AN ALTERNATIVE SCANNING STRATEGY TO DETECT FACES

Bala Subburaman Venkatesh [1,2], Sébastien Marcel [1]

[1]Idiap Research Institute, 1920, Martigny, Switzerland
[2]École Polytechnique Fédérale de Lausanne (EPFL), 1015, Lausanne,Switzerland
{venkatesh.bala, marcel}@idiap.ch

ABSTRACT

The sliding window approach is the most widely used technique to detect faces in an image. Usually a classifier is applied on a regular grid and to speed up the scanning, the grid spacing is increased, which increases the number of miss detections. In this paper we propose an alternative scanning method which minimizes the number of misses, while improving the speed of detection. To achieve this we use an additional classifier that predicts the bounding box of a face within a local search area. Then a face/non-face classifier is used to verify the presence or absence of a face. We propose a new combination of binary features which we term as μ-Ferns for bounding box estimation, which performs comparable or better than former techniques. Experimental evaluation on benchmark database show that we can achieve 15-30% improvement in detection rate or speed when compared to the standard scanning technique.

Index Terms— Face detection, Binary features, Naive Bayesian, Boosting

1. INTRODUCTION

The most popular technique to detect an object from an image is the sliding window approach since the pioneering work from Rowley [1]. With the introduction of cascade of classifiers and fast computation of features [2], it is possible to speed up the search for faces in an image. As more and more applications are integrating more processing (face tracking and recognition) in addition to face detection, and still needing them to run in real-time, it is necessary to speed up further without loosing much of the performance.

Most of the work on face detection concentrated on building a good classifier using Neural Networks [1, 3], SVM [4] or boosting [2, 5], but not much work was done to develop alternative scanning techniques. Given an image the standard scanning technique creates a pyramid of images according to a scale factor. Then a classifier is applied at every location in the image (usually on a regular grid) to detect an object. The grid spacing controls the speed of scanning process. Unfortunately, as the grid spacing is increased the number of miss detection increases. In this paper we propose an alternative scanning strategy, to speed up the search, while maintaining the detection rate. We analyze the probability for a classifier to fall within its detection range both for the standard scanning technique and for the proposed approach. The key to our alternative scanning technique is to build a classifier that predicts the face bounding box with high performance (both in speed and accuracy).

This paper is organized as follows. Section 2 gives the motivation behind our approach. The baseline face/non-face classifier is described in Section 3. In Section 4 we present the proposed approach and the face patch classifier which is used for estimating the bounding box. We show our experiment results in Section 5 and finally conclude and provide future directions in Section 6.

2. MOTIVATION

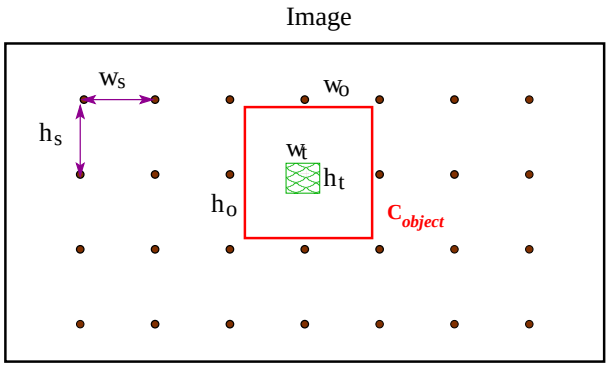
In this section we describe the motivation for coming up with alternative scanning technique. We start by formulating the probability of hit P_h, as the probability for the target object to be within the classifier detection range, with respect to the scanning grid interval (s_w, s_h), and to the translation tolerance (t_w, t_h) of the classifier C_{object}, (see Fig. 1a).

$$P_h \approx \frac{t_w t_h}{s_w s_h} \tag{1}$$

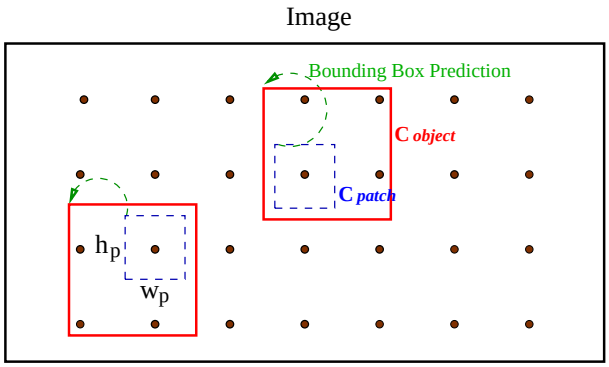
As an example, lets assume that the object present in the image is of the same size as the classifier is trained with, and if $t_w = t_h = 3$ and $s_w = s_h = 6$ then the probability of getting a hit P_h is 0.25, which is very low. As we decrease s_w and s_h (a finer search), P_h increases, while scanning speed decreases (slower). Our goal is to increase P_h without decreasing too much of the scanning speed (make it faster), and how we achieve this is described in Section 4.

