1

The Architecture of the Face and Eyes Detection System Based on Cascade Classifiers

Andrzej Kasinski[1] and Adam Schmidt[2]

[1] Institute of Control and Information Engineering str. Piotrowo 3a 60-965 Poznan, Poland Andrzej.Kasinski@put.poznan.pl

[2] Institute of Control and Information Engineering str. Piotrowo 3a 60-965 Poznan, Poland Schmidt.Adam@gmail.com

Summary. The precise face and eyes detection is crucial in many Human-Machine Interface system. The important issue is the reliable object detection method. In this paper we present the architecture of a 3-stage face and eye detection system based on the Haar Cascade Classifiers. By applying the proposed system to the set of 10000 test images the 94% of the eyes were properly detected and precisely localized.

1.1 Introduction

Many Human-Machine Interface (HMI) tasks, such as face tracking, expression recognition and human person recognition to be efficient require the proper initialization. For example, face recognition (FR) techniques are sensitive to the accurate face alignment. Detecting only a face is often insufficient to achieve the desired final classification results. Information such as face in-plane rotation, scale and precise location can be obtained by localizing eyes on previously extracted faces.

As precise eyes location enables accurate alignment, one has to first design an efficient face and eye localization method in order to develop an automatic face recognition system.

Currently, the Haar Cascade Classifiers (HCC) are getting increasing attention. High detection ratio obtained with those computationally-efficient detectors suggests the possibility of using them in a reliable real-time HMI systems. Therefore, our goal was first to train the efficient face and eyes HCC detectors and then to combine them into a hierarchical system. Moreover, we wanted to improve the detection rates of the HCC by introducing the additional knowledge-based criteria.

In this paper we present a 3-stage hierarchical face and eye detection system based on the HCC. Firstly, we present the state of art in both face and eyes detection. Secondly, we present the idea of the HCC. Then we describe the architecture of our system. Finally, give the preliminary results and the achieved detection rates.

1.2 The state of art in face and eyes detection

A human face is a highly non-rigid 3D object whose image is susceptible to both pose and expression variations. This combined with variability of personal face features and possible structural disturbances (such as glasses, facial hair, make-up) makes face detection a challenging task. The poor performance of features detectors based only on human knowledge as well as of those based on simple template matching, points out to the necessity of involving machine learning-based approaches.

Huang et al. [1] applied the Polynomial Neural Network (PNN) to the task of detecting faces. PNN is a single-layer network taking polynomial expansion of pattern features as inputs. The feature pool was based on pixel intensity values, Sobel filter responses, and on directional gradient decomposition. The Principal Components Analysis (PCA) was then used to reduce feature's vector dimension. It was proved that the system based on gradient decomposition outperformed systems using simpler features.

The system proposed by Heisele et al. [2] consisted of three independent first-level Support-Vector Machine (SVM) detectors for finding potential eyes, nose and mouth regions. The second-level classifier checked if their relative position could correspond to that typical for the human face. Bileschi and Heisele [3] improved this method by training SVM not against a rich background but only against some other facial features. The authors claimed that component-based face detection system is more resistant to face pose changes than holistic detectors.

Viola and Jones [4] were first to introduce Haar Cascade Classifiers and to apply HCC to the task of face detection. The idea of using cascade of simple classifiers led to the creation of an accurate and computationally efficient detection system. Lienhart et al. [5] improved the HCC by enlarging the feature pool with the rotated Haar-like features. He also tested the influence of various weak classifiers and boosting algorithms on the performance of cascades.

Weak classifiers ensembles were also used by Meynet et al. [6]. They combined a simple HCC with another parallel weak classifiers ensemble. The HCC was used to discard easy to classify non-faces. The remaining windows have been tested with boosted classifiers based on the Anisotropic Gaussian Features; the final classification depending on votes of those classifiers.

Eyes detectors are usually applied to already localized face regions, which are fairly similar. Here the detectors must discriminate between eyes and other facial features.

Wang et al. [7] used homomorphic filtering to compensate for illumination variations. The binary template matching was then applied to preprocessed images in order to extract potential eyes. Candidate regions were verified with the SVM and the precise eyes' location was acquired with variance filters.

Many authors tried to use the HCC in the task of eyes detection. Wilson and Fernandez [8] used cascades trained against other features to extract eyes, a mouth and a nose from the face region. As the processing of a whole face led to many false positives (FP) they proposed the regionalized search approach. This explicitly means the use of the knowledge about a face structure i.e. looking for a left eye in an upper-left, for a right eye in an upper-right, a nose in a central and a mouth in a lower part of the face. This simple solution significantly reduced the FP ratio.

