An Study on Ear Detection and its Applications to Face Detection

Modesto Castrillón-Santana, Javier Lorenzo-Navarro, and Daniel Hernández-Sosa*

SIANI
Campus de Tafira
Universidad de Las Palmas de Gran Canaria
35017 - Spain

Abstract. OpenCV includes different object detectors based on the Viola-Jones framework. Most of them are specialized to deal with the frontal face pattern and its inner elements: eyes, nose, and mouth. In this paper, we focus on the ear pattern detection, particularly when a head profile or almost profile view is present in the image. We aim at creating real-time ear detectors based on the general object detection framework provided with OpenCV. After training classifiers to detect left ears, right ears, and ears in general, the performance achieved is valid to be used to feed not only a head pose estimation system but also other applications such as those based on ear biometrics.

Key words: face detection, facial feature detection, ear detection, Viola-Jones

1 Introduction

Among the wide literature on the face detection problem, the well known Viola-Jones face detector [21] has received lots of attention. This interest is justified not only thanks to its remarkable performance, but also due to its availability to a large community via the OpenCV library [13, 14].

However, Viola and Jones [21] designed a general object detection framework. The approach is therefore suitable to be applied not only to the face pattern. Indeed, several researchers have already trained classifiers to detect different targets, and distributed to the community in the current OpenCV release [7, 18]. Among those available classification cascades, it is observed that most of them are focused on the frontal face and its inner facial features (eyes, mouth and nose). There are two exceptions within the available classifiers in OpenCV, but also related with human detection, these are the profile face detector [2], and the head and shoulders [12] detector. Both are particularly less reliable than those designed for the frontal pose [3].

* Work partially funded by the Spanish Ministry of Science and Innovation funds (TIN 2008-06068).

In this paper, we are interested in putting in practice the Viola-Jones framework to automatically detect the ear pattern in images. It is evident that the ear pattern, as present in Figure 1, would be visible only if it is not occluded and the head is not frontal, i.e. the head presents a profile (or almost) pose.

Automatic ear detection is indeed an useful ability in the human machine interaction scenario. Its detection can for instance be applied in conjunction with a face detector to better fit a 3D head model onto an individual, achieving better pose estimation [16, 20]. Additionally, the ear pattern has been used in biometrics by its own or as supplementary cue [10]. Therefore, its live and automatic detection would be of interest to multimodal recognition based on ear and face images [4, 9].

To reach this objective we have adopted, as mentioned above, the Viola-Jones framework. Indeed the adaboost approach has already been used to design ear detectors [1, 8]. The main difference with both works is that we are employing standard tools integrated with OpenCV to create the cascade. Our final aim is to make the classifiers available, including them in OpenCV, to serve as baseline. We consider that this is an advantage as previous researchers have provided their results but not released their detectors to the community. Abaza et al. are also concerned about reducing the training time. We will see later that the processing time is not so high using the standard OpenCV commands. Additionally, we present a larger experimental setup, in comparison to [8] and less restricted, in terms of controlled imagery, if compared to [1].

Section 2 summarizes the Viola-Jones detection framework. Section 3 describes the data used for the experimental setup. Results and conclusions are presented in sections 4 and 5, respectively.

2 Viola-Jones general object detection framework

Creating a detector with the Viola-Jones framework requires: 1) a large training set (at least some thousands) of roughly aligned images of the object to detect or target (positive samples), and 2) another even larger set of images not containing the target (negative samples). This setup is a tedious, slow and costly phase, that has been summarized in different brief tutorials, e.g. [19].

Both images sets are used to train a boosted cascade of weak and simple classifiers. The main idea behind this framework is to apply less effort in processing the image. Each weak classifier is fast and has the ability to provide a high detection ratio, with a small true reject ratio, i.e. it is able to detect the target most of the time. However, a weak classifier is not able to reject all the patterns without interest. Indeed, it would be enough if it is able to reject half of them. In terms of execution time, the resulting cascade of classifiers would be much faster than a strong classifier with similar detection rates.

Each weak classifier uses a set of Haar-like features, acting as a filter chain. Only those image regions that manage to pass through all the stages of the detector are considered as containing the target. For each stage in the cascade, see Figure 2, a separate subclassifier is trained to detect almost all target objects

(a) (b)

Fig. 1. FERET dataset [17] ear annotation examples. If mostly visible, the ear pattern has been annotated manually rotated in faces rotated along the vertical axis (out-of-plane rotation.)

while rejecting a certain fraction of those non-object patterns that have been incorrectly accepted by previous stage classifiers.

