Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Modesto Castrillón · Oscar Déniz · Daniel Hernández · Javier Lorenzo

A Comparison of Face and Facial Feature Detectors Based on the Viola-Jones General Object Detection Framework

Received: date / Accepted: date

Abstract The human face provides useful information during interaction, therefore any system integrating Vision Based Human Computer Interaction requires fast and reliable face and facial feature detection. Different approaches have focused on this ability but only open source implementations have been extensively used by researchers. A good example is the Viola-Jones object detection framework that particularly in the context of facial processing has been frequently used. The OpenCV community shares a collection of public domain classifiers for the face detection scenario. However, these classifiers have been trained in different conditions and with different data but rarely tested on the same datasets. In this paper we try to fill that gap by analyzing the individual performance of all those public classifiers presenting their pros and cons with the aim of defining a baseline for other approaches. Solid comparisons will also help researchers to choose a specific classifier for their particular scenario. The experimental setup also describes some heuristics to increase the facial feature detection rate while reducing the face false detection rate.

Keywords face and facial feature detection · haar wavelets · human computer interaction · face datasets · OpenCV

1 Introduction

The human face is a main source of information during human interaction. Therefore, an automatic system designed for interacting with humans in a natural way based on Vision, i.e. comfortable Human Computer Interaction (HCI), requires to accomplish fast and reliable face and facial element detection to extract the information contained in a face [1,2] in both dynamic and static environments. In this context, such problems have received lots of attention in recent literature. Particularly the face detection approaches have evidenced a notorious improvement in terms of speed and reliability [3–5]. Among them, the work by Viola and Jones [5], designed for general object detection, has been quickly spread due to the availability of an open source implementation [6]. This approach has been extensively used for detecting different targets in the image, receiving great attention those related to the human face and its elements. Different authors have made their classifiers (generally not their training sets) public. However, as far as the authors of this paper know and not forgetting their preliminary work [7], except for face detection [8,9] and a reduced test for eye detection [9], no performance evaluation of them has been carried out.

In this paper we try to fill the gap by analyzing not only face detectors based on the Viola-Jones framework [5] but also facial feature detectors. Certainly there are different approaches for such task in the Computer Vision literature, but this framework offers a testbed to train and compare classifiers that use different training data and configuration setup.

The Viola-Jones based detectors can be labeled as coarse detectors as they do not provide precise location of the target. Indeed these detectors provide a target container that can evidence some alignment offset. However, they can provide detection results even at low resolution as we will see in the experiments, serving as a filter or even a preattentive stage for further and more precise detectors.

This two stage detection approach has already been applied in recent works as for example the system described in [10] where a coarse detection is applied before performing the more expensive and precise deformable template approach. Indeed facial feature detection is not a new topic, as it has been mainly tackled using model based approaches since the 90's [11], using range images [12], active illumination to localize the eyes [13], Gabor

M. Castrillón, O. Déniz, D. Hernández, J. Lorenzo
Universidad de Las Palmas de Gran Canaria
SIANI
Edificio Central del Parque Científico-Tecnológico
35017 Las Palmas, Spain.
Tel.: +34-928-459644
E-mail: mcastrillon@siani.es

filters [14], color and morphological operators for detection eyes and mouth [15], and deformable templates [16]. The face structure and symmetry was analyzed in [17] for precise eye location. More recent papers address the problem applying a hierarchical approach for precise location [18].

The utility of facial feature detection was exploited by face detection systems based on weak cues, such as color or motion, in order to confirm that a candidate blob is really a face [19,20]. Those systems were aimed to perform fast detection, therefore fast cues were needed to keep the system working in real time. For that reason their authors generally observed the darker appearance of facial elements in relation to their surroundings or local context particularly for frontal poses [19–22] or the brightness of the nose tip [23].

Their utility is also evident in terms of head pose estimation [22,24] and to align faces for further facial analysis. Just to give an example, some results suggest that coarse face element detection and the way that information is used for alignment seem to play an important role for reducing classification error [25].

Adaboost based detectors have been combined with temporal coherence to perform faster and reliable face detection in video stream processing [26,27]. For example, nose detection has been used for interface design as pointer [23], or combined with optical flow to implement non-contact interfaces for disabled users [28]. Similarly, mouth detection has been combined with speech recognition [29].

In this paper, we compare different public domain classifiers, based on Lienhart's implementation [6], related to face and facial element detection, in order to provide a baseline for future developments. Section 2 briefly introduces the detection approach. Sections 3 and 4 present respectively the experimental setup and the conclusions extracted.

2 Viola-Jones general object detection framework

The problem of face detection has experimented a notorious improvement in last years [30,31]. Recent approaches [4,5] have reduced dramatically the processing latency at high levels of accuracy, without restricting their behavior to some weak evidences such as skin color or motion presence. These approaches follow a sliding window approach, see Figure 1, where the pattern is searched at different scales of the input image by means of shifting the region of interest across the whole image and analyzing each time that area with a classifier.

Following the sliding window approach, the general object detector framework described in [5] has achieved remarkable results while becoming well known thanks to the implementation [8] integrated in OpenCV (Open Computer Vision Library) [32]. This framework is based on the idea of a boosted cascade of weak classifiers, i.e. each one has a high detection ratio, with a small true reject ratio. Each 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 while rejecting a certain fraction of those non-object patterns that have been incorrectly accepted by previous stage classifiers.

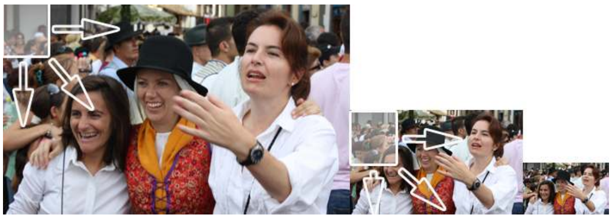


Fig. 1 The object detection approach shifts the matching window on the image at different resolutions.

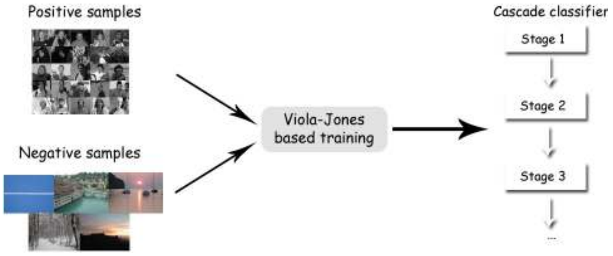


Fig. 2 Typical training procedure for a Viola-Jones' based classifier. Each classifier stage is obtained using positive and negative samples accepted by the previous stage.

Theoretically for a cascade of K independent 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:

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

Each stage classifier is selected considering a combination of features which are computed on the integral image, see Figure 3a. 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 [6] integrated in the OpenCV [32] extends the original feature set [5].

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.

Given an input image, the resulting classifier will report the presence and location of the object of interest.

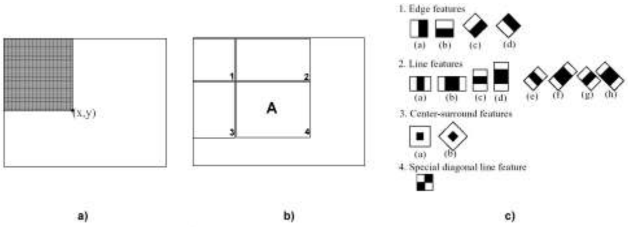

(x,y)
1
2
A
3
4
1. Edge features
(a) (b) (c) (d)
2. Line features
(a) (b) (c) (d) (e) (f) (g) (h)
3. Center-surround features
(a) (b)
4. Special diagonal line feature
a)
b)
c)

Fig. 3 a) The Integral Image stores integrals over subregions of the image. b) The sum of pixel values in A is $(x4, y4) - (x2, y2) - (x3, y3) + (x1, y1)$ [33] . c) Features prototypes considered in the implementation integrated in OpenCV [6,8].

3 Experiments

3.1 Experimental setup

As a first step, we have looked for researchers interested in face and facial feature detection based on the Viola-Jones framework in the community. The OpenCV group has been a very useful source in that sense, bringing together different authors who share an interest in this area. The resulting collection of classifiers gathered, thanks to their public availability or the kindness of their authors, are described in Table 1. Unfortunately the collection is not complete as some already published and referred classifiers could not be compared because they are not completely available [9,43]. Their targets are faces, separately frontal and profile faces, the upper body or also called head and shoulders, and the facial features: eyes, nose and mouth. Reference information is included for each one when it is available. The table contains also detailed information related to the minimal size of the pattern to be detected, and to the processing time in seconds needed to analyze the whole test set using a Windows XP box with an Intel Duo T5200 processor. The label assigned to each classifier will be used in the performance study presented below.