Feng et al.[9] used the HCC at the first stage of their detection system. In order to reduce the FP ratio the results have been verified with another boosted classifier, the one based on ordinal features rather than on Haar-like and trained with the algorithm similar to the AdaBoost.

1.3 The Haar Cascade Classifiers

The HCC emerged as a successful combination of three ideas. Instead of working directly on image function values, the detector uses an extensive set of features, which can be efficiently computed in a fixed time. This feature-based approach helps to reduce the in-class variability and increases variability between classes. Secondly, using a boosting algorithm allows for a concurrent selecting of a small subset of sufficient features and for the classifier training. Finally, creating a cascade structure of gradually more complex classifiers results in a fast and efficient detection scheme.

Haar-like features

According to Lienhart [5], Haar-like features can be calculated with the following equation:

$$feature = \sum_{i \in \{1 \ldots N\}} \omega_i \cdot RecSum(x, y, w, h) \tag{1.1}$$

Where *RecSum(x, y, w, h, ϕ)* is the sum of intensity values over any given upright or rotated rectangle enclosed in a detection window and x, y, w, h, ϕ stand for coordinates, dimensions and rotation of that rectangle (see Figure 1.1).

To reduce potentially infinite number of features, the following restrictions are applied:

- Pixel sums over only two rectangles are allowed (N=2)
- The weights are used to compensate the area difference of two rectangles and have opposite signs. Which means that $\omega_0 \cdot Area(r_0) = -\omega_1 \cdot Area(r_1)$, substituting $\omega_1 = 1$ one gets $\omega_0 = -Area(r_0)/Area(r_1)$

- The features should be similar to those used in early stages of human vision pathway, such as directional responses of Gabor filters.

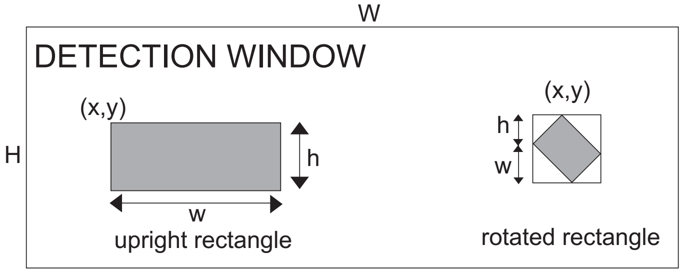

Fig. 1.1. Upright and rotated rectangles in the detection window

Those constraints leave 14 prototype features (see Figure 1.2), which can be scaled in both directions and placed in any part of the detection window. This allows to create an extensive feature pool. The features are computed as the proportion of pixel sums under black and white rectangles and scaled to compensate for the areas difference. It's worth mentioning, that line features can also be viewed as a combination of two rectangles: one of them containing both black and white, but the second only a black area.

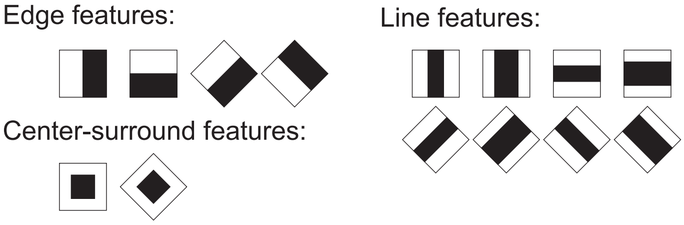

Fig. 1.2. Prototypes of Haar-like features

To efficiently evaluate features, two novel image representations are introduced. The Summed Area Table (SAT(x, y))[4] is used to calculate features based on upright rectangles. Each entry of the table is defined as the sum of pixel intensities over the upright rectangle extending from (0, 0) to (x, y) and is being filled according to the formula:

$$SAT(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y') \tag{1.2}$$

Once filled, the SAT enables computing pixel sum over any upright rectangle with only four look-ups:

$$\begin{aligned}
&RecSum(x,y,w,h,0) = SAT(x-1,y-1)+\\
&\quad + SAT(x+w-1,y+h-1) - SAT(x+w-1,y-1) -\\
&\quad - SAT(x-1,y+h-1)
\end{aligned} \tag{1.3}$$

