Randomized Trees for Real-Time One-Step Face Detection and Recognition

Vaishak Belle, Thomas Deselaers, Stefan Schiffer
Computer Science Department, RWTH Aachen University
{*belle,deselaers,schiffer*}*@cs.rwth-aachen.de*

Abstract

We present a system for detecting and recognizing faces in images in real-time which is able to learn new identities in instants. In mobile service robotics, interaction with persons is becoming increasingly important, real-time performance is required and the introduction of new persons is a necessary feature for many applications. Although face detection and face recognition are well studied, only a few papers address both problems jointly and only few systems are able to learn to identify new persons quickly. To achieve real-time performance on modest computing hardware, we use random forests for both detection and recognition, and compare with well-known techniques such as boosted face detection and support vector machines for identification. Results are presented on different datasets and compare favorably well to competitive methods.

1. Introduction

We introduce a new framework for the detection and recognition of human faces in images. To create an extremely fast system that is able to run in real-time on a *service robotic* platform, we use random forests (RF) [6] for the detection, recognition and learning of faces and identities in images.

Service robots aim at offering assistance to humans in general and to people with disabilities in particular. Such robots socially interact with human beings, i.e. they respond dynamically to requests and communicate. The interaction can be more "natural" if the robot can identify persons it encounters. We envision that on encountering unknown identities, the robot may introduce itself and add the new identity to its knowledge base. Therefore, a fast and reliable face recognition system is required, which, in a first step, detects faces and, in a second step, recognizes the persons. This task is complicated by the computational limitations of common robots which typically have only modest computing power that additionally have to be shared with other robot components such as motion control and localisation. In the literature, face detection and face recognition, typically, are addressed separately, although they share identical structural foundations [13, 7]. The work presented here is a one-step system that addresses both face detection and recognition in an integrated framework using *random forests* (RF). The advantages of RFs have been thoroughly investigated [2] and it has been shown that RFs are fast and have good generalization capabilities.

Additionally, we introduce *identity learning*, as an extension to this framework. A collection of face images for a new identity captured by the robot can be added to the knowledge base in real-time, i.e. the robot learns to recognize new persons from that instant. This is made feasible as a result of a very short training time. Similar to other approaches, we use local descriptors, which are known to be an excellent means for face authentication [9].

We compare the detection performance of our approach to the AdaBoost face detector [7], an excellent implementation of which is freely available in OpenCV[1]. The AdaBoost face detector is often considered a quasi standard [10] for face detection, comparable to detection with neural networks [12, 11]. Results of our recognition are later compared to those achieved with support vector machines (SVM), which have been succesfully used for face identification [5] before.

RFs have previously been used for biological image classification [8] where, similar to our approach randomly sampled rectangles are used as test candidates in the training procedure. RFs have also been successfully used for general object/image classification [1]. RFs have been used for foreground/background segmentation in a video chat application and systematically compared to boosting and bagging classifiers [14]. An approach most similar to our own has been proposed for real-time gesture recognition. Here, an RF is used for segmentation and subsequently a second RF is used for the recognition [3].

2. Random Forests

An RF is a collection of random trees (RT). Random trees are structurally identical to classical decision trees but are trained differently. During training not an exhaustive search of the possible test candidates is considered but only a randomized subset in order to allow for creating several different and independent RTs.

In our approach, we create a large number of randomized test candidates to the training procedure in each iteration. Here, a randomized test candidate is a local *fea-*

[1] http://www.intel.com/technology/computing/opencv/

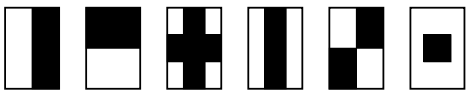

Figure 1. The six different Haar features used in our RTs.

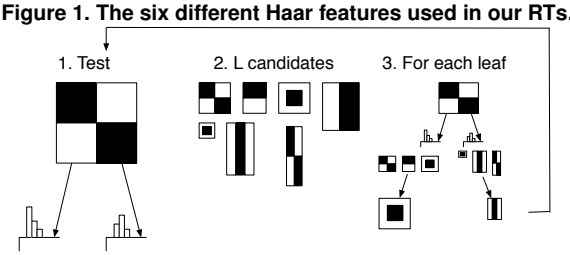

Figure 2. Schematic representation of the training method.

ture on a rectangle of random dimensions and at a random location in the training data images.