Once the classifiers are available, our second step was to select a wide collection of facial images to study their performance. For that purpose two datasets of facial images have been selected:

- The CMU database [4]. This dataset contains a collection of heterogeneous images, characteristic that from our point of view provides a better understanding of the classifier performance. As explained in [4], the dataset is divided into four different subsets *test*, *new-test*, *low-res* and *rotated* combining the test sets of Sung and Poggio [46] and Rowley, Baluja and Kanade [47]. The dataset and the annotation data correponding to 721 faces can be obtained at [48].
- The Yale Face database [49]. Contains higher resolution images of 165 frontal faces in different illumination conditions.

These datasets have been selected because of the different nature of the images contained in them. While the CMU dataset was conceived for face detection experiments, the Yale dataset presents frontal images of individuals at higher resolution. Features that, as seen below, make life easier to any face or facial feature detector. On the other side, the CMU dataset fits better real-life situations in concordance with the new trend that is being promoted lately among automatic face processing researchers [50].

3.2 Face detection

The detection performance of those classifiers designed to detect frontal and profile faces, and the head and shoulders pattern is compared in this Section. Thus observing Table 1, the frontal face classifiers being tested were provided thanks to R. Leinhart [8] (included in the OpenCV release), the profile face detector by D. Bradley [36], the set of face detectors described in [35], and the head and shoulders detector presented in [38].

Before performing the experiments, both datasets have been verified to check that the eyes were correctly annotated on all presented faces. Once they are available, the criterium to determine if a target (face or head in this case) was correctly detected is that the detection container overlaps all the facial elements and its width and height must not be greater than four times the distance between the annotated eyes.

For each classifier, its receiver operating characteristic (ROC) curve was computed applying the original release of each classifier, and some variants obtained reducing for each one its number of stages. Theoretically this action must increase both detection, *D*, and false detection, *F*, rates.

For the frontal face problem, observing the Area Under the Curve (AUC) of the resulting ROC curves for both datasets, see Figures 4 and 5, it is evidenced that three classifiers perform better than the others for both datasets. Zooming the results of the CMU dataset, two of them are slightly better than the third one. These classifiers are *FA1* (named $haarcascade\_frontalface\_alt$ in

Table 1 Public domain classifiers available.

Target	Reference	Availability	Size	Stages	Proc. time CMU	Proc. time Yale	Label
Frontal faces	[34,8]	[32]	24 × 24	25	66.6	6.3	FD
Frontal faces	[34,8]	[32]	20 × 20	21	70.8	8.9	FA1
Frontal faces	[34,8]	[32]	20 × 20	46	60.4	10.1	FAT
Frontal faces	[34,8]	[32]	20 × 20	20	63.6	9.0	FA2
Frontal faces	[35]	Personal com.	30 × 30	19	19.6	2.1	FW
Quarter turn faces	[35]	Personal com.	30 × 30	20	18.2	1.9	FWQ
Half turn faces	[35]	Personal com.	25 × 30	20	21.9	2.1	FWH
Profile faces	[36]	[37]	20 × 20	26	81.7	14.5	PR
Head and shoulders	[38]	[32]	22 × 18	30	125.5	11.7	HS1
Head and shoulders	[38]	[37]	22 × 20	19	194.8	19.6	HS2
Left eye	[26]	[37]	18 × 12	20	62	10.7	LE
Right eye	[26]	[37]	18 × 12	20	65.8	10.1	RE
Left eye	[39]	[32]	20 × 20	20	109.6	8.7	LES
Right eye	[39]	[32]	20 × 20	20	112.8	8.6	RES
Eye	[40]	[40]	25 × 15	5	20.6	4.7	EM
Eye	[41]	[41]	10 × 6	20	41	4.4	EU
Eye	[42]	[37]	24 × 12	104	40.1	6.6	ETS
Eye	[43]	Personal com.	28 × 14	16	41.8	4.3	EQT
Eye	[35]	Personal com.	24 × 21	20	26.6	3.8	EWF
Eye	[44]	[32]	20 × 20	24	44.7	5.2	ESH
Eye with glasses	[44]	[32]	20 × 20	30	56.4	8.0	ESHG
Quarter turn eye	[35]	Personal com.	24 × 23	20	27.9	3.7	EWQ
Half turn eye	[35]	Personal com.	22 × 24	20	32.8	4.9	EWH
Eye pair	[26]	[37]	45 × 11	19	24.3	2.8	EP
Eye pair	[26]	[37]	22 × 5	17	31.8	3.4	EPS
Eye pair	[45]	[37]	35 × 16	19	29.2	3.5	EPB
Nose	[26]	[37]	18 × 15	20	47.8	13.6	N
Nose	[35]	Personal com.	24 × 21	20	9.6	1.9	NWF
Quarter turn nose	[35]	Personal com.	24 × 22	20	12.6	2.3	NWQ
Half turn nose	[35]	Personal com.	24 × 24	20	12.2	2.7	NWH
Mouth	[26]	[37]	25 × 15	20	42.8	13.8	M
Mouth	[29]	[32]	32 × 18	18	12.1	2.5	Mouth
Mouth	[35]	Personal com.	24 × 16	19	14.1	3.1	MWF
Quarter turn mouth	[35]	Personal com.	24 × 15	20	13.6	3.3	MWQ
Half turn mouth	[35]	Personal com.	24 × 17	20	17.7	4.6	MWH

the OpenCV distribution [32]) and *FA2* (corresponding to *haarcascade_frontalface_alt2* in the OpenCV distribution [32]), closely followed by *FD* (the default classifier *haarcascade_frontalface_default* in OpenCV release). Among all the classifiers FW is the fastest, around three times, but its curve suggests a lower detection rate. The computational cost is similar for the best classifiers as suggested by Table 1. For those reasons *FA2* was selected as our preferred classifier and will be used in the experiments described in Section 3.4.

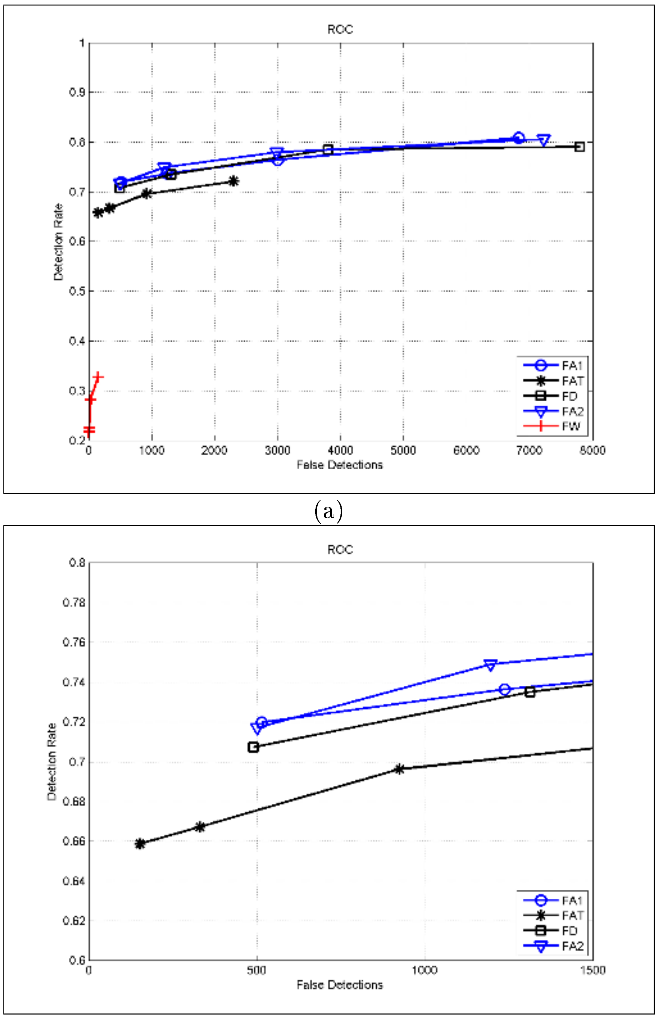
The comparison of the performance achieved for each dataset suggests a much better detection rate for the Yale dataset. This was an expected conclusion because the CMU dataset is less restricted in terms of head pose and resolution, i.e. it is closer to real-life conditions.

The CMU dataset selected has the additional characteristic of containing four subsets with images offering different conditions and therefore levels of difficulty. Observing Table 2 it is evidenced the lower performance achieved for the rotated subset. A fact that was expected as the Viola-Jones' framework is not able to accept more than slight variations in rotation with respect to the positive samples used.

