Detection Performance Evaluation of Boosted Random Ferns

Michael Villamizar, Francesc Moreno-Noguer, Juan Andrade-Cetto
and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
{mvillami,fmoreno,cetto,sanfeliu}@iri.upc.edu

Abstract. We present an experimental evaluation of Boosted Random Ferns in terms of the detection performance and the training data. We show that adding an iterative bootstrapping phase during the learning of the object classifier, it increases its detection rates given that additional positive and negative samples are collected (bootstrapped) for retraining the boosted classifier. After each bootstrapping iteration, the learning algorithm is concentrated on computing more discriminative and robust features (Random Ferns), since the bootstrapped samples extend the training data with more difficult images.

The resulting classifier has been validated in two different object datasets, yielding successful detections rates in spite of challenging image conditions such as lighting changes, mild occlusions and cluttered background.

1 Introduction

In the last years a large number of works have been addressed for detecting objects efficiently [1–3]. Some of them use boosting algorithms with the aim of computing a strong classifier from a collection of weak classifiers which could be calculated very fast. One well-known and seminal work is the detector proposed by Viola and Jones [2]. It relies on simple but fast features (Haar-like features) that in combination, by means of AdaBoost, yields a powerful and robust classifier. Nevertheless, this sort of methods depends strongly on the number of image samples used for learning the classifier. In a few words, more training samples (object and non-object samples) means a better detector performance. This fact is an important drawback when datasets with a reduced number of images are used. Traditionally, methods tend to introduce image transformations over the training data in order to enlarge the dataset, and to collect a large number of random patches over background images as negative samples.

Recently, Random Ferns [4] in addition to a boosting phase have shown to be a good alternative for detecting object categories in an efficient and discriminative way [3]. Although this method has shown impressive results in spite of its simplicity, its performance is conditioned to the size of its training data. It is because boosting requires a large number of training samples and because each Fern has an observation distribution

that depends on the number of features forming the Fern. As larger this distribution is, more training samples are needed.

In this work, we evaluate the Boosted Random Ferns (BRFs) according to the size and quality of the training data. Furthermore, we demonstrate that adding a bootstrapping phase during the learning step the BRFs improves its detection rates. In summary, the bootstrapping phase allows to having more difficult samples with which to retraining the BRFs and therefore to obtain a more robust and discriminative object classifier.

The remainder of present work is organized as follows: Section 2 describes the approach for learning the object classifier by means of an iterative bootstrapping phase. Subsequently, the computation of the BRFs is described (Section 2.1). The procedure to bootstrapping images from the training data is shown in Section 2.2. Finally, the experimental validation and conclusions are presented in Sections 3 and 4, respectively.

2 The Approach

Detecting object instances in images is addressed by testing an object-specific classifier over images using a sliding window approach. This classifier is computed via Boosted Random Ferns, that are basically a combination of feature sets which are calculated over local Histograms of Oriented Gradients [7]. Each feature set (Random Fern) captures image texture by using the output co-occurrence of multiple local binary features [6].

Initially, the Boosted Random Ferns are computed using the original training data, but subsequently, the classifier is iteratively retrained with the aim of improving its detection performance using an extended and more difficult training data. This is done via a bootstrapping phase that collects new positive and negative samples by testing the previously computed classifier over the initial training data. As a result, a more challenging training dataset with which to train following classifiers is obtained.

2.1 Boosted Random Ferns

The addressed classifier consists on a boosting combination of Random Ferns [4]. This classifier was proposed in the past with the aim of having an efficient and discriminative object classifier [3]. More formally, the classifier is built as a linear combination of weak classifiers, where each of them is based on a Random Fern F which has associated a spatial image location g. The boosting algorithm [5] computes the object classifier $H(x)$ by selecting, in each iteration t, the Fern F_t and location g_t that best discriminates positive samples from negative ones. The algorithm also updates a weight distribution over the training samples in order to focus its effort on the hard samples, which have been incorrectly classified by previous weak classifiers. The result is an assembling of Random Ferns which are computed at specific locations. To increase even more the computational efficiency, a shared pool ϑ of features (Random Ferns) is used [6]. It allows to reduce the cost of feature computation.