3. BASELINE FACE CLASSIFIER

Many different classifiers and features are available for face detection task. We choose Modified Census Transform (MCT) features as it has been shown to be robust to lighting



(a) Standard scanning technique



(b) Our proposed scanning framework

Fig. 1. Standard scanning technique vs our proposed scanning framework. The dots represent the scanning grid with interval (s_w, s_h), target object size (o_w, o_h), translation tolerance (t_w, t_h) of target object classifier C_{object}, target patch size (u_w, u_h), and target patch classifier C_{patch}. The classifier C_{patch} predicts the bounding box for C_{object} in our approach.

variations and does not require any preprocessing [5]. A face/non-face classifier C_{face} is built using boosted MCT features as described in [5]. A single stage classifier is given by

$$H_s(I) = \sum_{k=1}^{K_s} w_k h_k(I) \tag{2}$$

where I is the input image, s represents the stage number, w_k is the weight associated with the weak classifier $h_k(I)$, and K_s is the number of features in each stage. The weak classifier $h_k(I)$ in this case is parameterized by a location and a look up table (see [5] for more details).

For building our baseline face classifier we obtain approximately 35,000 cropped face images (19x19) from standard face database (BANCA, Purdue, and XM2VTS). A subset of 15,000 face images are used for training, 10,000 are used for validation and the rest 10,000 are used for testing. We use the non-face test dataset from [6]. A cascade of 5 stages is trained and for each stage a threshold is estimated on validation dataset by fixing the detection rate. The non-face samples for each stage are collected from many images containing no face using bootstrapping technique. The final baseline face classifier C_{face} has a detection rate of 99% with false positive rate of 0.02%.

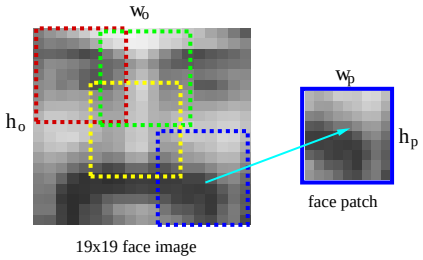


Fig. 2. Example of some overlapping face patches. All patches lie within the face region.

4. THE PROPOSED APPROACH

The approach described in this section tries to increase the probability of hit by using a patch classifier which identifies a part of face and infers the bounding box location (see Fig. 1b). If the bounding box estimation is good enough, we can achieve better chances to detect a face with larger grid spacing.

4.1. Probability of hit with our approach

In this subsection we explain how our method increases the probability of hit. Assuming that we have a classifier C_{patch} that predicts the patch location with prediction rate d_{patch} within the translation tolerance (t_w, t_h) of the classifier C_{object}, then the probability of hit can be approximately given by:

$$P_h \approx d_{patch} p_i \tag{3}$$

where $p_i = \frac{(o_w - u_w + 1)(o_h - u_h + 1)}{s_w s_h}$, (u_w, u_h) is the patch width and height, and (o_w, o_h) is the object width and height, with constraints $u_w < o_w$ and $u_h < o_h$ (see Fig. 2). For $u_w = u_h = 14$, $o_w = o_h = 19$, $s_w = s_h = 6$, and $d_{patch} = 0.8$ (this value is taken from our experiment results), we get $P_h = 0.8$, which is approximately 55% greater than standard scanning approach. The smaller the patch size is, the more the spacing between the grid can be, for a increase in scanning speed. Unfortunately, it also increases the number of classifiers that needs to be evaluated. Our goal is to build a patch classifier with high performance (both in speed and accuracy of estimation).

4.2. Face patch classifier

We represent a set of patches by $\{\mathcal{P}_i = (\mathcal{U}_i, c_i)\}$, where $\mathcal{U} \in \mathbb{R}^{u_w \times u_h}$ is the appearance and $c = \{1, ..., N\}$ is class label

of the patch. We have $N = (o_w - u_w + 1) \times (o_h - u_h + 1)$ possible overlapping patches. The goal here is to build a classifier which estimates the class label for a new patch. We use similar approach as described in [7] to build class conditional probabilities of binary features (Ferns) and at run-time use these probabilities to select the pattern with highest likelihood. Ferns are considered over SIFT features [8] as it is shown in [7] that it performs better and has less computation time. We propose here a new binary feature, referred to as μ-Ferns, as a simple comparison of a pixel with the average value of pixels in patch $\mathcal{U}$, where as Ferns compare two pixels at two pixels at random locations. The binary feature f_k is defined as