2.1. Features

We use Haar features similar to those used in the boosted face detection cascade by Viola and Jones [7]. They allow for fast evaluation and they are known to be good features for face detection. We allow the RT to choose among six different Haar features (depicted in Figure 1) and during training, in each iteration a set of these features is generated by choosing the type (one of the six), the size, and the test threshold randomly. A test is successful, if the sum of the pixel gray values in the black area minus the sum of the pixel gray values in the white area is higher than the test threshold. These tests can efficiently be calculated using integral images [7].

2.2. Tree Training

An RT is grown in an iterative procedure. In each iteration, a set of L test candidates is randomly generated. The test candidates are applied to all the training samples at a node and the entropy gain is measured on the corresponding split. The best candidate is chosen and the left and right branches are appended to the existing node and the procedure continues until the leaf nodes maintain training data of only a single class.

The training data for face detection consists of segmented faces (positives) and patches of background images (negatives). These images are all scaled to a common size and subwindows (up to a heuristically determined percentage of the image size) are sampled as test candidates. The initialization of the RT is done using a normal training iteration by choosing the candidate among a set of L random test candidates that optimizes the classification according to the entropy gain. To measure the entropy gain we use *class histograms*. The class histogram compares the number of images for each class in a leaf node on a split to the image count in the parent node to calculate an information gain. The candidate with the highest gain is chosen for each node in each iteration. The divided training samples are propagated to the left and right children of a split node. The training process is schematically shown in Figure 2. This procedure is repeated until either a predefined threshold is reached or until perfect class separation is reached.

It is known that RTs are prone to overfitting if trained long enough (i.e. too many nodes added). It is furthermore known that averaging classifiers improves the generalization ability and to benefit from both effects, RFs are one possibility [2]. Following the aforementioned ideas, we create a RF by training a set of T RTs simultaneously.

2.3. Detection

To detect faces in an input image, we use integral images to allow for rapid evaluation of our RTs. Then, each RT is used in a sliding windows manner on an image to determine for each position whether the surrounding area is a face or not. Eventually, in each RT a leaf node is reached delivering a probability (by looking up the relative frequency of the nodes class histogram) for the region to contain a face:

$$\text{FMP} = \frac{\text{number of face images in leaf node}}{\text{total number of images in leaf node}}. \quad (1)$$

Each RT is applied in this manner, resulting in T probabilities at each position. These probabilities are fused to determine the absence/presence of a face for each position.

To detect faces of different sizes, either the input image is scaled or the RTs are adapted by scaling the dimensions of tests, which is faster because it does not require for recalculating the integral image.

The fusion of the different RTs at different scales is performed using an aggressive merging step, much simpler than the one proposed in [7]. Our merging technique finds an *area of interest* by listing neighbors of a detection window. Then, a weighted average over the detected face areas is computed to deliver the final detection, where we chose the weights from experiments on a preliminary in-house dataset.

2.4. Recognition

For face recognition, we have to discriminate between P known identities with the additional option of classifying a person as unknown. In our integrated joint detection and recognition approach, we use the face/no-face information jointly with the identity labels. Therefore, we first train a normal detection RF as described above, then we add the identity labels and continue training to be able to discriminate among identities (i.e. grow additional leaves until these contain either only one identity or the "*unknown identity*" label).

The detected face is propagated into the leaf nodes of the detection RT, the corresponding class histograms are extended with the additional classes and additional training iterations are applied to discriminate among the identities and the large set of unknown identities. This step is repeated for each RT and a final identity label is obtained using majority voting.

New Identities. This training step only needs few iterations, thus this method allows for adding new identities on the fly in the same way as initially the detection RT is extended to become a joint detection and recognition RT. Similarly, new identities can be added to a detection/recognition tree: A new set of face images are supplied for the new identity. These images are propagated to the leaf nodes of the RT and additional nodes are added until the new identity is separated from other identities.

Another option to achieve detection and recognition is to first create a detection RF and then in a second step to create a recognition RF. A similar setup was presented for the task of hand gesture and object recognition in [3]. This setup is also open to the learning of new identities by rebuilding the recognition forest.