Rotated features are computed using another auxiliary representation called the Rotated Summed Area Table [5]. Each entry is filled with the following value:

$$RSAT(x,y) = \sum_{|x-x'|\leq y-y', y'\leq y} I(x',y') \tag{1.4}$$

Pixel sum of any rotated rectangle can be computed according to:

$$\begin{aligned}
&RecSum(x,y,w,h,45) = RSAT(x-h+w,y+w+h-1)+\\
&\quad + RSAT(x,y-1) - RSAT(h-x,y+h-1) -\\
&\quad - RSAT(x+w-1,y+w-1)
\end{aligned} \tag{1.5}$$

Classifiers cascade

In the most of cases the detected object occupies only a small part of the image. Thus it's better to discard non-object regions quickly and focus only on those which are relevant, than to examine every window thoroughly. The cascade structure allows for such an approach. It consists of the N stages i.e. of serially connected classifiers distinguishing between the detected object and the background. Every stage is trained to achieve TP ratio p close to 1 with the FP ratio f kept usually 0.5. The positively classified windows are passed to the subsequent stage; the others are excluded from the further processing.

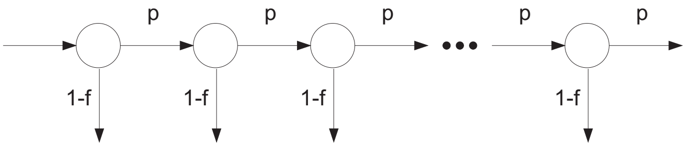

Fig. 1.3. Structure of the cascade detector

Due to the serial nature, the overall detection ratios are exponential function of single stage efficiencies:

$$TP_{cas} = \prod_{i=1}^{i\leq N} p_i \approx p^N \tag{1.6}$$

$$FP_{cas} = \prod_{i=1}^{i\leq N} f_i \approx f^N \tag{1.7}$$

The adequate selection of p, f and N results in a detector achieving a high TP ratio (slightly less than 100%) and a low FP ratio at the same time. The stages are consecutively trained to achieve the desired detection rates. The whole sets of positive and negative samples are presented only to the first stage classifier. The others are trained only with subsets which have passed previous stages. In that way classifiers at succesive stages are faced with more difficult tasks and have to discover subtler differences to keep up the desired p and f ratios.

The single stage classifier

Having such extensive features pool one needs to find a way to select a minimal subset guaranteeing the desired detection rate. Boosting is a machine learning concept combining performance of many 'weak classifiers' (where 'weak' means only slightly better than a random choice) into a single powerful ensemble called 'strong classifier'.

In the HCC simple CARTs (Classification And Regression Tree) are used as weak classifiers. Their size is restricted only to several splits. In the simplest case (single-split CARTs called "stumps") they rely on a single feature only. Using slightly more complex weak classifiers (e.g. 4-splits CARTs) slows down the training but allows for preserving some relations between features in a weak classifier. Even those more complex classifiers could not be sufficient to achieve the desired detection rates. To assemble weak classifiers into a strong one the boosting algorithm called AdaBoost[10] is used. In [5] Lienhart with colleagues proved that by using the version called the Gentle AdaBoost one obtains a detector having lower FP ratio than those detectors trained by using other AdaBoost versions.

Gentle Adaboost algorithm specification according to [10]:

```
1. Given $N$ examples $(x_1, y_1), ..., (x_N, y_N)$ where $x_i \in R^k, y_i \in (-1, 1)$
2. Start with weights $w_i = 1/N, i = 1, ..., N$
3. Repeat until $p$ and $f$ are achieved, for m = 1,...,M
   a. Fit the regression function (the CART) $f_m(x)$ by the weighted least-squares of $y_i$ to $x_i$ having weights $w_i$
   b. Set $w_i = w_i \cdot \exp(-y_i \cdot f_m(x_i))$
4. Output the classifier: $sign[\sum_{m=1}^{M} f_m(x)]$
```

1.4 The system's architecture

Our detection system consists of the three stages. At the first one, the HCC face detector is applied to the whole image. As the neighboring positive responses of HCC are merged into a single detection result, it is possible to fine-tune the detection ratios by relaxing the constraint on the minimum number of neighbors (*Nb*). The detected candidate regions are futher processed independently.

At the second stage, the left and right eye HCC detectors are used on previously found face regions and their results are stored in the two lists. As with the face detector the constraint on the minimum number of merged neighbors can be set. Moreover, instead of searching for eyes over the whole face,the regionalized search can be used. This means, applying the left eye detector the rightmost and the right eye detector to the leftmost 60% of the upper half of the face.

