OPTIMIZATION OF FACE DETECTION WITH ADABOOST CASCADE ALGORITHM

Krzysztof Kucharski

Institute of Radioelectronics, Warsaw University of Technology
K.Z.Kucharski@elka.pw.edu.pl

ABSTRACT

The variant of AdaBoost cascade algorithm is described together with the optimization steps that make possible to achieve the real-time face and eyes detection. The output screen shots of such a system made for automatic face recognition purposes illustrate the successful application of presented techniques.

1. INTRODUCTION

Face detection using AdaBoost cascade classifier was introduced by Viola and Jones in their seminal paper in December 2001 [2] and extended to detect multipose faces by Xiao, Li and Zhang [3]. Their really novel approach has shown how local contrast features found in specific positions of the object can be combined to create a strong face detector. AdaBoost is known from the late 1980s as a multiclassifier and a training procedure for a collection of weak classifiers, e.g. having the success rate about 0.5, to boost them by suitable voting process to very high level of performance [1]. Altough employing such a training scheme allows for obtaining an ideal classification rate on the training stage the output strong classifier has too many weak components to detect faces in real time. The AdaBoost cascade is the solution that overomes these difficulties.

In the following paper we present the optimization techniques that help to incorporate the AdaBoost cascade algorithm into the face and eyes detection system that would process various scales in real time.

2. ADABOOST CASCADE ALGORITHM

The Adaboost cascade algorithm for object detection may be described as nested three-level process. On the lowest level parameters of the best single weak classifier are found, that on the higher level is incorporated into the set of such weak classifier forming the strong classifier. The third and the highest level creates serial connections between strong classifiers to maintain the trade-off between crucial detection performance measures, false acceptance rate and false rejection rate. The resulting final structure is called the AdaBoost cascade.

2.1. The weak classifier

For each object window o, the weak classifier elaborates a decision $\delta_\omega(o) \in -1, +1$ on the basis of membership of the object o to one of two classes labeled by -1 (a negative) and +1 (a positive) (Figure 1).

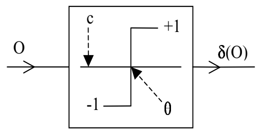

Fig. 1. Diagram of the weak classifier

In face and eyes detection region contrasts $c(R)$ are used as weak classifiers, where R is the sub-window of the image window o:

$$c(R) = \sum_{(x,y)\in R^+} o(x,y) - \sum_{(x,y)\in R^-} o(x,y)$$

The regions used to evaluate contrast in our implementation of AdaBoost cascade, which were taken directly from [2], are presented in Figure 2. The positive sub-region R^+ is drawn in white whereas the negative sub-region R^- is drawn in black.

region	type A	type B	type C	type D
width	2a	a	3a	2a
height	b	2b	b	2b

Fig. 2. Types of regions used to determine the weak classifier

During the training process the optimal choice of threshold θ and parameter sequence (x, y, a, b, t) is made, where a pair (x, y) is the anchor point of the region defined by the type $t \in \{A, B, C, D\}$ and size a and b in the object window o. The training is performed with respect to two labelled datasets consisting of P positives and N negatives respectively. The examples referring to the case of face detection are illustrated in Figure 3.

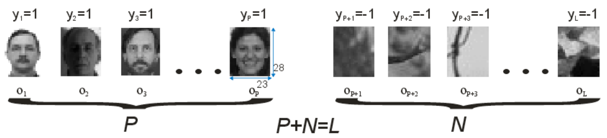

Fig. 3. Positive examples (faces) on the left and negative examples (non-faces) on the right used in the process of the weak classifier training

2.2. The strong classifier

For each object window o, the strong classifier conducts a procedure of weak classifier weighted voting where each weight is called the cost of a decision $\gamma_t(o)$ of the corresponding weak classifier i.e. the error it causes when applied to the training set (Figure 4).

In our version of the AdaBoost algorithm [4] it is assumed that the cost $\gamma_t = \beta_t$ of the negative decision $\delta(o) = -1$ is k_t times greater than the cost $\gamma_t = \alpha_t$ of the positive decision $\delta(o) = 1$.