2.5. Parameters

We provide insights into the parameters modeled and the values experimentally considered as optimal. The *forest size* T is theoretically and empirically related to the classification accuracy. We grow *ten* RTs. The *feature size* limits the dimensions of the rectangles sampled and we find $0.5 \cdot \sqrt{K \times L}$, where $K \times L$ is the dimension in pixels of the training samples, as optimal. We set $L = 200$ for our evaluations. Note that it has been shown, with as few as 3 or 5 RTs, acceptable performance was achieved [3].

3. Experimental Evaluation

In this section, we present the results from the experimental evaluation of our proposed methods and discuss the integration into our RWTH-Aachen RoboCup@Home [4] mobile robots.

3.1. Detection

We compare our detection performance to the AdaBoost face detector [7] of the OpenCV library. For this purpose, we use the pre-trained model delivered with the library and we also compare using a model that we trained using our training data. It is well known that the pre-trained model was very carefully engineered and performs extremely well. Unfortunately, it is unclear which data was used for training and thus comparison with this model is not completely fair.

Our training collection includes a collection of 4,000 faces and 4,000 background images collected from various sources on the Internet, all scaled to 24×24 pixels. We trained the AdaBoost detector for four and 13 days, denoted as *B-4* and *B-13*, respectively. Training was performed on an Opteron machine with 2.2GHz.

The RF is trained on the same data, where we allow for up to 8,000 nodes (which allows perfect discrimination between all training samples). Here, the training takes approx. 400 sec/RT on a 2.0 GHz Intel Core2Duo machine.

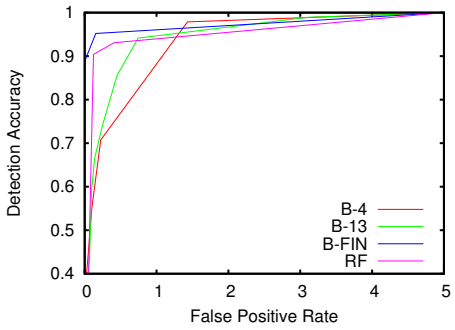

Figure 3. ROC curve for the detection results.

Table 1. Results (error rates) from the recognition.

Test Collection	RF	SVM
Yale[RF]	46.6	69.4
BioID[RF]	86.3	68.63
Yale[B-FIN]	15	7.7
BioID[B-FIN]	13.5	2.7

ROC curves for the detection results are shown in Figure 3 for a combination of the MIT+CMU test collection [12], Yale Face Database[2] and the BioID Face Database[3]. In the figure, both *B-4* and *B-13* attain a detection accuracy of 90% with a very high false positive rate, emphasizing the need for weeks of training time for AdaBoost models [10]. RFs outperform *B-4* and *B-13*. Further, RFs perform comparatively to *B-FIN* albeit with a slightly higher false positive rate. We however note that our detections suffer from alignment errors. One option to tune our method would be to incorporate a more carefully engineered filtering/merging step similar to the one proposed by Viola and Jones [7].

3.2. Recognition

The recognition is evaluated after detection on the BioID and Yale collections (here, we cannot use the MIT+CMU test set since identities are not annotated). We compare the performance of our RF classifier to an SVM classifier. Results are given in Table 1. Due to the alignment errors of our detection framework, the error rates in Yale[RF] and BioID[RF] are rather high. If, however, we use the pre-trained OpenCV face detector (*B-FIN*), a drastic performance improvement can be observed in the corresponding Yale[B-FIN] and BioID[B-FIN]. In general, it must be noted that the SVM outperforms the RFs but at the cost of a) a much higher training time, and b) a much higher time required for the classification.

To further analyse the difference in the classification speed, we trained RTs for recognition only and detail the training time. It takes about 7.5ms to create an RF consisting of ten RTs, each grown up to a 1000 nodes with

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[3] http://www.bioid.com/downloads/facedb/

Table 2. Relations between L,N, number of nodes, the training time, and the recognition rate (RR).

Dataset	Nodes	L	N	Time	RR [%]
Detection	8000	200	8000	400s	–
BioID	1000	50	464	0.75ms	86.5
Lab	200	50	88	0.25ms	92.86
Yale	200	50	40	0.2ms	85.0

$L = 50$ on our BioID training collection of 464 face images and six identities. As our Yale training collection is smaller, a total of 40 face images and 5 identities, we are able to build ten RTs in only 2ms by letting each RT grow up to 200 nodes and $L = 50$. Clearly, if the number of training images per identity can be approximately estimated in advance, we can optimize the corresponding node count ($2 \cdot$ number of identities $\cdot$ images per identity) and grow an RF extremely rapidly with rank-1 recognition rates comparable to SVM.

