested at runtime. The simplest approach consists in learning each category-specific classifier independently from the rest. While this procedure allows focusing on the most discriminative features for each category, it has a high computational cost at runtime, because the total number of features increases with the number of object categories. This is specially critical, when features are computed by using a sliding window under different scales over the whole image.

Recent approaches have attempted to reduce this computational burden inherent in the multiclass object detection problem by splitting the detection process in two steps: initially the object class is estimated by either using joint class classifiers [16] or a rough class estimator [12], and subsequently the object is accurately detected through category-specific classifiers. Nevertheless, in both situations the initial estimation is only reliable when object categories may be represented by predefined-regions with the same size, and when dealing with classes that may have very distinct aspect-ratios this procedure is prone to introduce many false positives and negatives in the initial estimation.



**Figure 2.** Local Binary Feature. Our features are computed from binary comparisons between different bins of the HoG.

In order to address this situation we proceed as the former methods mentioned above, independently learning a robust classifier for each category. However, in order to make their computation as efficient as possible, we propose to build a pool of features that is shared by all the categories. For instance, in the example shown in Fig.1, three features are shared by the “car” and “motorbike” classes. Then, at runtime, only these three features have to be evaluated over the input image, and the decision to classify such an image to belong to one class or the other, depends on the response of category classifiers built from the common features. Note that the process to compute the features only needs to be done once, and it is independent from the number of categories to detect.

More specifically, in order to build the shared pool of features, we apply the very same Random Ferns (RF’s) [11] on the HOG’s of a set of training images from multiple classes. Given these features, we then use a boosting step to learn discriminative object classifiers. The result of the boosting step is a specific combination of Random Ferns for each category, that although sharing the same Ferns, it has a geometric distribution that is particular for each class. As will be shown in the Results section, our detector yields similar recognition and detection results as state-of-the-art approaches when applied to each of the individual classes.

The rest of paper is organized as follows. Section 2 explains the procedure to compute local binary features over the HOG. Section 3 describes how discriminative RFs are computed. Category classifier computation and experiments are described in Sections 4 and 5, respectively.

## 2 Local Binary Features

A Local Binary Feature (LBF) maps the image sample  $x$  to a boolean space in the form,

$$f : x \rightarrow \{0, 1\} , \quad x \in X , \quad (1)$$

by simple comparison between a pair of image values (e.g pixel intensities). Traditionally, LBFs are com-

puted in the image intensity domain yielding successful detection results for specific objects [11, 15]. We extend the same idea and propose to compute LBFs in the HOG domain because in recent years HOG-based methods have demonstrated remarkable results for object categorization showing robustness to illumination and object appearance changes. Therefore, for our purposes, an LBF is defined as a signed comparison between two HOG cells,

$$f(x) = \begin{cases} 1 & x_{\Omega_i} > x_{\Omega_j} \\ 0 & x_{\Omega_i} \leq x_{\Omega_j} \end{cases} , \quad \Omega \in \mathbb{R}^3 , \quad (2)$$

where  $\Omega_i$  and  $\Omega_j$  are the feature component locations defined by spatial and orientation coordinates  $(u, v, \theta)$ . Figure 2 shows one LBF instance in a HOG.

## 3 Discriminative Random Ferns

In order to compute object features, we use the Random Ferns proposed in [11] for keypoint classification. However, and in contrast to this original formulation of the Random Ferns, we write the Ferns expression in terms of likelihood ratios between classes. This allows us to seek for the feature combinations that maximize this ratio, by means of a boosting algorithm.

Our goal is to model the posterior object class probability given a set of  $n$  features (LBF). This can be expressed by means of the Bayes rule as

$$P(C_j | f_1, f_2, \dots, f_n) = \frac{P(f_1, f_2, \dots, f_n | C_j) P(C_j)}{P(f_1, f_2, \dots, f_n)} , \quad (3)$$

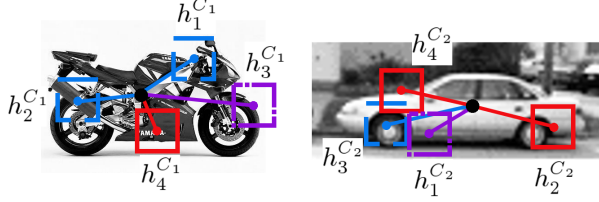
where  $C_j$  refers to the category and  $f_i$  is a feature. An equivalent expression may be written for the rest of categories and for the background (B) class. For each of the classes we seek to maximize its posterior probability ratio w.r.t. the background class. By removing the priors  $P(f_1, f_2, \dots, f_n)$ , common for all the classes, and assuming uniform prior probabilities,  $P(C_j) = P(B)$ , the posterior probability may be written by the likelihood ratio

$$\log \frac{P(C_j | f_1, f_2, \dots, f_n)}{P(B | f_1, f_2, \dots, f_n)} = \log \frac{P(f_1, f_2, \dots, f_n | C_j)}{P(f_1, f_2, \dots, f_n | B)} . \quad (4)$$