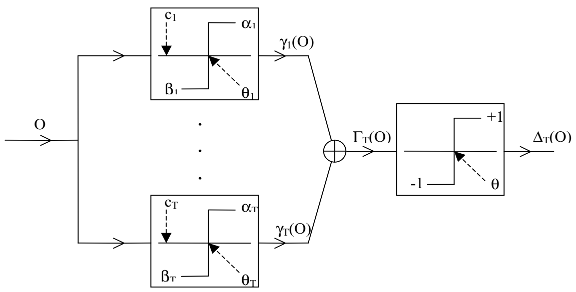

Fig. 4. Diagram of the strong classifier

The fact that the weak classifiers within the strong classifier are different follows from the main property of the AdaBoost theory, namely the step of changing the training examples weights after every single weak classifier was found. The weights are uniformly initialized i.e. $w_{i,1} = \frac{1}{L}, i = 1, \ldots, L$ and are modified due to the formula:

$$w_{i,t+1} = \frac{w_{i,t}e^{-\gamma_i(o_i)y_i}}{\sum_{i=1}^{L} w_{i,t}e^{-\gamma_i(o_i)y_i}}$$

The assumption about the relation $\gamma_t = k_t\alpha_t$ with $k_t = 3$ leads to the strong classifier that consists of eight weak classifiers and has a false rejection rate $fr = 0.0005$, false acceptance rate $fa = 0.48$ on the training set composed of $P = 4000$ faces and $N = 8000$ non-faces. The respective contract regions overlayed on a positive example are shown in Figure 5.

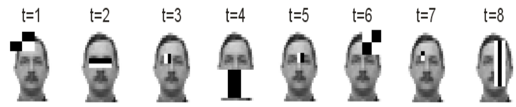

Fig. 5. Components of the strong classifier obtained in course of AdaBoost training process applied to the faces.

2.3. The cascade

A stage of the cascade is built by the AdaBoost algorithm i.e. the process of finding the strong classifier, in which the termination condition is satisfied when the false acceptance rate fa is below a threshold, while the false rejection rate fr does not exceed a given level. The specific values of these quantities have to be chosen depending on the properties of the object that is to be detected. For instance, the false acceptance threshold for both face and eyes detection was set to 0.5 whereas the false rejection level was set to 0.005 in case of face detection and 0.01 for eyes detection.

The termination condition for the cascade consisting of K stages is fulfilled if the total false acceptance rate $FA = fr_1 \times fr_2 \times \ldots \times fr_K$ drops below a threshold which we assumed to be 0.000005 for face detection and 0.001 for eyes detection. Value between 10^{-6} and 10^{-5} in face detection case is widely reported in literature, e.g. [2], but in case of the eyes it turned out that these data differ too little between various examples from the training set to reach such an extreme value of total false acceptance. Hence the conclusion follows that eyes detector cannot be used to detect eyes in the whole input image but only within the image of face found earlier with the face detector.

In face detection case, the output AdaBoost cascade structure we trained consists of 17 stages and 972 weak classifiers, subsequent stages having respectively 8, 11, 18, 19, 26, 38, 48, 48, 62, 68, 65, 67, 91, 84, 100, 112, 107 weak classifiers. In eyes detection case output structure consists of 8 stages and 596 weak classifiers, subsequent stages having respectively 8, 16, 26, 45, 87, 99, 142, 173 weak classifiers. The much more rapid growth of cascade stage size in eyes detector may be noticed, it confirms that there is less discriminatory information contained in eyes images. Also, about one of every 100 object windows can pass through first 6 stages of the cascade, therefore high numbers of weak classifiers in the further stages basically do not affect speed of the detection process.

3. THE DETECTOR OPTIMIZATION STEPS

The detection process by the AdaBoost cascade algorithm is simply a scanning the input high-resolution image, anchoring the object window in every single pixel of the image and classifying it to one of two classes i.e. object or non-object, using the AdaBoost cascade structure.

