Viola-Jones Based Detectors: How Much Affects the Training Set?

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

SIANI
Edif. Central del Parque Científico Tecnológico
Universidad de Las Palmas de Gran Canaria
35017 - Spain

Abstract. This paper presents a study on the facial feature detection performance achieved using the Viola-Jones framework. A set of classifiers using two different focuses to gather the training samples is created and tested on four different datasets covering a wide range of possibilities. The results achieved should serve researchers to choose the classifier that better fits their demands.

Keywords: Viola-Jones detectors, facial feature detection, training sets.

1 Introduction

The Viola-Jones face detector [16] has been extensively used thanks to the implementation available [10] in the OpenCV (Open Computer Vision) library [7]. However, Viola and Jones designed a general object detection framework that can be used for other objects. Its OpenCV implementation allows researchers to train their own classifier(s). Previously, during the sample gathering stage a large set of images is built with samples containing the object to detect (positive samples) and others not containing the target (negative samples).

Positive and negative samples gathering, data annotation, data preparation and training are uncomfortable and slow tasks that have been summarized in different brief tutorials, e.g. [14]. In this sense more recent implementations [2] have tried to keep the performance while reducing the training and test processing.

Within the facial analysis scenario, facial feature detection is a topic of interest as is may serve to reduce false positive detections when using a face detector, or to better align a detected face. Thanks to OpenCV, different face related classifiers are available to a large community of researchers [7,11]. Their performance have already been compared with different test sets, but no details related to the samples used during their training stage are available.

In this paper we train different facial feature classifiers making use of training sets of different nature, and test them with a large heterogeneous collection of face datasets, in terms of pose, illumination and resolution. We aim at providing researchers hints about how to build a detector for their particular application characteristics.

J. Vitrià, J.M. Sanches, and M. Hernández (Eds.): IbPRIA 2011, LNCS 6669, pp. 297–304, 2011.

Section 2 summarizes the Viola-Jones object detection framework. The different datasets are briefly described in Section 3 and the results and conclusions in sections 4 and 5 respectively.

2 Viola-Jones General Object Detection Framework

Automatic face detectors have received researchers attention in last years, evolving notoriously [5,17]. In this sense recent approaches [13,16] have reduced dramatically the processing latency at high levels of accuracy, without requiring restricted heuristics based on cues such as skin color or motion. These approaches make use of a sliding window that is shifted at different scales across the whole image. Each time the area is checked with a classifier to verify whether the target pattern is present.

Following the sliding window approach, face detectors based on the framework described in [16] have achieved remarkable results while becoming well known thanks to the implementation [10] integrated in OpenCV [7]. 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 reduced 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, 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.

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:

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

Each stage classifier is selected considering a combination of features which are computed on the integral image. These features are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses. The implementation [10] integrated in the OpenCV [7] extends the original feature set [16].

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.

The availability of different tutorials, e.g. [14], help OpenCV users to collect, annotate and structure the data before building the different classifiers that are later tested with an independent set of images.

3 Datasets

Being interested in testing a facial feature detector performance, we previously selected some face datasets to test. Different datasets have been used in the past to analyze face detection performance. However, we wanted to cover a wide range of situations to better characterize the classifiers under study. For that purpose four datasets of facial images have been selected:

- The CMU database [13] contains a collection of heterogeneous images divided into four different subsets *test*, *new-test*, *low-res* and *rotated* combining the test sets of Sung and Poggio [15] and Rowley, Baluja and Kanade [12]. The dataset and the annotation data corresponding to 721 faces can be obtained at [3].
- More recently initiatives such as FIW [6] have introduced new challenging situations to test the performance of the face related detectors with much larger datasets. The availability of annotation data [8] increases the number of annotated faces in real situations. In this dataset the authors provide face location information in terms of ellipses.
- The Yale Face database [1] contains a homogeneous collection of face images in different illumination conditions.
- Facity[1] is an online photo project presenting high quality frontal face images with natural illumination, no facial expression and open eyes.

Table 1 summarizes the number of images and faces available in each dataset. CMU and FDDB datasets can contain more than one face per image. The average image size (it is fixed for Yale and Facity sets), the average eye distance (in pixels) of each face annotated and the dataset standard deviation is also provided to indicate the dataset variability.