The third stage is a simple knowledge-based rule of combining left and right eye detections into valid eye-pairs. For each left and right eye combination in a given face rectangle an in-plane rotation ϕ is calculated. Eye-pairs with $\phi > 20$ are discared, as too unlikely to belong to a upright view of the face.

1.5 The results and conclusions

Our HCCs were trained with the Gentle AdaBoost by using 4-split CART as a weak classifier and setting the required TP ratio of each stage for 0.999.

The positive training set for the face HCC consisted of 2500 face pictures from our base. The negative set was created by randomly gathering 3500 images not containing any faces. Eyes detectors were trained with positive sets of 2500 left or right eye images from our base. Negative sets consisted of face images with the appriopriate eye being hidden.

The eye localization error measure was the same as the one used by Campadelli [11].

$$error = \frac{\max(\|C_l - C_{lGT}\|, \|C_r - C_{rGT}\|)}{\|C_{lGT} - C_{rGT}\|} \tag{1.8}$$

Where C_l stands for the center of the left eye found, C_r stands for the center of the right eye found, C_{lGT} and C_{rGT} stand for the centers of ground truth eyes.

Detections with error less than 0.1 were treated as TPs, the others were counted as FPs. Pictures without any positive eyes detection were considered to be false negatives (FNs).

While no constraints to the minimum number of neighbors were set, the proposed system achieved the TP ratio of 94% and FP ratio of 13%. Constraining solely the eyes detector (*Nb=3*) resulted in the TP ratio of 88% with less than 1% FP. The average processing time on a PC with Intel Celeron 2,8 GHz processor and 512 MB RAM was 321 ms.

Our detection system was proved to be efficient both w.r.t detection rates and computation costs. It turned out to be resistant to pose variations and to structural disturbances.

Future work will focus on using this solution in a complete face recognition system. The influence of the training sets used, of the weak classifiers applied

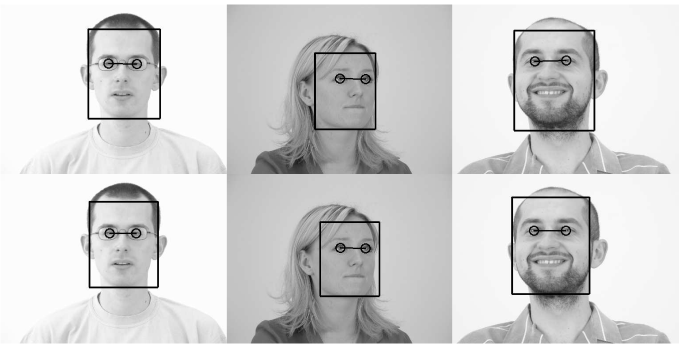

Fig. 1.4. Exemplary results: upper row - manually marked face and eyes, lower row - automatically detected face and eyes

and of the each stage's desired detection rates settings on the overall detection ratios will be investigated. Additionaly, our detectors performance will be compared with the performance of custom HCCs trained by other authors.

References

1. Huang L-L, Shimizu A, Hagihara Y and Kobatake H (2003) Gradient feature extraction for classification-based face detection. Pattern Recognition 36:2501:2511
2. Heisele B, Serre T, Pontil M and Poggio T (2001) Component-based Face Detection. CVPR 01:657–662
3. Bileschi S, Heisele B (2002) Advances in Component-Based Face Detection. In: Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines. Springer-Verlag, London, UK
4. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of CVPR 1:511–518
5. Lienhart R, Kuranov A, Pisarevsky V (2002) Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. Technical report, Microprocessor Research Lab, Intel Labs
6. Meynet J, Popovici V, Thiran J (2005) Face Detection with Mixtures of Boosted Discriminant Features. Technical report, EPFL
7. Wang Q, Yang J (2006) Eye Detection in Facial Images with Unconstrained Background. Journal of Pattern Recognition Research, 1:55–62
8. Wilson P, Fernandez J (2006) Facial feature detection using Haar classifiers. J. Comput. Small Coll. 21:127–133
9. Feng X, Wang Y, Li B (2006) A fast eye location method using ordinal features. In: Proceedings of the 2006 ACM SIGCHI
10. Freund Y, Schapire R (1996) Experiments with a New Boosting Algorithm. In: Proc. of International Conference on Machine Learning
11. Campadelli P, Lanzarotti R, Lipori G (2006 (preprint)) Eye localization: a survey. In: The fundamentals of verbal and non verbal communication and the biometrical issues. NATO Science Series