3.1. Normalization

Reliable design of any set of discriminative features requires a normalization of every object window processed during a detection. The goals of normalisation are, to get zero mean and common value of standard deviation for all object windows by a simple affine transformation in the luminance domain:

$$o'(x, y) = d_1 o(x, y) + d_2$$

The coefficients of normalisation for the window $O : d_1 = 255/\sigma_O,\ d_2 = -d_1\mu_O$, where σ_O is the standard deviation and μ_O is the mean of pixels within the object window O. Fast normalization is based on an observation that each object window need not be separately normalized. Instead, the values of σ_O and μ_O have to be calculated for every O and used to modify a region contrast $c(R)$ in a single additional operation. Because the calculation of σ_O and μ_O requires only a few additions given the integral and squared integral images, that are calculated once for the whole input image, the substantial acceleration is achieved. It can be shown [] the region contrasts $c(R')$ for a normalized object window are expressed as follows:

$c(R') = d_1 c(R)$ for regions of types A, B, D.
$c(R') = d_1 c(R) - d_2 ab$ for regions of types C

3.2. Scaling

Scaling the object window s_x times horizontally and s_y vertically changes the region contrast $c(R)$ and the contrast threshold θ $s_x s_y$ times:

$$c_{(s_x,s_y)}(R) = s_x s_y c(R),\ \theta_{(s_x,s_y)} = s_x s_y \theta$$

Applying only above update formula is sufficient to detect object at arbitrary scale because the rest of the AdaBoost detector remains unchanged. Still, the reasonably small number of scales has to be consider not to increase too much the computational complexity of the detector. On the other hand, the detector must ensure the localization of the object whose size varies in the consecutive frames of the video sequence.

Thus we assumed the scale factor to be $s_x = s_y = 1.25$ and obtained direct six different scales covering the object window size from the original one 23x28 up to 171x211. Indirectly, we obtained in this way also intermediate values, what will be explained in details in the next section.

3.3. Multiple alarms

In spite of a very low level of false acceptance rate, the detector has to be prepared for the situations where false alarms occur. Moreover, in the case of correct detection we can expect for sure the more that one alarm because more than one bounding box surrounds the object, e.g. a face or an eye pair. The way of tackling such a problem is presented in Figure 6.

The input image undergoes detection process in six scales numbered from 1 to 6, where the three largest ones correspond to the sought faces of size 76x94, 114x141 and 171x211. If a object window passes through 12 stages of the AdaBoost cascade detector it is assigned a weight 1, in Figure 6; after passing 13 stages it is assigned a weight 2 and so on; if it passes the whole cascade it is assigned a weight 6.

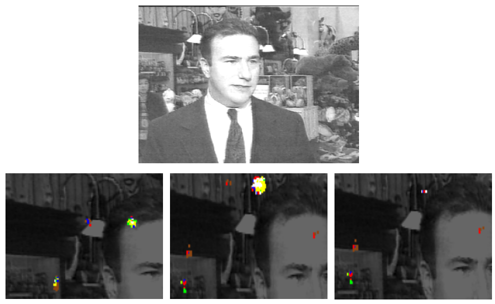

Fig. 6. The input image and the alarms in scale no. 4,5, and 6 ordered from the left respectively. The correspondence between weight values and the colours is as follows: brown(1), red(2), blue(3), green(4), yellow(5), white (6).

Next, every pixel of alarm has the level of alarm in its $3x3$ neighbourhood calculated, if it is below a specific threshold the pixel is rejected (compare Figure 7). The representatives of so-formed pixel clusters are then calculated and, finally, the representatives coming from various scales are merged. The process of merging face rectangles between two subsequent scales is started only if the field of the their intersection exceeds a half of the field of the smaller of the two rectangles. The process is illustrated in Figure 7. The size of the output face rectangle lies in between the sizes indicated by the scales participating in the merging process.

The eye pair detector starts its work on the face rectangles returned by the face detector. Every pixel becomes an alarm pixel if it successfully passes the whole AdaBoost cascade, i.e. 8 stages. Next, the alarm pixels are grouped in regions using the connected component analysis and the region having the greatest number of members is assumed to be a cluster of eye pair upper left corners. The whole process is illustrated in Figure 8.