Excepting the CMU dataset, no other dataset is provided with information related to the facial features, therefore we have roughly annotated the center point of the main facial features: eyes, nose and mouth.

The criterion adopted to consider a facial feature, f_i, detection as correct, is that the euclidean distance between the annotated location, $pos_{f_i,annotated}$, and the detected location, $pos_{f_i,detected}$, must be lower than one fourth the actual eye distance. This criterion was used to estimate the eye detection success originally in [9].

[1] `www.facity.com`

Table 1. Datasets statistics. The average image dimension, average eye distance and standard deviation are expressed in pixels.

Dataset	Number of images	Average image dimension	Number of annotated faces	Average eye distance	Standard deviation
FDDB	2845	377×399	5171	99	177
CMU	180	421×422	721	64	203
Yale	165	320×243	165	55	3.4
Facity	3114	600×600	3114	206	14

4 Experiments

4.1 Classifiers

For each facial feature (eyes, nose and mouth) we have made use of two different training datasets:

- **Set A:** A collection of 6000 heterogeneous images taken randomly from the web. Using this dataset four different classifiers were trained: left eye, right eye, nose and mouth. These classifiers are already included in the current OpenCV release and have been analyzed in [4].
- **Set B:** A subset of 2300 faces of the Facity collection. Using this dataset different classifiers were trained: left eye, right eye, iris, nose, mouth, left mouth corner and right mouth corner.

For both training sets the flipped image was also used for training purposes, therefore we had around 12000 positive samples for the first family and around 4600 for the second. For both configurations around 15000 images were used as negative samples.

4.2 Results

The receiver operating characteristic (ROC) curve of each classifier is computed applying first the original release of each classifier, and two variants reducing its number of stages. Theoretically, this action must increase both correct, D, and false, F, detection rates.

The processing cost and detection precision are reflected in Table 2 for each classifier. The processing cost indicates the total time needed, in seconds, to process the whole dataset in a Core2 PC. The precision is related to the actual eye distance of the face, only for those detections considered true detections. It is observed that the classifiers computed making use of the training set B are much faster. This is justified by the simplicity of the resulting classifiers on each stage, the simpler the training images, the faster the resulting classifier. These classifiers are also similar or slightly more precise than those obtained using the training set A. They are particularly much more precise for images of similar nature than those used to train the classifiers, i.e. Facity.

Table 2. Classifier processing cost in seconds per dataset and positive detection precision, relative to the actual eye distance

Classifier	FDDB		CMU		Yale		Facity	
	Time	Precision	Time	Precision	Time	Precision	Time	Precision
Classifiers trained with set A								
Right eye	550	0.073	46	0.054	14	0.03	1546	0.02
Left eye	563	0.072	47	0.068	15	0.04	1625	0.04
Nose	927	0.069	72	0.11	19	0.04	1990	0.05
Mouth	677	0.17	58	0.11	16	0.06	1672	0.13
Classifiers trained with set B								
Iris	83	0.04	8	-	2	0.04	229	0.009
Left Eye	63	0.01	7	-	1.3	-	185	0.01
Right eye	61	0.09	7	-	1.4	-	174	0.006
Nose	90	0.1	9	0.16	2	0.07	233	0.08
Left mouth	88	-	9	0.05	2	0.04	256	0.02
Right mouth	100	-	10	0.04	2	0.04	289	0.02
Mouth	86	-	9	0.08	2	0.09	231	0.02

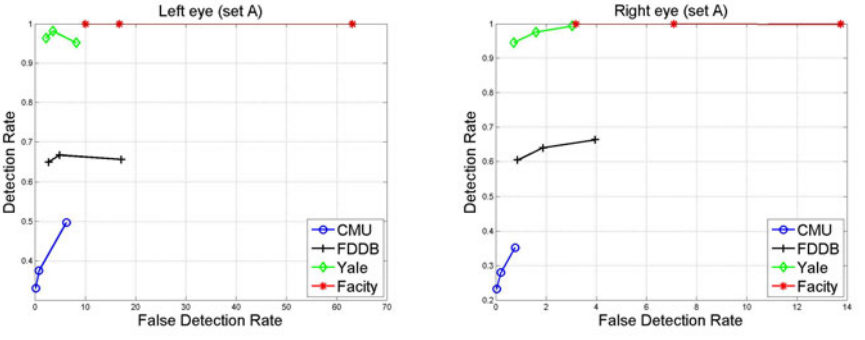

Fig. 1. Performance achieved using the left and right eye detector computed with the training set A