$$f_k = \begin{cases} 1 & \text{if } \mathcal{U}(x_k, y_k) \leq avg(\mathcal{U}) \\ 0 & \text{otherwise} \end{cases}$$

where (x_k, y_k) is the pixel location within patch $\mathcal{U}$, $k = 1, ..., K$, and K is the total number of binary features. Given a set of features $f_1, f_2, ..., f_K$ the idea is to find the best class c such that

$$\hat{c} = \arg\max_j P(c_j | f_1, f_2, ..., f_K) \quad (4)$$

Using Bayes' Formula, assuming uniform prior $P(c_j)$ and independence between features, the problem is reduced to:

$$\hat{c} = \arg\max_j \Pi_k P(f_k | c_j) \quad (5)$$

To obtain the probability $P(f_k|c_j)$, we just count the number of times the feature f_k takes the value 1 and 0. It is then normalized by dividing by the number of training examples.

5. EXPERIMENT EVALUATION

We evaluate the performance of the face patch classifier and then use this classifier with our proposed scanning framework. The detection rate and scanning speed are evaluated with respect to the scanning grid interval.

5.1. Evaluation of face patch estimation

We compare the performance of our proposed feature μ-Ferns with Ferns for patch estimation. We use the same training and test dataset as described in Section 3 for this evaluation. We follow the same procedure as described in [7] to train Ferns for a patch. Since the pixel pairs in Ferns are selected randomly, the performance at each run varies. Therefore we run many trials and keep the one with best performance. We have considered the location (x_k, y_k) to be on a uniform grid for μ-Ferns. To make a fair comparison we use the same number of binary features for both the approaches. Given a test patch, we use Equation 5 to estimate the best class label c. Each class label c has an associated (x_c, y_c) location within the face region. We consider (x_c, y_c) to be the top left corner of the patch in the face region. Since we want to measure how close the estimated patch location is to the true patch location, we use squared L_2 norm to evaluate the estimation error:

$$\lambda = (\hat{x_c} - x_c)^2 + (\hat{y_c} - y_c)^2 \quad (6)$$

where $(\hat{x_c}, \hat{y_c})$ and (x_c, y_c) are the estimated and true patch location. We define $p(\lambda)$ as the number of test patches that have estimation error of λ, and the cumulative distribution of estimation error as $c(\lambda) = \sum_{j=0}^{\lambda} p(j)$. Fig. 3 shows the cumulative distribution of estimation error for μ-Ferns and Ferns for square patch sizes of 14, 13 and 12. From Fig. 3, we see that μ-Ferns perform slightly better than Ferns. The best patch prediction is obtained for the patch size of 14 for both features. Ideally we would like to have a smaller patch size so that the grid spacing could be increased to speed up the search, but we see that the accuracy of estimation drops as the number of classifier grows. There is a trade off between the grid spacing and the patch size. We select patch size of 14 for our proposed scanning framework, since it achieves good detection rate with less computation time compared to other smaller patch sizes.

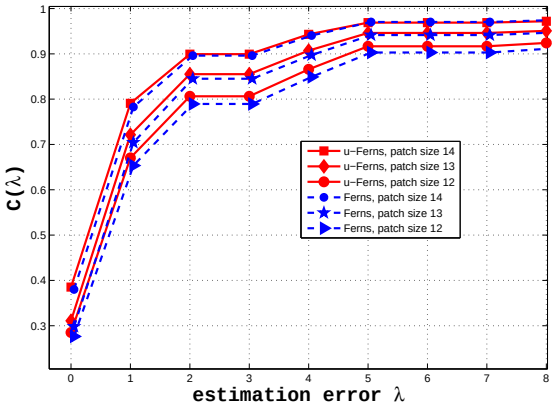


Fig. 3. Cumulative distribution of patch estimation error λ for patch sizes of 14,13 and 12, for μ-Ferns and Ferns.