The threshold used to exclude some alarm points dur-

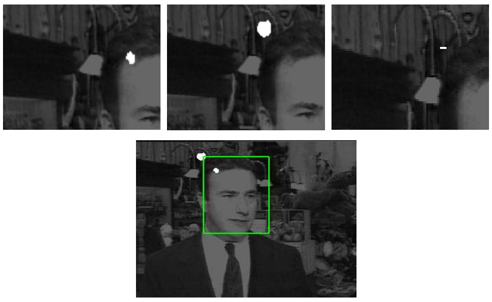

Fig. 7. The alarms after thresholding in scales no. 4, 5, 5 and the output face rectangle.

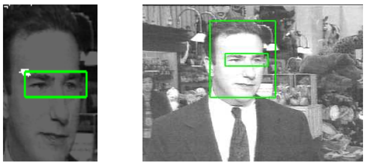

Fig. 8. The process of determining an eye pair position within the face and the final result of face and eyes detection.

ing the face detection is selected heuristically, basing on the fact how small faces the designed detector is to deal with. Actually, the faces in smaller scales have smaller number of alarms than larger faces.

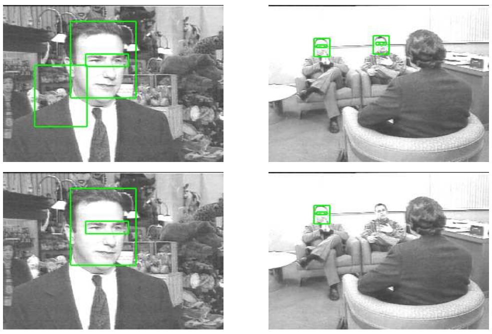

Fig. 9. Detection results in presence of smaller (upper row) and greater (bottom row) thresholds.

Therefore, lowering the threshold for small scale can help to detect small faces but, on the other hand can include also the false rectangles into the set of results. Example of such a situation is presented in Figure 9.

4. REAL-TIME FACE DETECTION SYSTEM

Both face and eye pair detector described in previous sections was used in real-time face detection system which goal is to recognize faces automatically. Real-time automatic face localization is a necessary step to gain robust recognition performance. Hence, the eye localization is used here to normalize input face of arbitrary size to the predefined size $46x56$ for further processing for recognition purposes. Because the face recognition hardly benefits from very poor resolution photos it was sufficient to detect faces only in four scales starting from the scale no. 3. The results placed in Figure 10 were obtained on the PC with Pentium IV 2.4GHz and 512 MB RAM. The empty frames i.e. frames without any face are processed in about 300 ms (3 fps), the frames containing from one to three faces are processed in about 450-600 ms (2fps).

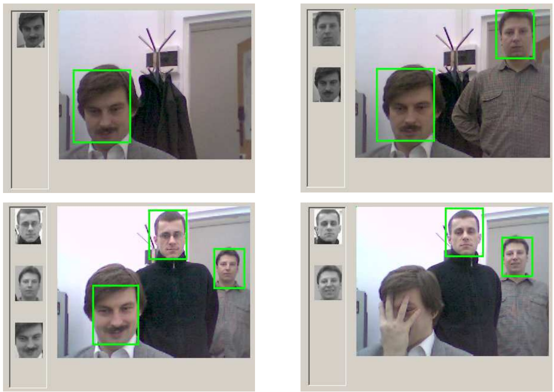

Fig. 10. Real time detection results in various scenarios. On the left images normalized to size 46x56 using the eye pair detector.

5. REFERENCES

[1] Freund Y, Schapire R.E.: A decision theoretic generalization of on-line learning and an application to boosting, Journal of Computer and Systems Sciences, 55(1):119-139, August 1997.

[2] Viola P., Jones M.: Rapid Object Detection using a Boosted Cascade of Simple Features, Computer Vision and Pattern Recognition, 2001.

[3] Xiao R., Li M.-J., Zhang H.-J.: Robust Multipose Face Detection in Images, IEEE Trans. on Circuits and Systems for Video Technology, January 2004.

[4] Skarbek W., Kucharski K., Image Object Localization by AdaBoost Classifier, ICIAR 2004, Porto, Portugal, Sept. 29 - Oct. 1, 2004.