Those classifiers trained using set B present two important characteristics, they are faster, almost ten times for some facial features, and similar or more precise. Unfortunately, they are not so reliable to the whole dataset collection as seen in Figures 1-5. Their respective areas under the ROC curve are smaller than those presented bu the family of classifies computed with set A. To analyze each feature, Figure 1 and 2 compares the detection rate of the two classifiers specialized in the eye detection. The detectors based on the set A perform similarly for both eyes. However they are worst, as expected, for those datasets with unrestricted pose, while being really reliable with the frontal face datasets: Facity and Yale. On the other side, the left eye detector based on the set B offers a poor performance even for the Facity dataset. The iris detector presents better performance, and a reduced false detection rate, but far from that achieved using set A.

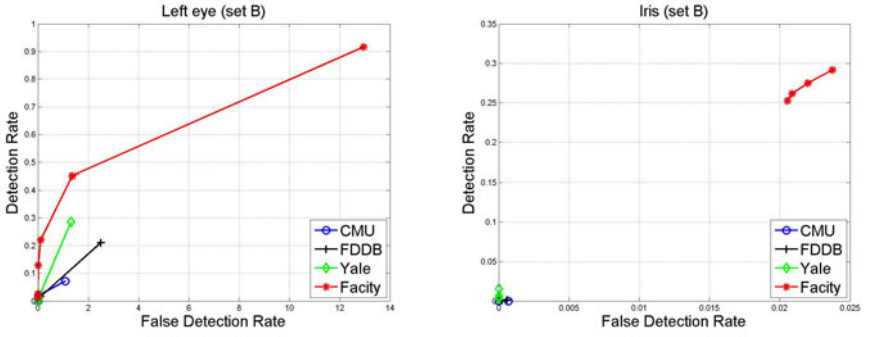

Fig. 2. Left) Left eye detection performance using the training set B. Right) Performance achieved using the iris detector computed with the training set B.

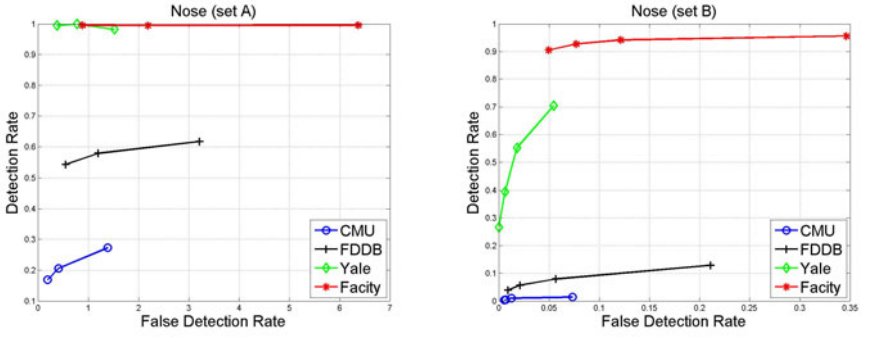

Fig. 3. Left) Nose detection performance using the training set A. Right) Nose detection performance using the training set B.

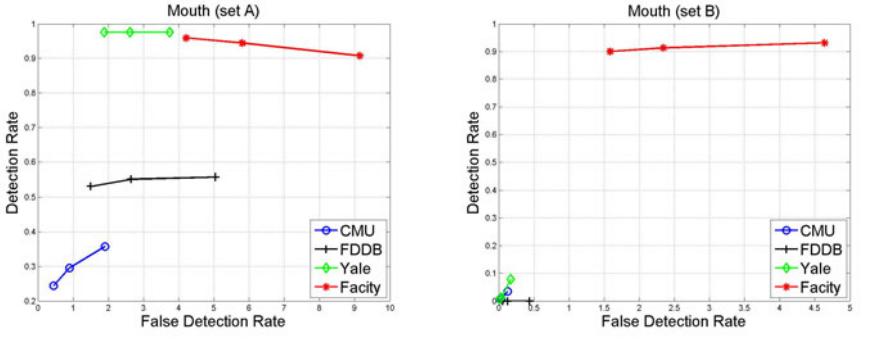

Fig. 4. Left) Mouth detection performance using the training set A. Right) Mouth detection performance using the training set B.