These results have been improved in the literature by training additional detectors that can cope with some configurations of rotated faces. The performance of non frontal face detectors is presented in Figure 6. Observing the figure, the best detector for both datasets is *PR*. Reporting in any case lower, as expected for frontal face imagery, and less homogeneous performance than those exhibited by the frontal face detectors. Table 3 presents the results achieved for the two leading detectors considering the different subsets of the CMU database. Remember that the rotated subset contains in plane rotations that are less suitable for any profile detector, i.e. out of plane rotated faces.

Table 2 Frontal face detection for each subset of the CMU dataset

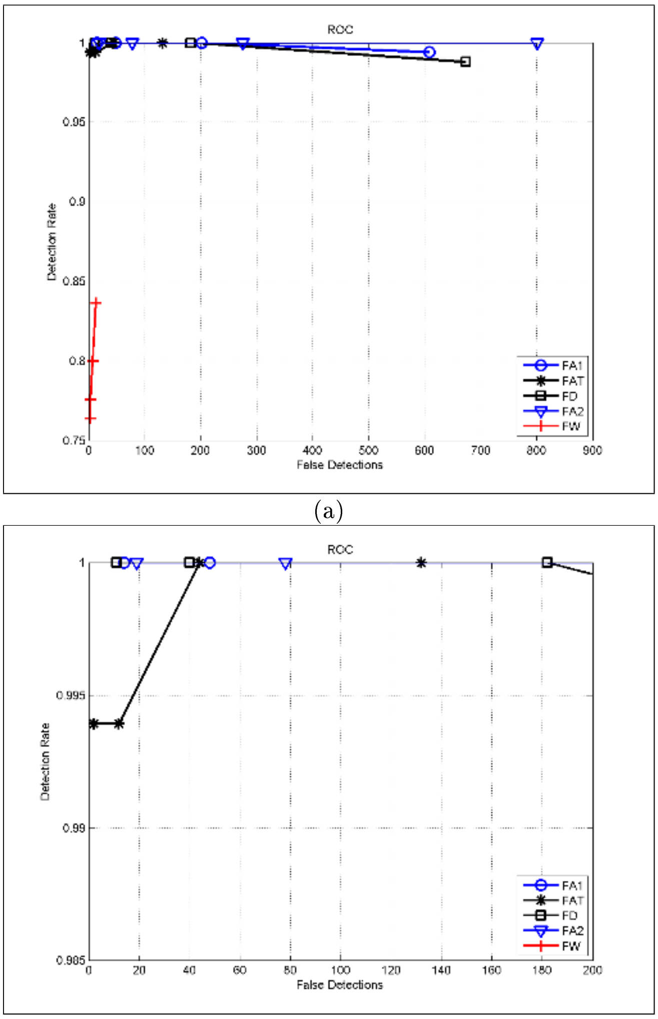
Subset	Detection performance
newtest	89.07%
test	86.98%
rotated	19.28%
test-low	83.56%



(a)

(b)

Fig. 4 (a) Face detection results obtained with the CMU database, (b) zoomed in results.



(a)

(b)

Fig. 5 (a) Face detection results obtained with the Yale database, (b) zoomed in results.

Table 3 Profile detection performance for each subset of the CMU dataset using the best two classifiers, i.e. PR and FWQ

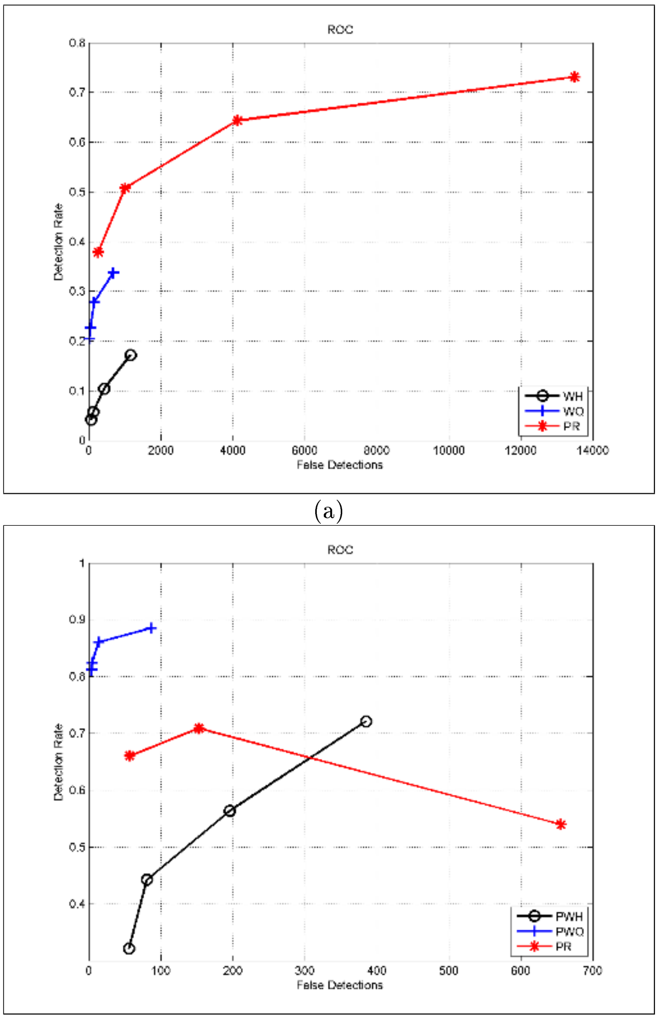
Subset	PR	FWQ
newtest	44.80%	44.26%
test	26.77%	33.72%
rotated	5.38%	0.11%
test-low	19.86%	19.11%

We have also paid attention to the head and shoulders detector, included in OpenCV 1.0 release [32], by H. Kruppa et al. [38], labelled as *HS1*. Figure 7 presents the results achieved considering both datasets, evidencing a rather low performance. For that reason a more recently trained classifier, *HS2*, following the same approach but with a larger training dataset has also been included in the experiments. It is notoriously better, a fact that made us suspect the presence of a bug in the classifier included in the OpenCV release. It must be noticed that this classifier requires the presence of the local context to detect, circumstance that does not occur for different images contained in the CMU dataset and for almost all the images contained in the Yale dataset.

3.3 Facial element detection

After the face detection experiments, a similar evaluation was performed with the different facial feature detectors. Their results are presented separately to emphasize that they belong to patterns of smaller size than a face or head, basically due to the fact that they are contained within a face. Five different targets are considered in this section: the left eye, the right eye, both eyes together, the nose and the mouth.

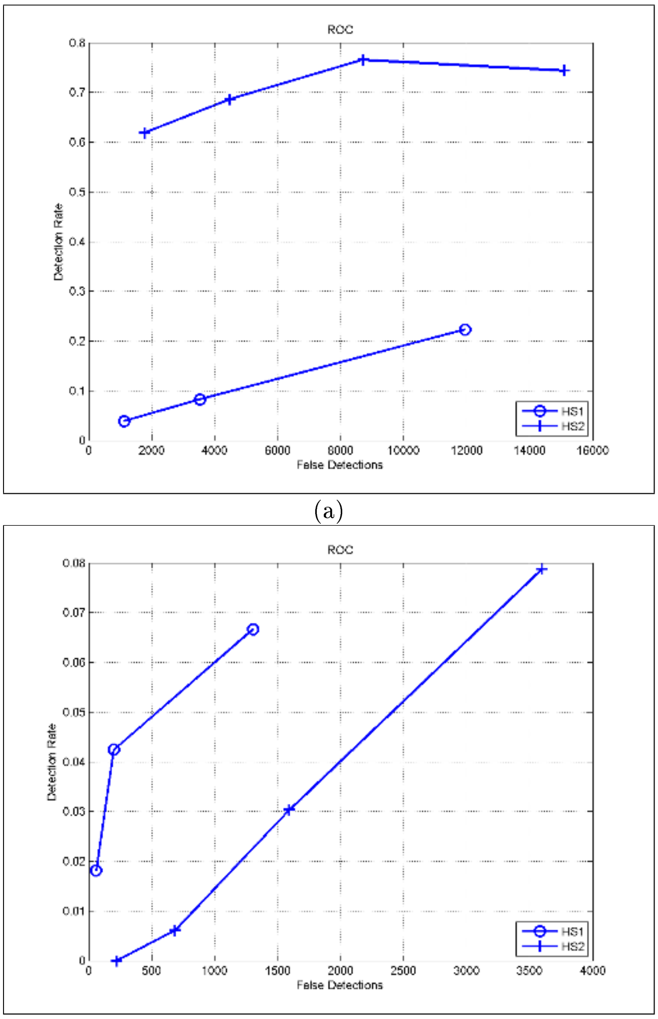
A facial feature is considered correctly detected if the distance to the annotated location is less than 1/4 of the actual distance between the annotated eyes. This



(a)

(b)

Fig. 6 (a) Profile detection results obtained with the (a) CMU database, (b) Yale database.