Theoretically for a cascade of K independent weak classifiers, the resulting detection rate, D, and the false positive rate, F, of the cascade are given by the combination of each single stage classifier rates:

$$D = \prod_{i=1}^{K} d_i \qquad\qquad F = \prod_{i=1}^{K} f_i \tag{1}$$

Each stage classifier is selected considering a combination of features which are computed on the integral image, see Figure 3a-b. These features are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses, see Figure 3c. The implementation [15] integrated in OpenCV [7] extended the original feature set [21].

With this approach, given a 20 stage detector designed for refusing at each stage 50% of the non-object patterns (target false positive rate) while falsely eliminating only 0.1% of the object patterns (target detection rate), its expected overall detection rate is $0.999^{20} \approx 0.98$ with a false positive rate of $0.5^{20} \approx 0.9 * 10^{-6}$. This schema allows a high image processing rate, due to the fact that background regions of the image are quickly discarded, while spending more time on promising object-like regions. Thus, the detector designer chooses the desired number of stages, the target false positive rate and the target detection rate per stage, achieving a trade-off between accuracy and speed for the resulting classifier.

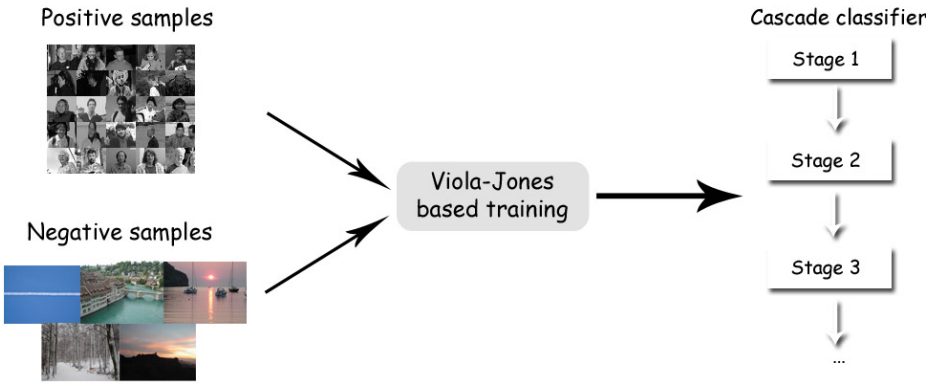


Fig. 2. Typical training procedure for a Viola-Jones' based classifier. Each classifier stage is obtained using positive and negative samples accepted by the previous stage. Adapted from [3].

Given an input image, the resulting classifier will report the presence and location, in terms of rectangular container, of the object(s) of interest in the image.

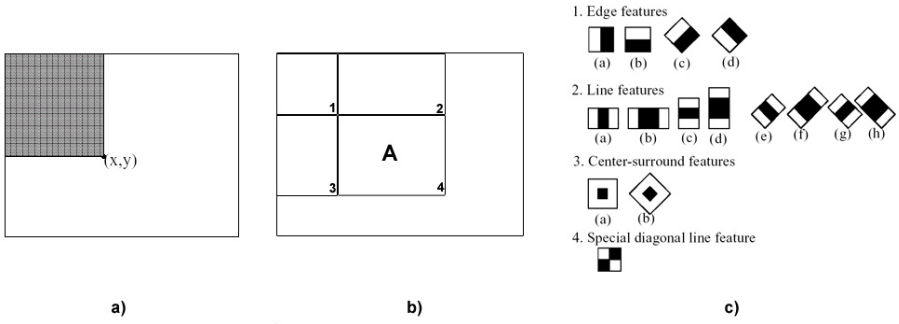


Fig. 3. a) The Integral Image stores integrals over subregions of the image. b) The sum of pixel values in A is $(x4, y4) - (x2, y2) - (x3, y3) + (x1, y1)$ [5] . c) Features prototypes considered in the implementation integrated in OpenCV [13, 15].

The availability of different tutorials, e.g. [19], guides OpenCV users to collect, annotate and structure the data before building the different classifier training. To test their performance, they must later be tested with an independent set of images.

3 Datasets

The imagery used to train and evaluate the ear detection performance is the FERET dataset [17]. Even when this dataset is mainly known in the face recognition literature, it is used in this paper to evaluate the ear detection performance. The dataset contains two subsets that we refer below as FERETCD1 and FERETCD2.

The dataset includes frontal, profile and inbetween faces. For our purpose, we have considered just the profile or almost profile images contained in the thumbnails folder of both subsets. Each ear present in those images, has been manually annotated defining a container with four points as seen in Figure 1.