5.2. Evaluation of proposed scanning framework

We now evaluate the performance of standard scanning technique and our proposed scanning approach. For this task we take CMU+MIT [9] and Fleuret [10] face databases, with a total of 375 images and 1085 faces of various size. We use a pyramid based scanning approach to detect faces at different scales. The scaling parameter is set to 1.2. Multiple detections are merged by averaging the detection within a certain radius which is a function of scale. The estimated eye coordinate of merged detection are compared with ground truth eye coordinates using Jesorsky measure [11], which is set to 0.3 for all our experiments. We obtain for each parameter

(patch size and grid spacing), the number of correct detection and time taken to scan 375 images. Fig. 4 shows the performance of both scanning techniques with respect to grid spacing. We can see clearly that we obtain higher detection rate for larger grid spacing. We also plot the average time taken to scan an image with respect to detection rate in Fig. 5. We achieve roughly 15-30% improvement in detection rate or speed when using the bounding box estimation for scanning. We also notice that when the grid spacing gets smaller and smaller, μ-Ferns are faster than Ferns.

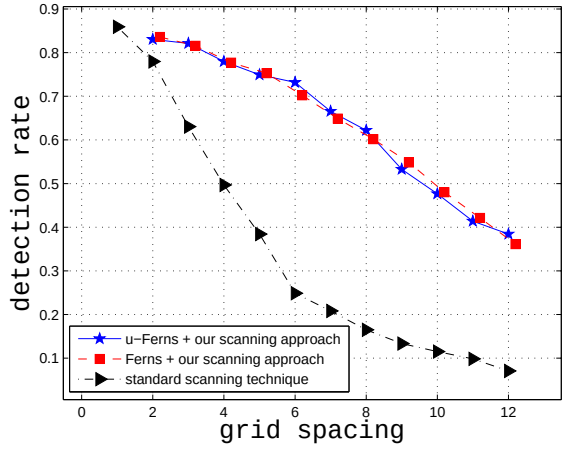


Fig. 4. Comparison of our proposed scanning approach to standard scanning approach with respect to grid spacing.

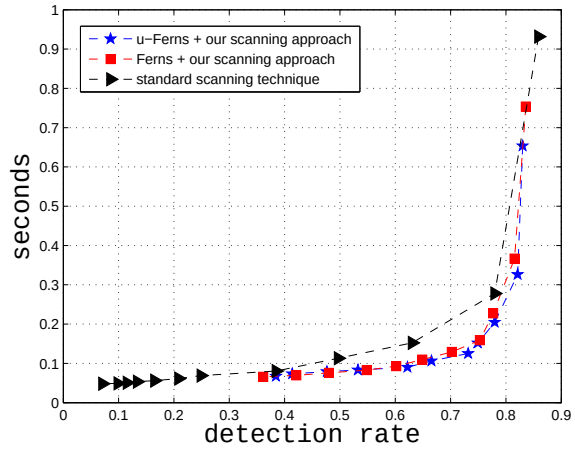


Fig. 5. Average time taken to scan an image in seconds vs detection rate.

6. CONCLUSION AND FUTURE WORK

In this paper we proposed an alternative scanning strategy to speed up the scanning process while maintaining the detection rate. We also proposed a new feature μ-Fern which is comparable or better than Ferns for our task. For our future work, we would like to investigate if any further improvements in speed can be achieved. One of the immediate extension of our approach is to predict the scale or rotation or different views of an object. The other extension would be to detect interest points first and use the bounding box prediction only at those locations.

7. ACKNOWLEDGMENT

The authors would like to thank the Swiss National Science Foundation, projects MultiModal Interaction and MultiMedia Data Mining (MULTI, 200020-122062) and Interactive Multimodal Information Management (IM2, 51NF40-111401) and the FP7 European MOBIO project (IST-214324) for their financial support.

8. REFERENCES

[1] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 511–518.

[3] Raphaël Féraud, Olivier Bernier, Jean-Emmanuel Viallet, and Michel Collobert, "A fast and accurate face detector based on neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, 2001.

[4] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 1997, p. 130.

[5] B. Froba and A. Ernst, "Face detection with the modified census transform," in *IEEE International Conference on Automatic Face and Gesture Recognition*, May 2004, pp. 91–96.

[6] "CBCL Face Database 1," in *MIT Center For Biological and Computation Learning*, http://www.ai.mit.edu/projects/cbcl.

[7] M. Özuysal, P. Fua, and V. Lepetit, "Fast keypoint recognition in ten lines of code," in *Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007, pp. 1–8.

[8] David G. Lowe, "Distinctive image features from scale-invariant keypoints," vol. 60, no. 2, pp. 91–110, 2004.

[9] CMU Face Group, "Frontal and profile face databases," 2009.

[10] F. Fleuret, "Fast binary feature selection with conditional mutual information," in *Journal of Machine Learning Research (JMLR)*, 2004, vol. 5, pp. 1531–1555.

[11] O. Jesorsky, K. Kirchberg, and R. Frischholz, "Robust face detection using the Hausdorff distance," in *3rd International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Halmstad, Sweden, 2001, pp. 90–95.