(a)

(b)

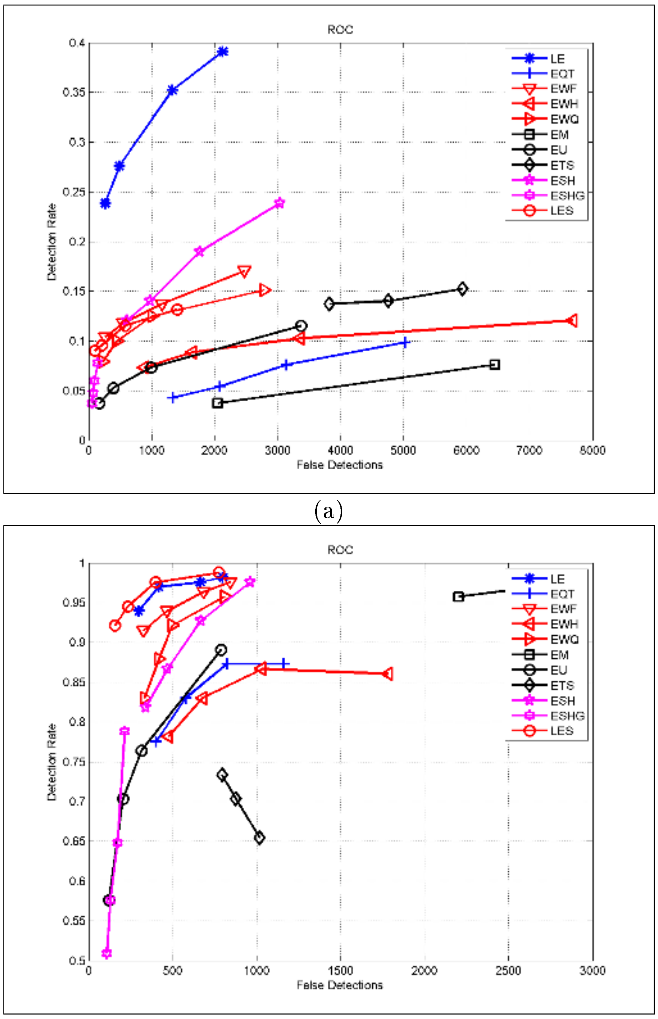
Fig. 7 (a) Head and shoulders detection results obtained with the (a) CMU database, (b) Yale database.

criterion was used to estimate the eye detection success originally in [51].

The results achieved for the left and right eyes, taken separately, are presented respectively in Figures 8 and 9. The behavior among the different classifiers is pretty similar for both eyes, but the performance is quite different for each dataset, i.e. the Yale dataset is clearly better suited for any classifier. Observe again the large difference in detection rate among both datasets. For the CMU dataset the detection rate is only for one classifier greater than 20% while greater than 90% for the Yale dataset. For all the classifiers, we have not considered the detection of the opposite eye as a false detection. It must be remarked that some classifiers presented in Table 1 are not specialized for a specific eye, therefore they were trained without paying attention to that aspect.

For the CMU dataset, the hardest attending to the results presented, a pair of classifiers clearly detects more while keeping a reduced rate of false detection, i.e. showing the lowest AUC of the resulting ROC curve. Those classifiers were *LE* for the left eye and *RE* for the right one (remember to refer to Table 1 to identify each classifier label). In both cases these classifiers are followed by *LES*, *RES* and *ESH*, all of them recently included in the OpenCV release. The latter classifier requires around 2/3 of the time consumed by those performing best, being also the second in processing cost.

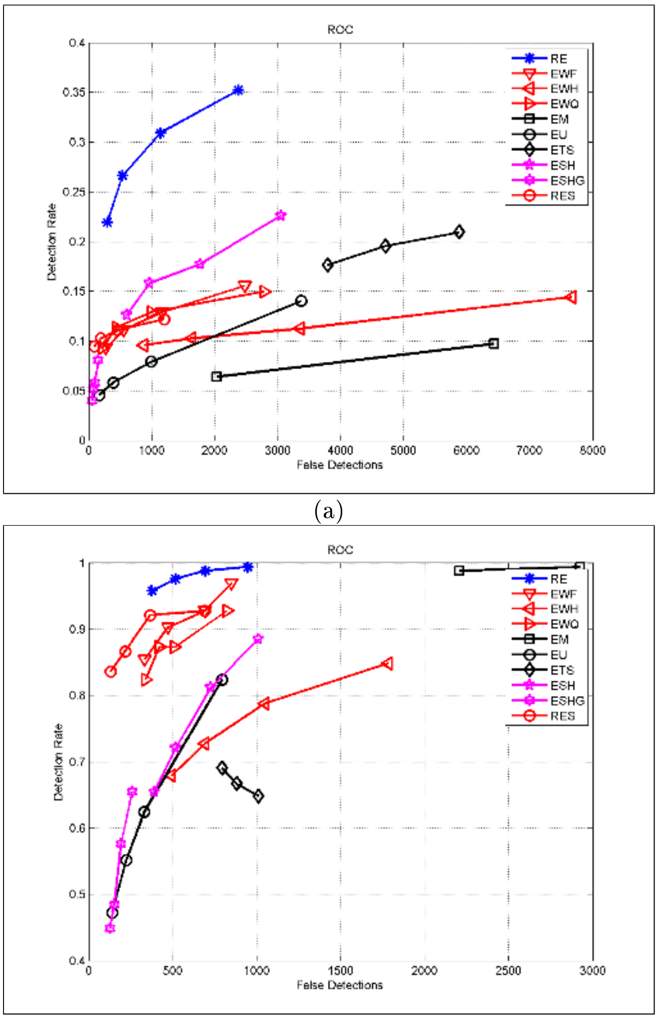
For the Yale dataset the general behavior is clearly improved thanks to the restricted scenario of this dataset: the faces are basically frontal, bigger and avoid any rotation. As before, classifiers *LE* and *RE* are in the leading group in terms of AUC but they also perform slower than any other. The reader must observe that *LES* got a slightly better rate, while *RES* got a slightly worse, reducing both the processing cost for this scenario. All these classifiers detect more than 90% of the eyes presented in the images once the face has been detected. For this dataset, the third classifier in terms of AUC is



(a)

(b)

Fig. 8 (a) Left eye detection results obtained with the (a) CMU database, (b) Yale database.



(a)

(b)

Fig. 9 (a) Right eye detection results obtained with the (a) CMU database, (b) Yale database.

EWF, which is additionaly one of the fastest.

LE and *RE* classifiers have a drawback in terms of processing needs, almost twice slower in relation to *EWF*, however its effect would be reduced if the search is performed, for example within the previously detected face.

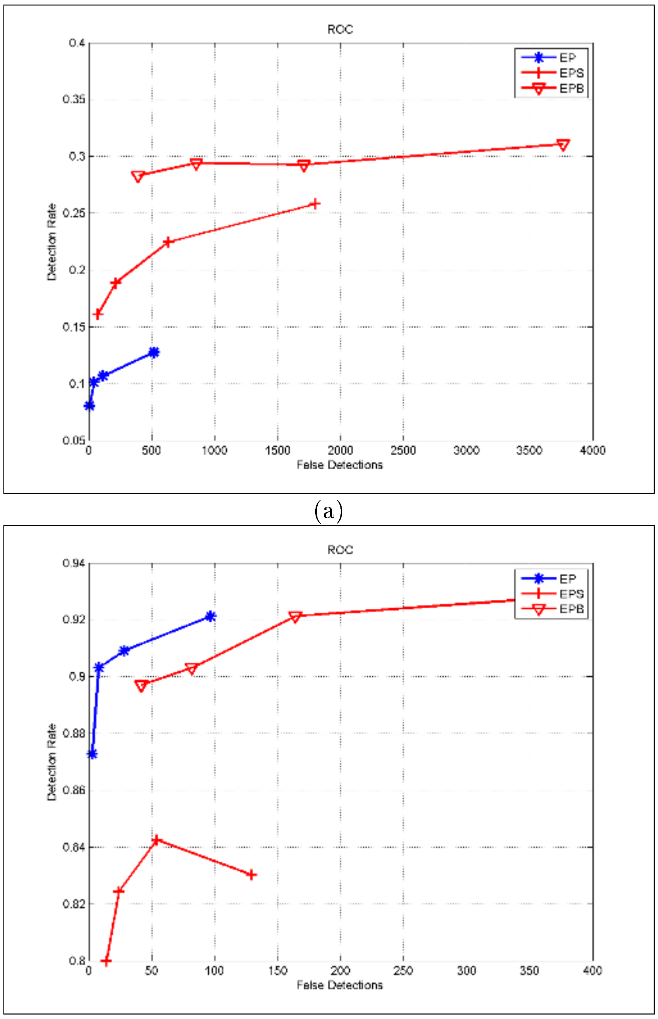
Reminding the reader the availability of four different subsets in the CMU dataset, Table 4 shows the different performance among the subsets for the best left eye detector, *LE*. If these rates are compared with those presented in Table 2, even when the pattern to track is smaller than the face where it is contained, the detection rate is better for the *rotated* subset. This is possible thanks to the apparent less sensitivity of the detector to slight in plane rotations. However, this feature detector is much more sensitive to the resolution providing a very low detection rate for the *test-low* dataset. This behavior was also observed in the other facial feature detectors.