Since computing the complete joint probability for a large feature set is not feasible, it is split into  $m$  subsets ( $F_i = \{f_1, f_2, \dots, f_r\}$ ), with  $r = n/m$ . These feature subsets are known as Ferns, and assuming they are independent, their joint log-probability is computed as

$$\log \frac{\prod_{i=1}^m P(F_i | C_j, g_i)}{\prod_{i=1}^m P(F_i | B, g_i)} = \sum_{i=1}^m \log \frac{P(F_i | C_j, g_i)}{P(F_i | B, g_i)} , \quad (5)$$

where the parameter  $g_i$  ( $g \in \mathbb{R}^2$ ) corresponds to the image spatial location where the Fern  $F_i$  is evaluated, measured from the object image center. Each Fern captures the co-occurrence of  $r$  binary features computed



**Figure 3.** Category-specific classifiers. Several weak classifiers for different classes built from the same pool of features.

on the HOG space, and encodes object local appearances. Its response is represented by a combination of boolean outputs. For instance, the observation  $z_i$  of a Fern  $F_i$  made of  $r = 3$  features with binary outputs 0, 1, 1, would be  $(011)_2 = 3$ . In other words, each Fern maps 2D image coordinates to a  $K = 2^r$ -dimensional space

$$F : x \rightarrow z, \quad x \in X, \quad z \in \mathbb{R}. \quad (6)$$

The Fern probability may then be written using the class conditional probability, the Fern location and the feature set observations:

$$\sum_{i=1}^m \log \frac{P(F_i|C_j, g_i, z_i = k)}{P(F_i|B, g_i, z_i = k)}, \quad k = 1, 2, \dots, K, \quad (7)$$

with  $k$ , the observation index.

## 4 Building Robust Category Classifiers

The Random Ferns  $F_i$ , defined in the previous section, are shared by all of the object categories. This is what we call a *pool* ( $\vartheta$ ) of Random Ferns (See Algorithm 1). We next build a robust category-specific classifier as a linear combination of weak classifiers, where each of them is based on a Random Fern selected from the feature pool, with an associated spatial image location.

More formally, we want to build the object category classifier  $H^j(x)$ , that yields the Ferns  $F_i$  and locations  $g_i$  that are most discriminative, that is, that maximizes Eq.7. This is achieved by means of a Real Adaboost algorithm [13], that iteratively assembles weak classifiers and adapts their weighting values.

Then, a category-specific classifier based on Random Ferns is defined as

$$H^j(x) = \sum_{t=1}^T h_t^j(x) > \beta_j, \quad (8)$$

where  $\beta_j$  is a threshold with a zero default value,  $h_t^j$  is a weak classifier defined by

$$h_t^j(x) = \frac{1}{2} \log \frac{P(F_t|C_j, g_t, z_t = k) + \epsilon}{P(F_t|B, g_t, z_t = k) + \epsilon}, \quad k = 1, \dots, K, \quad (9)$$

---

### Algorithm 1 Detector computation.

---

- 1: Given a number of weak classifiers  $T$ , a shared feature pool  $\vartheta$  consisting of  $M$  Random Ferns, and  $N$  image samples  $(x_1, y_1) \dots (x_n, y_n)$ , where  $y_i \in \{+1, -1\}$  is the label for object category  $C_j$  and background classes  $B$ , respectively:
  - 2: Initialize sample weights  $D_1(i) = \frac{1}{N}$ .
  - 3: **for**  $t = 1$  to  $T$  **do**
  - 4:   **for**  $m = 1$  to  $M$  **do**
  - 5:     **for**  $g \in x$  **do**
  - 6:       Under current distribution  $D_t$ , calculate  $h_{m,g}$  and its distance  $Q_{m,g}$ .
  - 7:     **end for**
  - 8:   **end for**
  - 9:   Select the  $h_t$  that minimizes  $Q_t$ .
  - 10:   Update sample weights.  

$$D_{t+1}(i) = \frac{D_t(i) \exp[-y_i h_t(x_i)]}{\sum_{i=1}^N D_t(i) \exp[-y_i h_t(x_i)]}$$
  - 11: **end for**
  - 12: Final strong classifier.  

$$H^j(x) = \text{sign} \left( \sum_{t=1}^T h_t^j(x) - \beta_j \right)$$
- 

and  $\epsilon$  is a smoothing factor. At iteration  $t$ , the probability  $P(F_t|C_j, g_t, z_t)$  is computed under the distribution of sample weights  $D(i)$  as

$$P(F_t|C_j, g_t, z_t = k) = \sum_{i: z_t(x_i) = k} D_t(i), \quad k = 1, \dots, K. \quad (10)$$