As the reader can observe in Figure 1, some annotated ears will correspond to the left and some to the right ear. We have created three different classifiers to detect ears (left ear, right ear, just ear). For that purpose, we have flipped all the annotated images in FERETCD1 to build a larger training set suitable to train the different patterns. Those annotated images contained in FERETCD1 constitute the set of positive samples. The number of annotated images in both sets is reflected in Table 1. The set of negative samples (also been flipped to avoid any bias) is composed mainly of large wallpaper images that do not contain the target pattern. These datasets are used to train the different ear detectors.

The FERETCD2 subset is used for evaluation. The resulting classifiers provide detection results, that must be compared with the annotation data to determine the classifier goodness. The criterion adopted to consider an ear detection, e_d, as true detection, will observe the overlap with the annotated container, e_a, and the distance between both containers:

$$correct\ detection = overlap\ OR\ close \tag{2}$$

where *overlap* is

$$overlap = \begin{cases} true \text{ if } \frac{a_a \bigcap a_d}{a_a} > 0.5 \\ false \text{ otherwise} \end{cases} \tag{3}$$

being a_a and a_d the area of the annotated and detected container. And *close* is defined as

$$close = \begin{cases} true \text{ if } \ dist(e_d, e_a) < 0.25 \times e_{a-width} \\ false \text{ otherwise} \end{cases} \tag{4}$$

where *dist* refers to the distance between both container centers.

Table 1. Total number of images contained in each subset, and the number of ears annotated in each set (observe that not all the images present a profile or almost profile pose). Hidden ears have not been annotated, but some partially hidden ears have been approximately estimated.

Set	Total number of images	Annotated ears
FERETCD1	5033	2798
FERETCD2	4394	1947

4 Experimental results

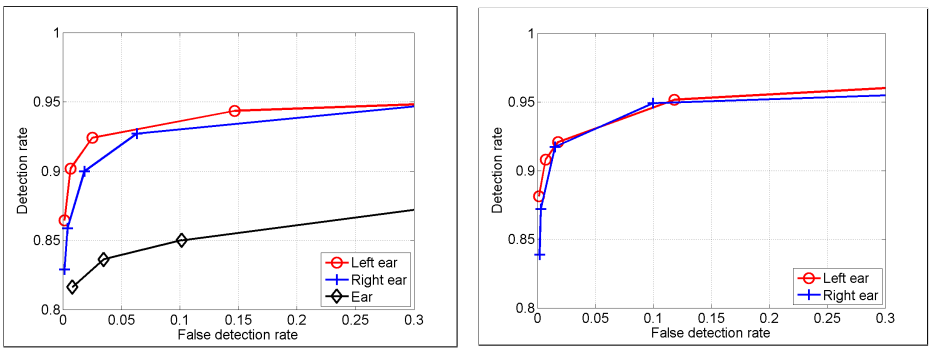
4.1 Ear detection performance

As mentioned above, we have used the FERETCD1 and the negative images sets to train the different target classifiers, while the FERETCD2 set has been used to evaluate both classifiers.

Giving some training details, on one side, the number of positive samples used to create each classifier based on the OpenCV implementation was 3000 (6000 for the ear detector). The reader can observe that this number is slightly larger than the number of positive samples indicated in Table 1. Indeed, the utility integrated in OpenCV to compile the file of positive samples creates additional training samples making use of reduced affine transformations. On the other side, 10000 was the number fixed as negative samples. The rest of the training parameters employed were mainly default values, excepting the pattern size selected, 12×20, and the tag indicating that the target pattern is not symmetric.

The training time to compute each 20 stages classifiers, using a 2.66Ghz processor, was around 30 hours for the left and right ear detectors, and 40 hours for the general ear detector.

The detection results achieved for the FERETCD2 set are presented in Figure 4a. For each classifier, its receiver operating characteristic (ROC) curve was computed applying first the 20 stages of each classifier, and four variants reducing its number of stages (18, 16, 14 an 12 respectively). Theoretically, this action must increase both correct, D, and false, F, detection rates. The precise positive and negative detection rates for both specialized classifiers using 20, 18, 16 and 14 stages are presented in Table 2.



(a) (b)

Fig. 4. (a) Left and right ear detection results, training with CD1 and testing with CD2. (b) Left and right ear detection results, training with CD1 and CD2, and testing with CD2.

Observing the figure, it is evident that the specialized detectors, i.e. those trained to detect only the left or only the right ear, perform better. For similar detection error rates, e.g. 5% the detection is around 92% and only 84% for the ear detector. The precise results presented in the table for the left and right ear detectors, suggest that both detectors do not perform exactly the same. Indeed, the left ear detector seems to have a lower false detection rate for similar positive detection rate. This effect can be justified by the fact that the false negative samples selection integrates some random decision during the training phase.