Testing those classifiers designed to find both eyes simultaneously, i.e. the eye pair, the results achieved are presented in Figure 10. The best performance for the CMU dataset is given by *EPB*, but by *EP* for the Yale dataset. In both cases the processing cost was quite similar for all of them. The performance achieved suggests that the detection of the eye pair is more restrictive and sensitive to slight face rotation, similarly to face detection. Observe the tendency to present worse performance compared to the best single eye detectors.

Table 4 Left eye detection, using *LE* classifier, for each subset of the CMU dataset

Subset	Detection performance
newtest	36.06%
test	27.21%
rotated	25.11%
test-low	9.58%

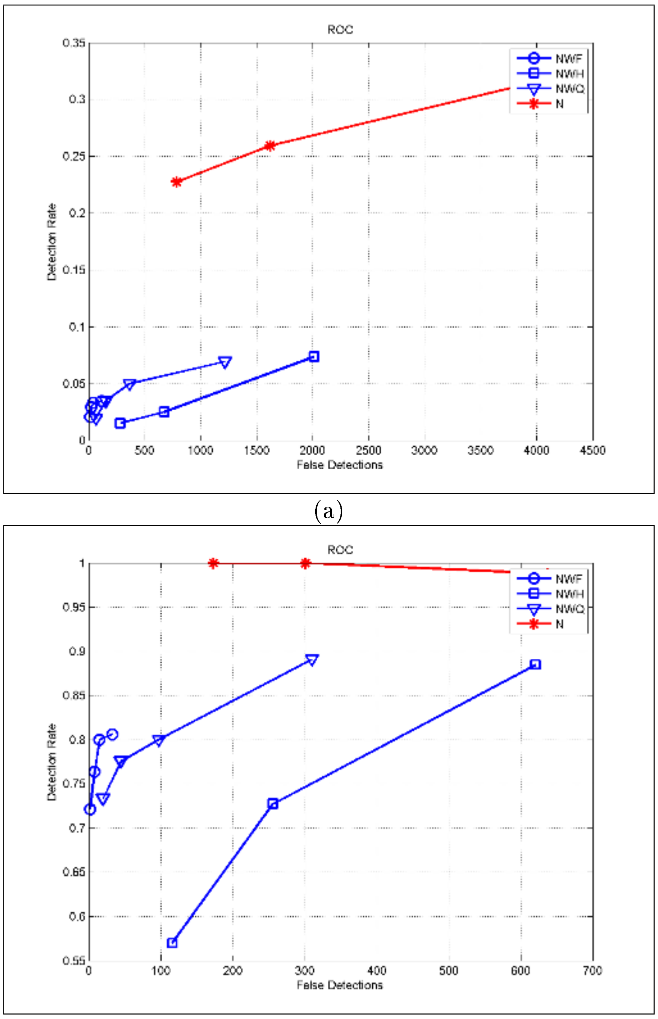
Nose detection results are presented in Figure 11. Observe again the better performance for the Yale dataset, reaching the best classifier reaching an AUC close to 1.



(a)

(b)

Fig. 10 (a) Eye pair detection results obtained with the (a) CMU database, (b) Yale database.



(a)

(b)

Fig. 11 (a) Nose detection results obtained with the (a) CMU database, (b) Yale database.

The clear winner for both datasets is *N*, however the processing time is four times higher.

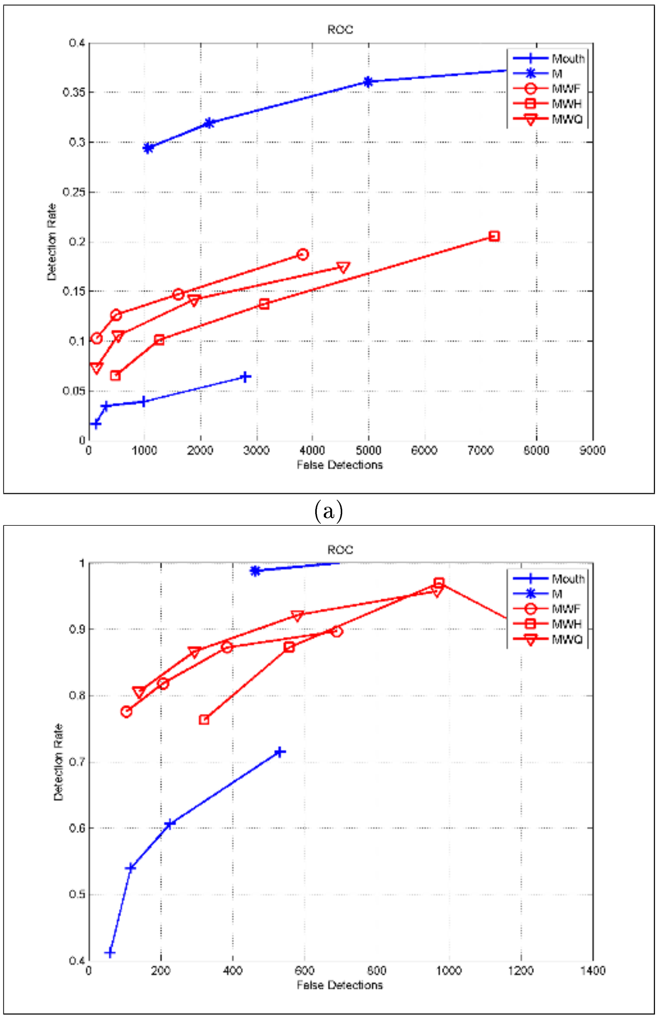
Mouth detectors have been also compared, and their performance evaluation is presented in Figure 12. The detector labelled as *M* is the best one in terms of the AUC for both datasets, but the processing cost is around four times the others. It can also be observed that its false detection rate is increased compared to the best nose detector. The larger appearance variability of a mouth could explain this fact.

In summary, according to the experiments carried out in this work, the best facial element detectors available in the public domain in terms of AUC are: *RE*, *LE*, *EPB*, *N* and *M*. Facial feature detectors evidenced a lower detection rate when processing the CMU images, i.e. less restricted faces.

This fact is justified by the less controlled conditions of the images contained in that dataset, circumstance that leaded to try to detect quite low resolution facial feature patterns. However, their detection rate is higher when processing the Yale dataset. Even the best face detector was not able to locate the always evident face. Certainly face feature detection could cope with difficult faces for face detectors, with the advantage of the need of geometric coherence. The only drawback is that the processing time will be larger.

3.4 Combined detection

The previous sections suggested that facial feature detection behaves worse than face detection in low resolution images. A likely reason for that effect, already mentioned, is related to the difference in terms of dimensions among them. A face will always be bigger than the facial elements contained in it, and therefore, easier to detect with classifier specialized in targets of similar sizes.



(a)

(b)

Fig. 12 (a) Mouth detection results obtained with the (a) CMU database, (b) Yale database.

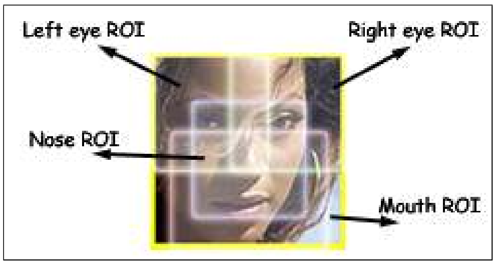


Fig. 13 Rough ROIs used to bound the search area for the facial feature detectors corresponding to the left and right eyes (in the image), the nose and the mouth in approaches **C2** and **C3**.