The classification power of each weak classifier is measured by means of the Bhattacharyya distance between object and background distributions. We then choose the classifier  $h_t^j$  that minimizes the following criterion,

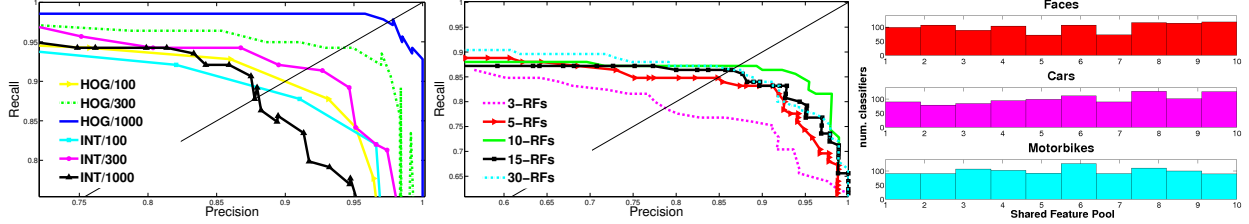
$$Q_t = 2 \sum_{k=1}^K \sqrt{P(F_t|C_j, g_t, z_t = k) P(F_t|B, g_t, z_t = k)}. \quad (11)$$

Figure 3 depicts one example of several weak classifiers retrieved for two object categories that share the same features (Random Ferns).

## 5 Experiments

We next validate several aspects of our algorithm on public datasets, and compare its performance to state-of-the-art methods that are focused on single object detection. The datasets we consider are the well-known UIUC car-side dataset [1], the TUD motorbike dataset [5] and the Caltech face dataset [4].

**HOG-based Ferns.** The proposed method has been tested using Ferns on the HOG and intensity domains. This experiment is performed over the UIUC car-side dataset that has 108 images of 139 cars at different scales. Figure 4-Left shows how using HOG-based features we obtain better results than when using intensity



**Figure 4.** Detection performances. Left: HOG vs intensity based features under varying numbers of weak classifiers. Middle: Shared feature pool size. Right: Selected RFs for several categories.

Method	UIUC Multi-scale	Caltech Faces	TUD motorbikes
[1]	39.6%	-	-
[4]	-	96.4%	-
[5]	87.8%	-	81.0%
[10]	90.6%	-	-
[14]	-	94.0%	-
[9]	94.7%	-	89.0%
[8]	95.0%	-	87.0%
[6]	98.6%	-	-
<b>Our Method</b>	<b>97.8%</b>	<b>99.1%</b>	<b>86.7%</b>

**Table 1.** Performances on Object Categorization.

based features. In addition, we observe that increasing the number of weak classifiers also improves detection performance. Yet, since the number of features remains the same, the cost of the algorithm does not significantly increase.

**Shared Feature Pool Size.** The classifier performance has been evaluated in terms of the feature pool size. This evaluation is performed on the TUD motorbike dataset that consists of 115 test images under challenging situations such as occlusions. For training, we have used about 400 motorbike images from the Caltech dataset. Figure 4 shows that with only 10 Random Ferns the classifier achieves remarkable results comparable to state-of-the-art methods specifically tailored to single object detection (Table 1).

**Feature Sharing.** Figure 4 shows the Random Ferns distributions for different category-specific classifiers. Since these classifiers share the same features, the computational cost of detecting multiple categories is  $O(M)$  where  $M$  is the cost of feature computation. This is in contrast to the  $O(KM)$  cost of using different features per category, with  $K$  the number of categories.

## 6 Conclusions

We have presented an algorithm for multiple object detection, that makes use of a common pool of features, computed using Random Ferns over the HOG domain. We have shown that sharing common features yields an efficient multiple object detector while the detection rates are similar to current approaches that compute specific features for each category.

## References

- [1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. *ECCV*, 2002.
- [2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.
- [3] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008.
- [4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003.
- [5] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. *ICCV*, 2005.
- [6] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient sub-window search. *CVPR*, 2008.
- [7] I. Laptev. Improving object detection using boosted histograms. *IVCJ*, 2009.
- [8] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 2008.
- [9] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. *CVPR*, 2006.
- [10] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. *CVPR*, 2006.
- [11] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. *CVPR*, 2007.
- [12] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. *CVPR*, 2008.
- [13] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999.
- [14] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. *ICCV*, 2005.
- [15] J. Shotton, M. Johnson, and R. Cipolla. Semantic texon forests for image categorization and segmentation. *CVPR*, 2008.
- [16] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007.
- [17] Q. Zhu, S. Avidan, M. Ye, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *CVPR*, 2006.