In summary, both specialized classifiers perform remarkably well for this scenario, while keeping a low false detection rate. In fact both detectors locate correctly more than 92% of the target patterns presenting an error rate around 5%. They are therefore robust ear detectors in the experimental setup. To process the 1947 images contained in the FERETCD2 set, even when their size is not homogeneous, the average processing time was 45 and 48 milliseconds respectively for the right and left detector. Figure 4b, presents the results achieve training with both subsets and testing with the FERETCD2 set.

In addition, we have tested the detectors with real video using a 640×480 webcam achieving close to real-time performance. This is achieved even when no temporal information is used to speed up the processing. The detectors are therefore valid to be applied for interactive purposes.

Table 2. Detection results using 20, 18, 16 and 14 stages.

Approach	Detection rate	False detection rate
Left ear (20)	0.8644	0.0015
Left ear (18)	0.9019	0.0067
Left ear (16)	0.9240	0.0252
Left ear (16)	0.9435	0.1469
Right ear (20)	0.8290	0.0015
Right ear (18)	0.8588	0.0041
Right ear (16)	0.8998	0.0185
Right ear (16)	0.9271	0.0632

4.2 Face detection improvement

To illustrate the interest of the facial features detection ability in conjunction with a face detector, we have performed a brief analysis on the the FDDB (Face Detection Data Set and Benchmark) dataset [11]. This dataset has been designed to study the problem of real face detection. The dataset contains a 5171 annotated faces taken from the Faces in the Wild dataset [6].

On that dataset we have applied face detection using two different approaches:

- Face detection using an available in OpenCV detector.

Table 3. Face detection results on the FDDB set

Approach	Detection rate	False detection rate
Face detection	71.55	6.57
Face detection and 6 FFs	65.94	1.85

– Face detection using an available in OpenCV detector, but confirming the presence of at least 2 inner facial features (eyes, nose, mouth, and ears) using the facial feature detectors present in OpenCV, plus our ear detectors.

The additional restriction imposed forces the location for a face candidate (detected by a face detector) of at least two inner facial features. The main benefit is that the risk of false detections is reduced as reflected in Table 3. However, the main benefit of the ear detection inclusion, is that when an ear is detected it additionally provides an evidence about the head pose, this is illustrated in Figure 5.

5 Conclusions

Observing the reliable classifiers trained by other researchers, we have used the Viola-Jones framework to train ear detectors. After the slow task of data gathering and training, they have been tested using the FERET database. Their respective detection results achieved have been presented suggesting a high detection rate. The specialized left and right ear detector performances evidences a detection rate larger than 92%, remarkably larger than the performance achieved by a general ear detector. These detectors are additionally reliable to be used in real-time applications employing standard webcams. These classifiers are therefore useful to any application requiring ear detection. For instance, we have applied the detector to the FDDB set to test the ability to suggest a lateral view. Other applications such as ear biometric systems, require an ear registration step that is now fast and simple.

We expect to explore further the combination of these classifiers with other facial feature detectors to improve face detection performance based on the combination of the evidence accumulation provided by inner facial feature detection. Such a detector would be more robust to slight rotations and occlusions.

Both classifiers reported in Figure 4b are now included in the OpenCV library. therefore other researchers can take them as baseline for comparison and improvement. In the next future, we will consider the addition of slightly rotated ear patterns to the positive set with the objective to analyze if a more sensitive classifier can be built.

References

1. Ayman Abaza abd Christina hebert and Mary Ann F. Harrison. Fast learning ear detection for real-time surveillance. In *Biometrics: Theory Applications and Systems (BTAS)*, 2010.