Applying the collection of classifiers to the whole image will increase notoriously the processing time required per frame. When real time is required, that is a drawback. For that reason in this section we will apply facial feature detection only in those areas were a face has been detected, in order to analyze if their performance can be improved, and additionally, to observe if the no detection of facial elements can be used as a filter to remove some false face detections.

To detect faces we used the *FA2* classifier, see Table 1, but using only 18 stages. This classifier suggested in Section 3.2 the best relation between processing speed and detection ratio. Once the face has been detected, each facial element was searched using three different strategies:

1. Searching the facial elements within the container of the detected face (**C1**).
2. Once the face container is given, searching each facial element only in the Region of Interest (ROI) expected for a frontal face. Figure 13 illustrates the ROIs used (**C2**).
3. Scaling up the face container before searching each face feature within its expected ROI, and bounding the acceptable target size according to the width of the face container (up to 1/6 is accepted). (**C3**).

The intention of approach **C2** is to reduce processing cost and false detection, however the use of too restrictive ROIs can lead to a detection rate decrease. The third approach **C3** will try to overcome the problem due to the lower resolution of the inner facial features, by scaling up the ROI to increase the pattern size before searching them.

For approaches **C2** and **C3**, each feature is searched always in a subwindow coherent with the feature location for a frontal face. Their specialization could affect to reduce the false positive detection. The different ROIs, see Figure 13, considering that sx and sy are the face container dimensions (respectively width and height) used in the experiments are:

1. Left eye: Left upper corner of the ROI are $(0, 0)$ and the dimensions of the ROI $(sx * 0.6, sy * 0.6)$.
2. Right eye: Left upper corner of the ROI are $(sx * 0.4, 0)$ and the dimensions of the ROI $(sx * 0.6, sy * 0.6)$.
3. Eye pair: Left upper corner of the ROI are $(0, 0)$ and the dimensions of the ROI $(sx, sy * 0.6)$.
4. Nose: Left upper corner of the ROI are $(sx * 0.2, sy * 0.25)$ and the dimensions of the ROI $(sx*0.6, sy*0.6)$.
5. Mouth: Left upper corner of the ROI are $(sx*0.1, sy*0.4)$ and the dimensions of the ROI $(sx*0.8, sy*0.6)$.

The results achieved for each facial feature detector processing both datasets using the three different approaches, are summarized in Table 5. The error shown is related to the actual distance of the eyes in the image. The average error reported is rarely higher than 6%. As a general behavior it is observed the importance of scaling up the image before performing the search for the CMU dataset to increase the number of correct detections, being almost useless for the Yale dataset. Therefore scaling up the ROI might only make sense for a particular range of face dimensions because for bigger faces it does not

Table 5 Average distance errors obtained for the different facial feature detectors applying search strategies **C1**, **C2** and **C3**. A location error of 1 means that it is equal to the actual distance between the eyes. Only successful detections are considered to calculate the error.

		CMU database			Yale database		
Target	**Label**	**C1**	**C2**	**C3**	**C1**	**C2**	**C3**
Left eye	LE	0.0259	0.025	0.0527	0.0436	0.0441	0.0414
Right eye	RE	0.0295	0.0286	0.063	0.0379	0.0383	0.0370
Left Eye	EM	0.0365	0.0365	0.0781	0.0964	0.0962	0.0933
Right Eye	EM	0.0441	0.0475	0.0868	0.0996	0.1012	0.1119
Left Eye	EU	-	-	0.0102	0.06	0.06	0.0555
Right Eye	EU	-	0.0076	0.0099	0.0475	0.547	0.0497
Left Eye	ETS	0.0378	0.0267	0.0543	0.0552	0.0459	0.0471
Right Eye	ETS	0.0389	0.0274	0.0663	0.0499	0.0429	0.0397
Left Eye	EQT	0.0183	0.0082	0.0223	0.0412	0.0376	0.0475
Left eye	EWF	0.0523	0.0522	0.0836	0.0835	0.0837	0.1003
Right eye	EWF	0.0443	0.039	0.0741	0.0761	0.0794	0.0942
Left Eye	ESH	0.0328	0.0265	0.0628	0.0665	0.0674	0.0691
Right Eye	ESH	0.0398	0.0265	0.0715	0.0518	0.0498	0.0509
Left Eye	ESHG	0.0162	0.0151	0.0360	0.0518	0.0512	0.0444
Right Eye	ESHG	0.0194	0.0172	0.0337	0.0433	0.0399	0.0379
Left Eye	LES	0.0479	0.0453	0.0926	0.0921	0.0907	0.0987
Right Eye	RES	0.0524	0.0496	0.0892	0.0993	0.0994	0.0995
Quarter turn left eye	EWQ	0.0496	0.0474	0.0754	0.0784	0.0795	0.0966
Quarter turn right eye	EWQ	0.0483	0.0474	0.0747	0.0757	0.0755	0.0898
Half turn left eye	EWH	0.0521	0.0525	0.0802	0.0856	0.0871	0.1052
Half turn right eye	EWH	0.0472	0.0474	0.0796	0.0821	0.0866	0.0979
Eye pair	EP	0.0486	0.0486	0.0902	0.1058	0.1058	0.1116
Eye pair	EPS	0.0465	0.0465	0.0827	0.1232	0.1232	0.1261
Eye pair	EPB	0.0863	0.07863	0.123	0.1224	0.1301	0.1219
Nose	N	0.0724	0.0709	0.0936	0.0410	0.0365	0.0404
Nose	NWF	-	0.0136	0.0529	0.0872	0.0826	0.1075
Quarter turn nose	NWQ	0.0235	0.0124	0.0719	0.1017	0.0067	0.1112
Half turn nose	NWH	-	0.0496	0.0452	0.1722	0.1581	0.1602
Mouth	M	0.0526	0.0535	0.0761	0.0511	0.0556	0.0519
Mouth	Mouth	0.0064	0.0096	0.0126	0.0661	0.04	0.0446
Mouth	MWF	0.0247	0.0236	0.0375	0.0516	0.0516	0.0528
Quarter turn mouth	MWQ	0.02	0.0135	0.0309	0.0477	0.0422	0.0483
Half turn mouth	MWH	0.0338	0.0192	0.0467	0.0715	0.0677	0.0747

bring any benefit and will only increase the computational cost.

The relation between true and false detections is presented in Figures 14, 15 and 16, but only for the subset containing the best detectors of each facial feature, using labels from Table 1, to improve the plot legibility.

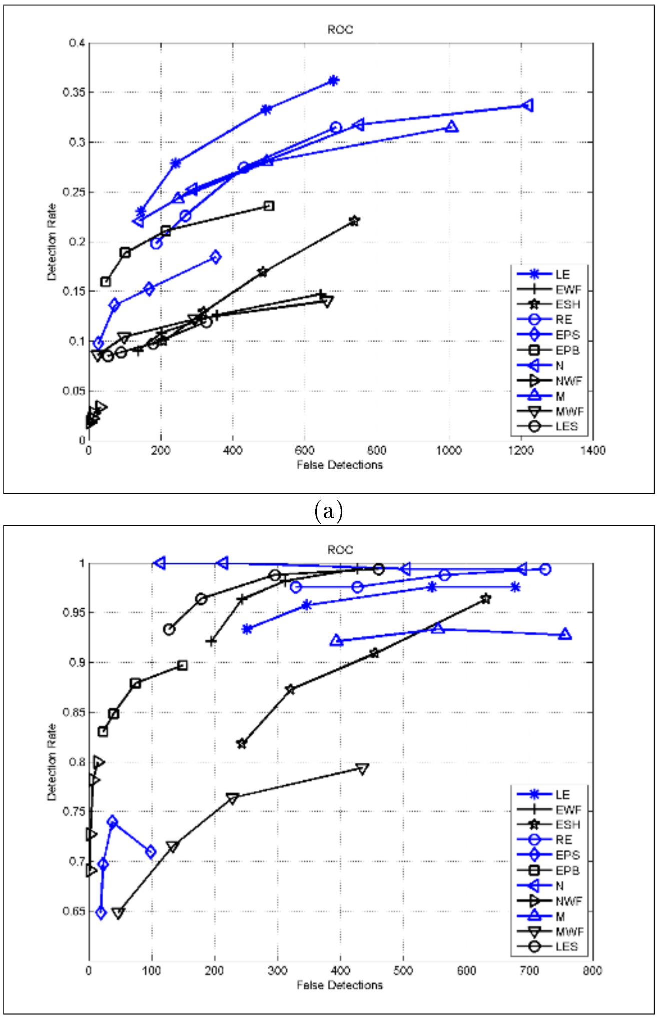
The Figure 14 presents the results for the first approach (**C1**). As we reduced the search area to the face detected, both positive detection and false detection rates are lower than those presented previously in Section 3.3. Observing the results for the CMU dataset, the left eye detector seems to be the best one in term of positive detections. For the Yale dataset, the nose detector is almost perfect.