Then, the Boosted Random Ferns can be defined as:

$$H(x) = \sum_{t=1}^{T} h_t(x) > \beta \,, \tag{1}$$

where x is an image sample, β is the classifier threshold, and h_t is a weak classifier computed by

---

Algorithm 1 Boosted Random Ferns Computation.

1: Given a number of weak classifiers T, a shared feature pool ϑ consisting of M Random Ferns, and N image samples $(x_1, y_1)...(x_N, y_N)$, where $y_i \in \{+1, -1\}$ indicates the label for object C and background classes B, respectively:

2: Initialize sample weights $D_1(x_i) = \frac{1}{N}$, where $i = 1, 2, ..N$.

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 the weak classifier $h_{(m,g)}$ 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.
$$D_{t+1}(x_i) = \frac{D_t(x_i)\exp[-y_i h_t(x_i)]}{\sum_{i=1}^{N} D_t(x_i)\exp[-y_i h_t(x_i)]}$$

11: **end for**

12: Final strong classifier.
$$H(x) = sign\left(\sum_{t=1}^{T} h_t(x) - \beta\right)$$

$$h_t(x) = \frac{1}{2}\log\frac{P(F_t|C, g_t, z_t(x) = k) + \varepsilon}{P(F_t|B, g_t, z_t(x) = k) + \varepsilon}, \quad k = 1, .., K. \tag{2}$$

The variables C and B indicate the object (positive) and background (negative) classes, respectively. The Fern output z is captured by the variable k, while the parameter ε is a smoothing factor.

At each iteration t, the probabilities $P(F_t|C, g_t, z_t)$ and $P(F_t|B, g_t, z_t)$ are computed under the weight distribution of training samples D. This is done by

$$P(F_t|C, g_t, z_t = k) = \sum_{i: z_t(x_i) = k \wedge y_i = +1} D_t(x_i), \quad k = 1, .., K. \tag{3}$$

$$P(F_t|B, g_t, z_t = k) = \sum_{i: z_t(x_i) = k \wedge y_i = -1} D_t(x_i), \quad k = 1, .., K. \tag{4}$$

The classification power of each weak classifier h is measured by means of the Bhattacharyya distance Q between the object and background distributions. The weak classifier h_t that minimizes the following criterion is selected,

$$Q_t = 2\sum_{k=1}^{K} \sqrt{P(F_t|C, g_t, z_t = k)P(F_t|B, g_t, z_t = k)}. \tag{5}$$

The pseudocode for computing the BRFs is shown in algorithm 1. For present work, all classifiers are learned using the same classifier parameters, since our objective is to show their detection performances in connection with only the training data. Therefore, each boosted classifier is built by 100 weak classifiers and using a shared pool ϑ consisting of 10 Random Ferns. Besides, each Fern captures the co-occurrence of 7 random binary features that are computed over local HOGs [7].

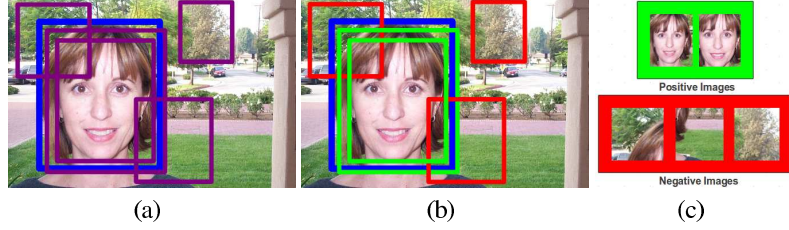|		
(a)	(b)	(c)

Fig. 1. The Bootstrapping phase. Once the classifier is computed, it is tested over training images (a). According to the ground truth (blue rectangle) and the bootstrapping criterion (eq. 6), current detections (magenta rectangles) are classified as either positive or negatives samples (b) . Positive samples correspond to object instances (green rectangles), while negative ones are background regions (blue rectangles). They are result of false positives emitted by the classifier (c).

2.2 The Bootstrapping Phase

In this section is described how the training data is enlarged with more difficult image samples in order to retraining the object-specific classifier. The procedure is carried out as follows: given the training data, consisting of sets of positive and negative samples, the classifier is first computed using those initial samples. Once the classifier has been computed, it is tested over the same training images with the aim of extracting new positive and negative samples.