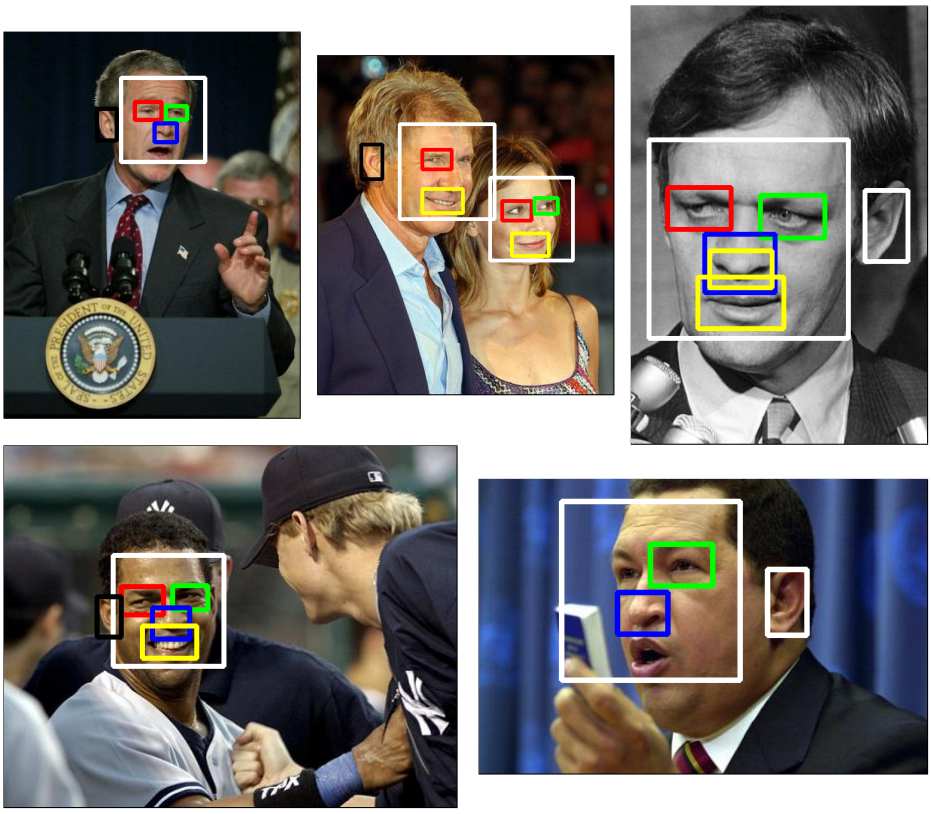


Fig. 5. FDDB detection samples with pose estimation based on facial features detection. The color code: red means left (in the image) eye detection, green means right (in the image) eye detection, blue means nose detection, yellow means mouth detection, black means left (in the image) ear detection, and white means right (in the image) ear detection

2. David Bradley. Profile face detection. http://www.davidbradley.info, 2003. Last accesed 15/7/2011.
3. Modesto Castrillón, Oscar Déniz, Daniel Hernández, and Javier Lorenzo. A comparison of face and facial feature detectors based on the violajones general object detection framework. *Machine Vision and Applications*, 22(3):481–494, 2011.
4. Kyong Chang, Kevin W. Bowyer, Sudeep Sarkar, and Barnabas Victor. Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1160–1165, 2003.
5. Robin Hewitt. Seeing with opencv. a computer-vision library. *Servo*, pages 62–65, January 2007.
6. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

7. Intel. Open Source Computer Vision Library, v2.3. http://opencv.willowgarage.com/wiki/, July 2011. (last visited July 2011).
8. S. M. S. Islam, M. Bennamoun, and R. Davies. Fast and fully automatic ear detection using cascaded adaboost. In *WACV*, 2008.
9. S. M. S. Islam, M. Bennamoun, R. Owens, and R. Davies. Biometric approaches of 2d-3d ear and face: A survey. *ADVANCES IN COMPUTER AND INFORMATION SCIENCES AND ENGINEERING*, pages 509–514, 2008.
10. Anil Jain, Patrick Flynn, and Arun A. Ross, editors. *HANDBOOK OF BIOMETRICS*. Springer, 2008.
11. Vidit Jain and Erik Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst., 2010.
12. Hannes Kruppa, Modesto Castrillón Santana, and Bernt Schiele. Fast and robust face finding via local context. In *Joint IEEE Internacional Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 157–164, October 2003.
13. Rainer Lienhart, Alexander Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM'03, 25th Pattern Recognition Symposium*, pages 297–304, Magdeburg, Germany, September 2003.
14. Rainer Lienhart, Luhong Liang, and Alexander Kuranov. A detector tree of boosted classifiers for real-time object detection and tracking. In *IEEE ICME2003*, pages 277–80, July 2003.
15. Rainer Lienhart and Jochen Maydt. An extended set of Haar-like features for rapid object detection. In *IEEE ICIP 2002*, volume 1, pages 900–903, September 2002.
16. Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 31(4):607 – 626, 2009.
17. P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face recognition algorithms. TR 6264, NISTIR, January 1999. `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00609311`.
18. Alejandro Reimondo. Haar cascades repository. http://alereimondo.no-ip.org/OpenCV/34, 2007. (last visited April 2010).
19. Naotoshi Seo. Tutorial: OpenCV haartraining (rapid object detection with a cascade of boosted classifiers based on haar-like features). `http://note.sonots.com/SciSoftware/haartraining.html`. (last visited June 2010).
20. Teodora Vatahska, Maren Bennewitz, and Sven Behnke. Feature-based head pose estimation from images. In *Proceedings of IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids)*, 2007.
21. Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):151–173, May 2004.