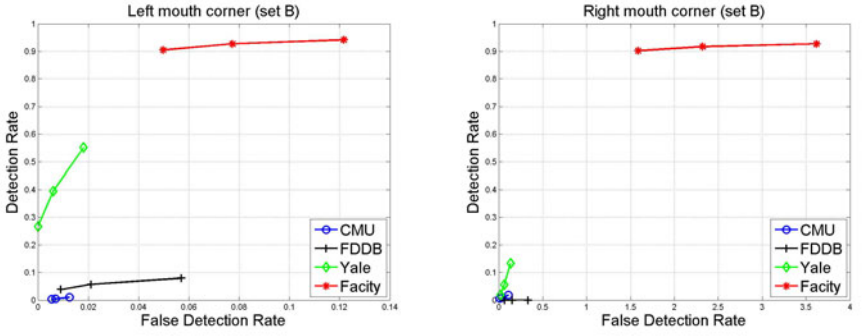

Fig. 5. Mouth corners detection performance using the training set B

The nose detection rates are presented in Figure 3. The behavior for the set A is similar to that observed for the eye pattern. The classifiers obtained with set B are now behaving better for the Yale and Facity but they never reach the reliability exhibited by those trained with set A. However, the reader must remember that this detector is much faster.

A similar performance is observed for the mouth detection, see Figure 4. We have also included the performance of the mouth corner classifiers, see Figure 5. The latter is only sensitive for the Facity dataset.

5 Conclusions

We have trained facial features detectors using two different kind of samples to build the training set. The training set A contains heterogeneous images under uncontrolled conditions, in contrast with the homogeneous training set B.

The results achieved with the training set A are more reliable than those achieved with the training set B. Those classifiers trained with set B are faster (almost ten times), with similar or better precision and present lower false detection rates, but their ROC curves suggest a clearly worse performance. They exhibit close performance only for datasets containing images of similar nature to those used for training. We can conclude that the training set does not encloses enough appearance information to build a robust facial feature detector

For future work we plan to combine the detectors and even the training sets. The effort must be done in terms of speeding up the process while keeping similar performance to those achieved with the training set A.

Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Innovation funds (TIN2008-06068).

References

1. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. on PAMI 19(7), 711–720 (1997)
2. Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D., Rehg, J.M.: On the design of cascades of boosted ensembles for face detection. International Journal of Computer Vision 77, 65–86 (2008)
3. Carnegie Mellon University: CMU/VACS image database: Frontal face images (1999), http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html (last accesed May 11, 2007)
4. Castrillón, M., Déniz, O., Hernández, D., Lorenzo, J.: A comparison of face and facial feature detectors based on the violajones general object detection framework. Machine Vision and Applications (2010) (in press)
5. Hjelmas, E., Low, B.K.: Face detection: A survey. Computer Vision and Image Understanding 83(3), 236–274 (2001), http://dx.doi.org/10.1006/cviu.2001.0921
6. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007)
7. Intel: Intel Open Source Computer Vision Library, v2.1 (April 2010), http://sourceforge.net/projects/opencvlibrary/ (last visited June 2010)
8. Jain, V., Learned-Miller., E.: Fddb: A benchmark for face detection in unconstrained settings. Tech. rep., University of Massachusetts, Amherst (2010)
9. Jesorsky, O., Kirchberg, K.J., Frischholz, R.W.: Robust face detection using the hausdorff distance. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 90–95. Springer, Heidelberg (2001)
10. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: IEEE ICIP 2002, vol. 1, pp. 900–903 (September 2002)
11. Reimondo, A.: Haar cascades repository (2007), http://alereimondo.no-ip.org/OpenCV/34 (last visited April 2010)
12. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(1), 23–38 (1998)
13. Schneiderman, H., Kanade, T.: A statistical method for 3d object detection applied to faces and cars. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1746–1759 (2000)
14. Seo, N.: 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)
15. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(1), 39–51 (1998)
16. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 151–173 (2004)
17. Yang, M.H., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002), http://dx.doi.org/10.1109/34.982883