The criterion for selecting the samples is based on the overlapping rate r between detections (bounding boxes) and the image ground truth given by the dataset. If that rate is over a defined threshold (0.3 by default), such detections are considered as new positive samples, otherwise, they are considered as false positives are aggregated to set of negative images. The overlapping rate can be written as:

$$r = \frac{|B_{GT} \bigcap B_D|}{|B_{GT} \bigcup B_D|} \tag{6}$$

where B_{GT} indicates the bounding box for the ground truth, and B_D is the bounding box for current detection. New positive samples represent the initial object samples under some image transformations like scales or image shifts (see Figure 3). The robustness of the classifier is then increased thanks to those images since object samples with small transformations are considered in the learning phase. By the other hand, the extracted negative samples correspond to background regions which have been misclassified by the current classifier. Those images force the boosting algorithm to seek out the most discriminative Ferns towards object classification in the following iterations. The extraction of new training samples is visualized in Figure 1.

3 Experiments

Several experiments have been carried out in order to show the strong dependency between the performance of the addressed classifier and the size of training data used for its learning. Moreover, it is observed that bootstrapping new samples yields more robustness and discriminative power to the object classifier. For testing the resulting classifier two different object datasets have been used. They are the Caltech Face dataset [8] and the Freestyle Motocross dataset proposed in [3].
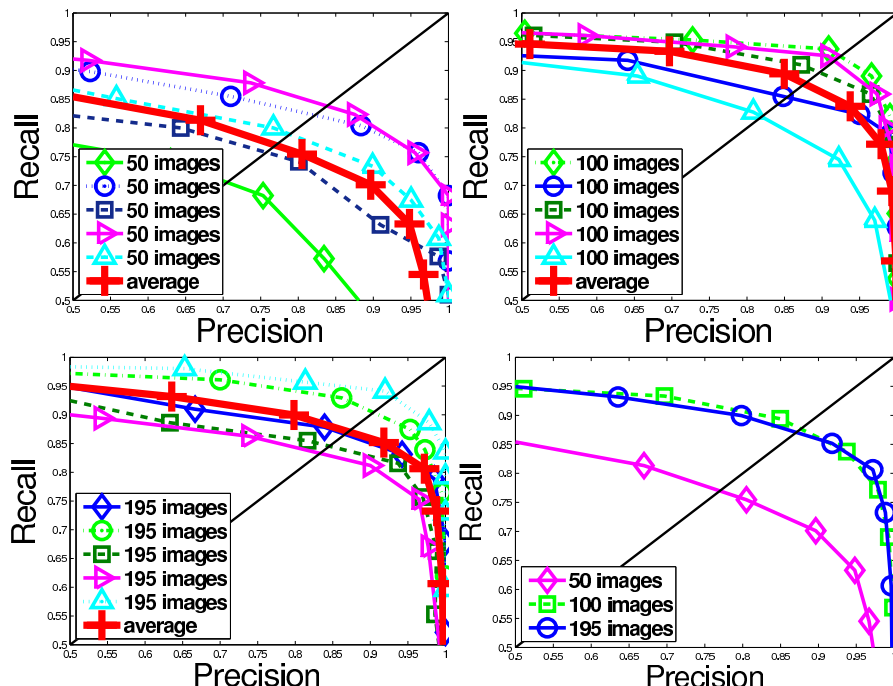
Fig. 2. Detection performance using several sizes of training data. For each case, the classifier has been learned five times due to it is based on random features. Subsequently, the average performance is calculated. Bottom-right figure shows the average curves.

Caltech Face Dataset - This dataset was proposed in [8] and it contains frontal faces from a small group of people. The images show extreme illumination changes because they were taken in indoor and outdoor scenes. The set of images is formed by 450 images. For present work, we use the first 195 images for training and the rest ones for validation. It is important to emphasize that images used for validation do not correspond to people used for training the classifier. In addition, this dataset also contains a collection of background images that are used as negative samples.

Freestyle Motocross Dataset - This dataset was originally proposed for validating a rotation-invariant detector given that it has two sets of images, one corresponding to motorbikes with rotations in the image plane, and a second one containing only motorbikes without rotations [3]. For our purposes the second set has been chosen. This set contains 69 images with 78 motorbike instances under challenging conditions such as extreme illumination, multiple scales and partial occlusions. The learning was done using images from the Caltech Motorbike dataset [8].