Making use of not very narrow ROIs, see Figure 13, for each facial feature instead of searching in the whole face, the second approach (**C2**), reduces considerably the number of false detections, as shown in the Figure 15 without increasing significantly the loss of true detections.

The third approach considered (**C3**), scales up the ROI before applying the detection, while restricting the possible size of the target found. This heuristic increases significantly the facial feature detection rate for the CMU dataset, more than twice, but also the false detection rate slightly, as suggested by comparing Figures 15 and 16.

Observing the results achieved, a soft heuristic can be used to remove some false detections. Big enough face detections, larger than 40 pixels in width, that report no facial feature detection are not accepted as face detections. Applying this filter to the CMU dataset the detection rate was almost not affected, while achieving an improvement in false detection reduction. To give an example, the initial number of false face detections was 47, but applying this approach was reduced to 35.

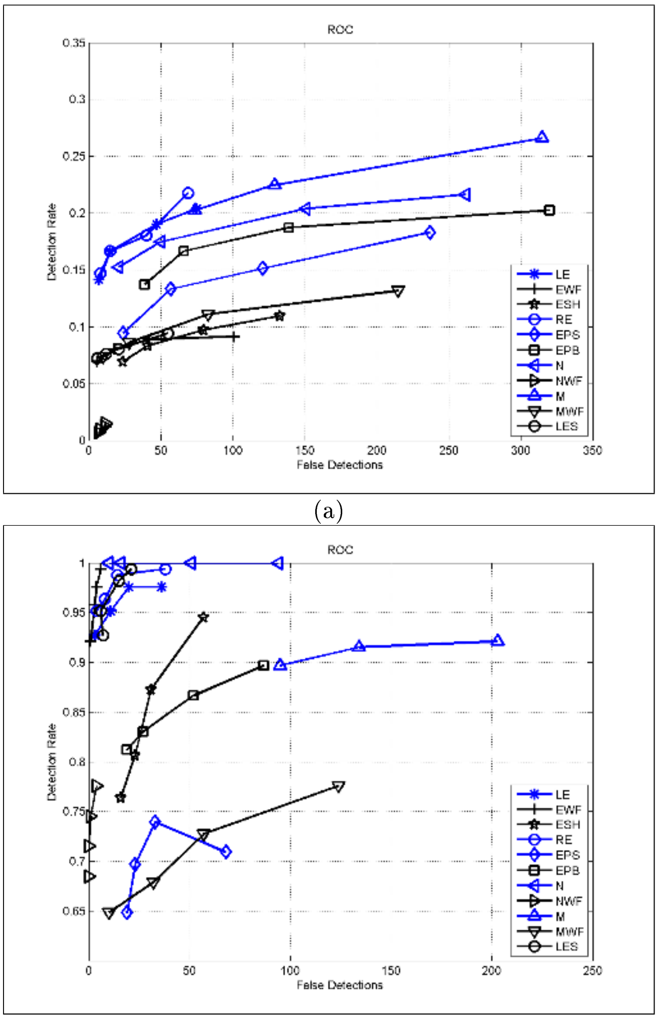
Some false facial detections reported also false face element detection. That situation happened for four images of the CMU dataset. Two of them are presented in Figure 17. For example in **our-cards**, see Figure 17a, no annotated face is present, but the face detector finds at least two (only the one with inner features detected is depicted). It does not seem to be a false detection. Similarly happens with ***rot5***, only the human faces are annotated, but Peggy is there, see Figure 17b. Other two detections are present with ***soccer*** and ***Germany*** images. The detections are incorrect only because the container is a little bit bigger than it should. Indeed the



(a)

(b)

Fig. 14 Facial element detection performance for **C1** (a) CMU database, (b) Yale database.



(a)

(b)

Fig. 15 Facial element detection performance for **C2** (a) CMU database, (b) Yale database.

facial features detected could be used to fit the container better.

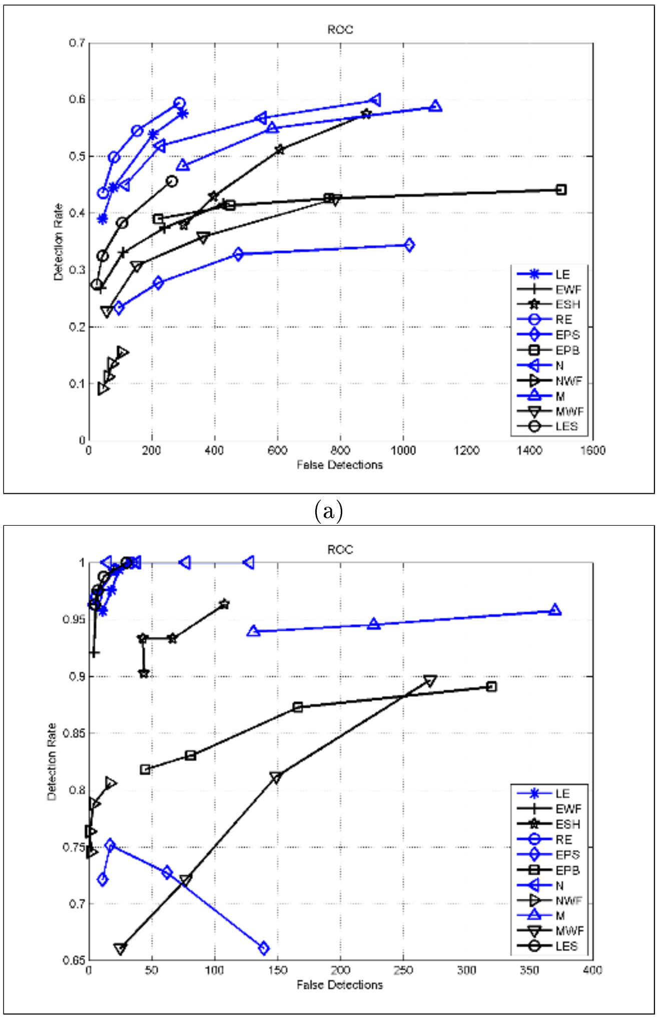
4 Conclusions

This paper studied the possibilities of current public domain classifiers, based on the Viola-Jones' general object detection framework [5], for face and facial feature detection. The available detectors based on this framework have been extensively analyzed with two public datasets of different nature. Among the frontal face detectors we have observed a similar performance for two of them *FA1* and *FA2* both included in the OpenCV release, and slightly worse for profile and head and shoulders detectors.

A similar experiment has been done with facial feature detectors. When processing the whole image, the best facial element detectors available in the public domain in terms of AUC are: *RE*, *LE*, *EPB*, *N* and *M*. Facial feature detectors evidenced a lower detection rate when processing the CMU images, i.e. less controlled facial images. Additionally we investigated the possibilities of combining face and facial feature detectors. Under this approach, facial feature detectors are applied only within regions containing a face, reducing the computational cost. The integration of a ROI for each facial feature and the scaling up of the image for a specific range of face dimensions, provide much better detectors in term of AUC.

An aspect that has not been covered in this work is the possibility of using multiple detectors to improve confidence, but certainly some of the detectors included in this study are right now not very flexible to be applied in a wide range of image resolutions closer to real life images.

The results presented here can be used by developers to trade off computational cost, detection rate, false de-



(a)

(b)

Fig. 16 Facial element detection performance for **C3** (a) CMU database, (b) Yale database.

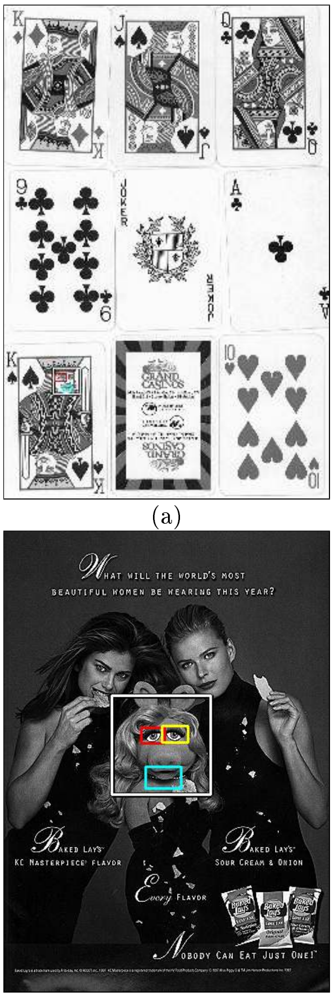
tection rate and precision, in order to build face and feature detection systems for any specific application. Additionally these results can serve as a valid baseline to compare with future or already computed classifiers achieved by any reader.