3.3. Realistic Scenario

The method was developed for mobile service robots, we also evaluate it on our in-house lab dataset. Images with faces in typical backgrounds that a mobile robot encounters were captured consisting of four identities and up to 30 images per per identity. Sample images are depicted in Figure 4. Faces were collected using *B-FIN*. We use 22 images for training and 8 to evaluate the performance. We train an RF of ten RTs with up to 200 nodes, training takes 2.5 milliseconds and the error rate is 7.1%. Researchers may find Table 2 useful for insights into the duration of RT growth. Here, the number of nodes, test candidates and training data that contribute to the training time are enumerated on the experiments presented in this paper. The rank-1 recognition rates correspond to the error rates discussed in the table above and that on our lab test.

The described system was employed by a RWTH Aachen University robot at the recent RoboCup German Open competition[4] and at the RoboCup World Championship[5] in the RoboCup@Home league. The task included detecting, recognizing and learning faces in the arena (Figure 4) to hand over objects of interest to the respective personnel.

4. Conclusions

In this paper, we presented a framework for one-step face detection and recognition with low training time, which is able to learn new identities at any moment on the mobile robotic platform.

The proposed method is evaluated on different datasets and compared to the standard AdaBoost detection method and an SVM-based recognition system and it is shown that the new method is faster by several orders of magnitude in the training and testing time with only little deterioration of the accuracy.

[4]http://www.robocup-german-open.de/

[5]http://www.robocup-cn.org/

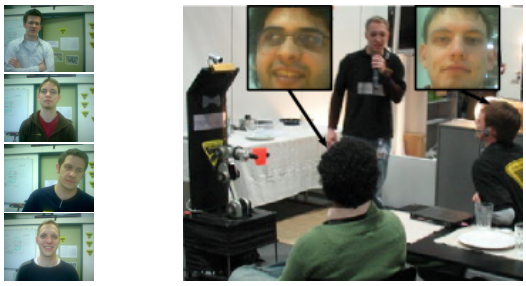

Figure 4. Left: examples from the lab test set. Right: detection, segmentation and recognition of faces at the 2008 RoboCup German Open, Hannover, Germany.

The problem of one-step detection and recognition is addressed using decision trees which can be trained to discriminate additional classes with only little effort and it is shown that the technique works well.

We plan to improve our detection aggregation technique in a similar way as the AdaBoost cascade does and we will employ the system on our service robot platform [4].

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, pp. 1–8, 2007.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] T. Deselaers, A. Criminisi, J. Winn, and A. Agarwal. Incorporating on-demand stereo for real time recognition. In *CVPR*, June 2007.

[4] S. Schiffer and G. Lakemeyer. Allemaniacs@Home 2008 team description. Technical report, RWTH Aachen University, Aachen, Germany 2008.

[5] G. Guo, S. Z. Li, and K. Chan. Face recognition by support vector machines. In *FG*, page 196, 2000.

[6] T. Ho. Random decision forest. In *ICDAR*, pp. 278–282, August 1995.

[7] M. Jones and P. Viola. Face recognition using boosted local features. In *ICCV*, April 2003.

[8] R. Marée, P. Geurts, and L. Wehenkel. Random subwindows and extremely randomized trees for image classification in cell biology. *Workshop of Multiscale Biological Imaging, Data Mining and Informatics*, 2007.

[9] R. Paredes, J. Perez-Cortes, A. Juan, and E. Vidal. Local representations and a direct voting scheme for face recognition. In *Workshop on Pattern Recognition in Information Systems*, pp. 71–79, July 2001.

[10] M. Pham and T. Cham. Fast training and selection of haar features using statistics in boosting-based face detection. *ICCV*, pp. 1–7, 2007.

[11] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, pp. 963, 1998.

[12] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 20(1):23–38, 1998.

[13] M. Turk and A. Pentland. Eigenfaces for recognition. *J Cognitive Neuroscience*, 3(1):71–86, 1991.

[14] P. Yin, A. Criminisi, J. Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *CVPR*, 2007.