3.1 Training Data

In this section, the detection performance of Boosted Random Ferns is measured in terms of the size of the training data. Here, the classifier is learned without a bootstrapping phase and using different number of training images, both positive and negative images. The classifier aims to detecting faces in the Caltech Face dataset. Figure 2

Fig. 3. Bootstrapped images. Some initial positive images are shown at top row. After one bootstrapping iteration, the algorithm extracts new positive and negative samples. New positive images contain the target object at different scales and shifts (middle row). Contrary, new negative images contain difficult backgrounds and some portion of the object (bottom row).

shows the detection curves (Recall-Precision Curves) for each data size. As the classifier relies on random features (Random Ferns), the learning has been repeated five times and the average performance has been calculated. We can note that detection rates increase with the number of image samples, and that using 100 images the classifier yields similar rates to using the entire training data (195 images).

3.2 Bootstrapping Image Samples

In order to increase the number of training samples and the quality of them for the classification problem, an iterative bootstrapping phase is carried out. In each iteration, the algorithm collects new positive and negative samples from the training data. The result is generating more difficult negative samples in order to force the boosting algorithm to extract more discriminative features. Furthermore, the extraction of new positive images allows to having a wide set of images that considers small transformation over the target object (e.g object scales). To illustrate the extraction of bootstrapped samples, Figure 3 shows some new positive and negative samples which are then used for re-training the object classifier. These images were collected after just one bootstrapping iteration. Negative samples correspond to portions of faces and difficult backgrounds.

With the aim of measuring the influence of the bootstrapping phase over the detection performance, two experiments has been elaborated. The first one consists on testing the face classifier over the test images of Caltech dataset, that is, over the remainder 255 images. Similarly, the second experiment consists on evaluating the classifier over the same dataset but adding an additional set of 2123 background images in order to visualize its robustness to false positives. Both experiments have been carried out using different number of bootstrapping iterations. The results are depicted at Figure 4.

Figure 4(a) shows the results for the first experiment. We see that adding a bootstrapping phase the detection rates increase remarkably. After one iteration, the rates are very similar for all cases and achieve an almost perfect classification. For the second experiment, the testing is carried out one more time but using the large set of background images. In Figure 4(b) are shown the detection performances for each case in comparison with the results of not using the additional images. It is realized that the addition of background images for validation affects considerably the detection rate of
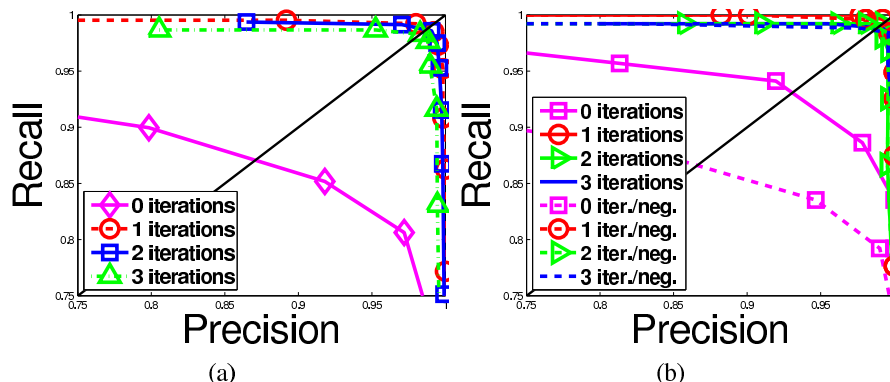
Fig. 4. Detection performance in terms of number of bootstrapping iterations. (a) The retrained classifier is evaluated over the original dataset (first experiment). (b) For detection evaluation, 2123 background images are added to set of testing images (second experiment). The classifier without retraining shows a considerable detection rate reduction when background images are added.

the object classifier trained without a bootstrapping phase (0 iterations). It demonstrates that trained classifier tends to yield more false positives, while the classifiers computed after bootstrapping are robust to them. Their detection rates are very similar and do not seem to decrease with the extended testing set.