ACKNOWLEDGEMENTS

We would like to thank Q. Tao and J. Fernández for allowing us to use their unpublished classifiers in the experiments.
Work partially funded by the Spanish Ministry of Science and Innovation funds (TIN2008-06068).

References

1. A. Pentland, Looking at people: Sensing for ubiquitous and wearable computing, IEEE Trans. on Pattern Analysis and Machine Intelligence (2000) 107–119.
2. M. Turk, Computer vision in the interface, Communications of the Association for Computing Machinery 47 (1) (2004) 61–67.
3. S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: European Conference Computer Vision, 2002, pp. 67–81.
4. H. Schneiderman, T. Kanade, A statistical method for 3d object detection applied to faces and cars, in: IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 1746–1759.
5. P. Viola, M. J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 151–173.
6. R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, in: IEEE ICIP 2002, Vol. 1, 2002, pp. 900–903.
7. M. Castrillón, O. Déniz, L. Antón, J. Lorenzo., Face and facial feature detection evaluation, in: Proceedings of the International Joint Conference on Computer Vi-


What will the world's most beautiful women be wearing this year?
Baked Lays KC Masterpiece flavor
Baked Lays Sour Cream & Onion
Every Flavor
Nobody Can Eat Just One!

(a)

(b)

Fig. 17 False face detections that reported at least two facials features detection corresponding to ***our-cards*** (see the rectangle on the bottom left card) and ***rot5*** images (both contained in CMU dataset [4]) from the *new-test* and the *rotated* subsets respectively. Does the reader consider them as false detections?

sion and Computer Graphics Theory and Applications, VISIGRAPP, 2008, pp. 167–172.

8. R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, in: DAGM'03, Magdeburg, Germany, 2003, pp. 297–304.
9. A. Schmidt, A. Kasinski, The performance of the haar cascade classifiers applied to the face and eyes detection, in: Springer (Ed.), Proceedings of the 5th International Conference on Computer Recognition Systems CORES2007, Vol. 45 of Advances in Soft Computing, Wroclaw, Poland, 2007, pp. 816–823.
10. K.-M. Lam, H. Yan, Locating and extracting the eye in human face images, Pattern Recognition 29 (5) (1996) 771–779.
11. G. Chow, X. Li, Towards a system for automatic facial feature detection, Pattern Recognition 26 (12) (1993) 1739–1755.
12. Y. Y., D. L. S., Labeling of human face components from range data, CVGIP: Image Understanding 60 (2) (1994) 168–178.
13. C. H. Morimoto, M. Flickner, Real time multiple face detection using active illumination, in: 4th IEEE International Conference on Automatic Face and Gesture Recognition, 2000, p. 8.
14. F. Smeraldi, O. Carmona, J. Bigün, Saccadic search with Gabor features applied to eye detection and real-time head tracking, Image and Vision Computing 18.
15. R.-L. Hsu, M. Abdel-Mottsleb, Face detection in color images, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 686–706.
16. A. L. Yuille, P. W. Hallinan, D. S. Cohen, Feature extraction from faces using deformable templates, International Journal of Computer Vision 8 (2) (1992) 99–111.
17. W. Huang, Q. Sun, C.-P. Lam, J.-K. Wu, A robust approach to face and eyes detection from image with cluttered background, in: International Conference Pattern Recognition, Vol. 1, 1998, pp. 110–114.
18. L. Ding, A. Martinez, Precise detailed detection of faces and facial features, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008., 2008, pp. 1–7.
19. L.-P. Bala, K. Talmi, J. Liu, Automatic detection and tracking of faces and facial features in video, in: Picture Coding Symposium, 1997, pp. 251–256.
20. K. Sobottka, I. Pitas, A novel method for automatic face segmentation, face feature extraction and tracking, Signal Processing: Image Communication 12 (3).
21. S. Feyrer, A. Zell, Detection, tracking and pursuit of humans with autonomous mobile robot, in: Proc. of International Conference on Intelligent Robots and Systems, Kyongju, Korea, 1999, pp. 864–869.
22. A. Nikolaidis, I. Pitas, Facial feature extraction and determination of pose, Pattern Recognition 33 (2000) 1783–1791.
23. D. O. Gorodnichy, G. Roth, Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces., Image and Vision Computing 22 (12) (2004) 931–942.
24. T. Horprasert, Y. Yacoob, L. Davis, Computing 3-d head orientation from a monocular image sequence, in: Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vermont, USA, 1996, pp. 242–247.
25. A. Rabie, C. Lang, M. Hanheide, M. Castrillón, G. Sagerer, Automatic initialization for facial analysis in interactive robotics, in: 6th International Conference on Computer Vision Systems, Vision for Cognitive Systems, 2008, pp. 517–526.
26. M. Castrillón Santana, O. Déniz Suárez, M. Hernández Tejera, C. Guerra Artal, ENCARA2: Real-time detection of multiple faces at different resolutions in video streams, Journal of Visual Communication and Image Representation (2007) 130–140.
27. K. Lin, J. Huang, J. Chen, C. Zhou, Real-time eye detection in video streams, in: Fourth International Conference on Natural Computation, 2008, pp. 193–197.
28. X. Ren, J. Song, H. Ying, Y. Zhu, X. Qiu, Robust nose detection and tracking using gentleboost and improved Lucas-Kanade optical flow algorithms, Lecture Notes in Computer Science: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues 4681 (2007) 1240–1246.
29. L. Liang, X. Liu, X. Pi, Y. Zhao, A. V. Nefian, Speaker independent audio-visual continuous speech recognition, in: International Conference on Multimedia and Expo, 2002, pp. 25–28.
30. E. Hjelmas, B. K. Low, Face detection: A survey, Computer Vision and Image Understanding 83 (3) (2001) 236–274, http://dx.doi.org/10.1006/cviu.2001.0921.
31. M.-H. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: A survey, Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58, http://dx.doi.org/10.1109/34.982883.
32. Intel, Intel Open Source Computer Vision Library, v1.1ore, http://sourceforge.net/projects/opencvlibrary/ (October 2008).
33. R. Hewitt, Seeing with opencv. a computer-vision library, Servo (2007) 62–65.
34. R. Lienhart, L. Liang, A. Kuranov, A detector tree of boosted classifiers for real-time object detection and tracking, in: IEEE ICME2003, 2003, pp. 277–80.
35. P. I. Wilson, J. Fernandez, Facial feature detection using haar classifiers, Journal of Computing Sciences in Colleges 21 (2006) 127–133.
36. D. Bradley, Profile face detection, http://www.davidbradley.info/publications/bradley-iurac-03.swf, last accesed 5/11/2007 (2003).
37. A. Reimondo, Haar cascades repository, http://alereimondo.no-ip.org/OpenCV/34 (2007).
38. H. Kruppa, M. Castrillón Santana, B. 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), 2003, pp. 157–164.
39. S. Yu, Tree-based 20x20 eye detectors, http://yushiqi.cn/research/eyedetection (2009).
40. M. Wimmer, Eyefinder, http://www9.cs.tum.edu/people/wimmerm/se/project.eyefinder/, last accesed 5/11/2007 (2004).
41. Urtho, Eye detector, http://face.urtho.net/, last accesed 5/9/2007 (2006).
42. T. Shan, Security and surveillance, http://www.itee.uq.edu.au/ sas/people.htm (2008).
43. G. Beumer, Q. Tao, A. Bazen, R. Veldhuis, A landmark paper in face recognition, in: 7th International Conference on Automatic Face and Gesture Recognition, 2006. FGR 2006, IEEE Computer Society Press, Southampton, UK, 2006, p. 78.
44. S. Hameed, Eye cascade, http://umich.edu/~shameem (October 2008).
45. Y. Bediz, G. B. Akar, View point tracking for 3d display systems, in: 3th European Signal Processing Conference, EUSIPCO-2005, 2005.
46. K.-K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (1) (1998) 39–51.
47. H. A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (1) (1998) 23–38.
48. Carnegie Mellon University, CMU/VACS image database: Frontal face images, http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html, last accesed 5/11/2007 (1999).
49. P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, IEEE Trans. on PAMI 19 (7) (1997) 711–720.

50. G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007).
51. O. Jesorsky, K. J. Kirchberg, R. W. Frischholz, Robust face detection using the Hausdorff distance, Lecture Notes in Computer Science. Procs. of the Third International Conference on Audio- and Video-Based Person Authentication 2091 (2001) 90–95.