3.3 Freestyle Motocross Dataset

In Figure 5(a) are shown the detection curves for the Freestyle Motocross dataset. It is seen how the number of bootstrapping iterations also improves the detection rates. Figure 5(b) depicts the detection performances of learning the object classifier using different sets of training images. In this experiment, the positive images are generated by adding affine transformations over the initial positive samples, while the negative ones are created by collecting random patches over background images. The bootstrapped classifier outperforms in all cases. This demonstrates that bootstrapped images help to the boosting algorithm to build a more robust classifier. Finally, some detection results for this dataset are shown in Figure 6.

4 Conclusions

We have presented an experimental evaluation of the Boosted Random Ferns as an object classifier. The strong dependency between detection performance and the size of training data has been evidenced. To improve detection results, an iterative bootstrapping phase has been presented. It allows to collect new positive and negative samples from the training data in order to force the learning algorithm to compute more discriminative features towards object classification. Furthermore, we can conclude that with just one bootstrapping iteration, the bootstrapped classifier achieves impressive detection rates and robustness to false positives.

References

1. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007.
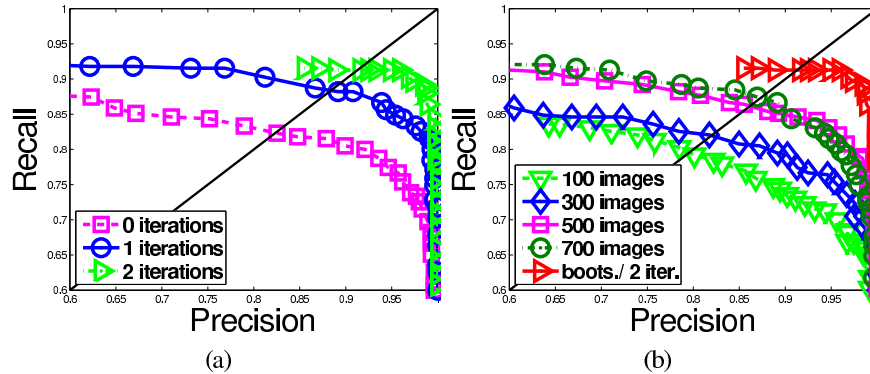
Fig. 5. Detection curves for the Freestyle Motocross dataset. (a) The bootstrapping phase improves the detection classification. (b) The bootstrapped classifier outperforms to traditional classifiers based on creating and collecting a large number of image samples.
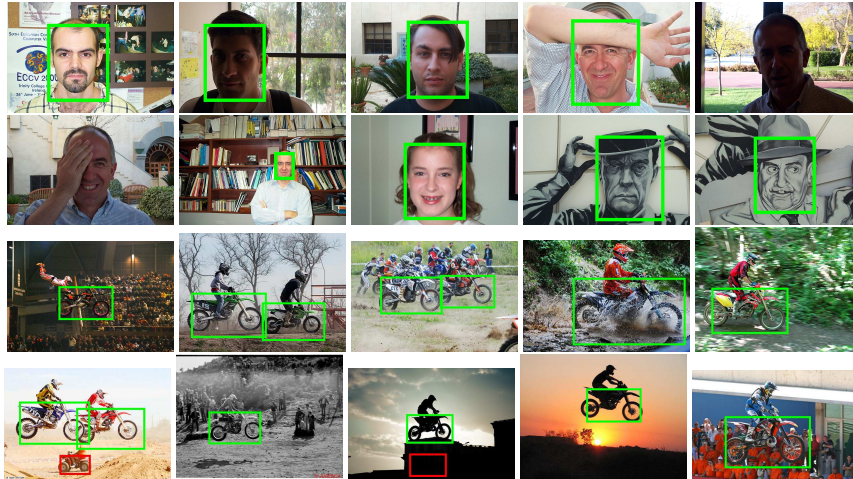

Fig. 6. Some detection results. Green rectangles correspond to object detections. The resulting classifier is able to detect object instances under difficult image conditions.

2. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.
3. M. Villamizar, F. Moreno-Noguer, J. Andrade-Cetto, and A. Sanfeliu. Efficient rotation invariant object detection using boosted random ferns. *CVPR*, 2010.
4. M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. *CVPR*, 2007.
5. R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 1999.
6. M. Villamizar, F. Moreno-Noguer, J. Andrade-Cetto, and A. Sanfeliu. Shared random ferns for efficient detection of multiple categories. *ICPR*, 2010.
7. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.
8. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2003.