A Rapidly Trainable and Global Illumination Invariant Object Detection System

Sri-Kaushik Pavani[1,2], David Delgado-Gomez[1,2], and Alejandro F. Frangi[1,2,3,⋆]

[1] Research Group for Computational Imaging & Simulation Technologies in Biomedicine, Universitat Pompeu Fabra, Barcelona, Spain
[2] Networking Research Center on Bioengineering, Biomaterials and Nanomedicine (CIBER-BBN), Spain
[3] Catalan Institution for Research and Advanced Studies (ICREA), Spain
{kaushik.pavani,david.delgado,alejandro.frangi}@upf.edu

Abstract. This paper addresses the main difficulty in adopting Viola-Jones-type object detection systems: their training time. Large training times are the result of having to repeatedly evaluate thousands of Haar-like features (HFs) in a database of object and clutter class images. The proposed object detector is fast to train mainly because of three reasons. Firstly, classifiers that exploit a clutter (non-object) model are used to build the object detector and, hence, they do not need to evaluate clutter images during training. Secondly, the redundant HFs are heuristically pre-eliminated from the feature pool to obtain a small set of independent features. Thirdly, classifiers that have fewer parameters to be optimized are used to build the object detector. As a result, they are faster to train than their traditional counterparts. Apart from faster training, an additional advantage of the proposed detector is that its output is invariant to global illumination changes. Our results indicate that if the object class does not exhibit substantial intra-class variation, then the proposed method can be used to build accurate and real-time object detectors whose training time is in the order of seconds. The quick training and testing speed of the proposed system makes it ideal for use in content-based image retrieval applications.

1 Introduction

Although object detectors based on Haar-like features (HFs) [6] achieve high accuracy rates in real-time [13], training them is a time-consuming task. This is because thousands of weak classifiers based on HFs need to be trained using a database of object and clutter (non-object) images. VJ reported training time in the order of weeks using 180,000 features on a 466 MHz AlphaStation XP900 [13]. Reduced training time of about 2 days using approximately 20,000 features can be achieved using the implementation in the OpenCV library [1] on a 3 GHz

⋆ This work was partially funded by grant TEC2006-03617/TCM, from the Spanish Ministry of Innovation & Science, and grants FIT-360000-2006-55 and FIT-360005-2007-9 from the Spanish Ministry of Industry.

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 877–884, 2009.

processor. Though at first glance, it may seem that two days of training time is affordable, the total algorithmic development time generally exceeds this time frame. Many trials may be required to optimize the performance of the detector, which could prolong the effective development time to months. As McCane and Novins [5] point out, long training times make testing new algorithms or verifying past results extremely difficult.

Several possible approaches have been proposed to reduce the high training time. For instance, Wu *et al.* [14] who achieved a reduction in training time of approximately two orders of magnitude by pre-training weak classifiers before the iterative classifier selection procedure. Stojmenovic [12] proposed to reduce the training time by pre-eliminating HFs from the original training set. They eliminate HFs which produce error greater than a pre-determined threshold value. On a database of images containing back-view of Honda Accord cars, they could eliminate 97% of the original features, thereby achieving a potential speed increase of up to two orders of magnitude. However, it is not clear what percentage of HFs can be removed on more challenging images like those of human frontal faces. Pham *et al.* [7] proposed decreasing the training time by pre-computing the global statistics of face and non-face images. They reported a training time of 5 hours and 30 minutes while achieving high accuracy.

In this work, we propose a novel algorithm that reduces the training time to the order of seconds in a conventional desktop computer with a 3 GHz processor. The high training speed is due to the following three reasons. Firstly, a clutter model is used instead of using clutter class images. This results in a substantial reduction of training time because approximately 10^7 clutter image regions are used for training by traditional training methods. The weak classifiers used in the prosed approach, as will be seen in Section 2.1, implicitly incorporate the clutter model and therefore, the model need not be trained. Secondly, we heuristically pre-eliminate HFs in the feature pool to obtain a set of features that make independent measurements on clutter. Using lesser HFs during training also contributes to the faster training speed. Further, the weak classifiers used in our procedure have fewer parameters to be optimized and therefore, are faster to train.

2 Haar-Like Features and Weak Classifiers

Haar-like features (HFs), shown in Fig. 1, are an over-complete set of two-dimensional Haar functions, which can be used to encode local appearance of

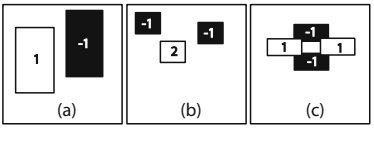

Fig. 1. Typical two-, three- and four-rectangle Haar-like features. The numbers shown on the rectangles refer to the weights assigned to each of them.

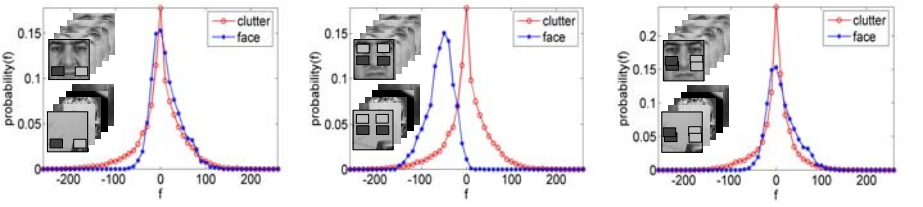

Fig. 2. Three histograms of feature values obtained by evaluating face and clutter class images on HFs are shown. To the left of the histograms, the HFs that were used for evaluation have been super-imposed on a typical training image. As Huang and Mumford [3] observed, the distribution of feature values from clutter images tends to a Laplacian distribution centered at zero.

objects [6]. The feature value f of a Haar-like feature which has k rectangles is obtained as in (1). The quantity $\mu^{(i)}$ is the mean intensity of the pixels in image $\mathbf{x}$ enclosed by the i^{th} rectangle and $w^{(i)}$ is the weight assigned to the i^{th} rectangle. The weights assigned to the rectangles of a HF are set to default numbers satisfying (2). Weak classifiers that label an image $\mathbf{x}$ as object (+1) or clutter (-1) can be expressed as in (3). The quantity $\theta \in \Re$ is a threshold value, and $p \in \{1, -1\}$ can be used to invert the inequality relationship. Training such a weak classifier involves setting appropriate values to its threshold and polarity coefficients (θ^*, p^*) such that the overall error is minimized. Formally, $[\theta^*, p^*] = \arg\min_{[\theta,p]} \sum_{i=0}^{n_o+n_c} \epsilon^{(i)}$. If a training image is correctly classified, then its error is $z^{(i)}$, else it is 0. The term $z^{(i)}$ is the weight assigned to the training image $\mathbf{x}^{(i)}$. The quantities n_o and n_c are the number of object and clutter class training images, respectively. Training the weak classifiers as in (3) can be intuitively understood from Fig. 2. For each HF shown in Fig. 2, histograms of the feature value, f, have been obtained from object (human frontal face) and clutter training images. During training, θ is set to the value of f that best separates object and clutter examples.

$$f = \sum_{i=1}^{k} w^{(i)} \cdot \mu^{(i)} \qquad (1)$$

$$\sum_{i=1}^{k} w^{(i)} = 0 \qquad (2)$$

$$h(\mathbf{x}) = \begin{cases} +1, f_{(\theta,p)} > 0 \\ -1, \text{otherwise} \end{cases} \qquad (3)$$

$$f_{(\theta,p)} = (f - \theta) \cdot p \qquad (4)$$

2.1 A Clutter Model

When a HF is evaluated on a clutter image, the expectation value of the output can be expressed as in (5).

$$E(f) = E\left(\sum_{i=1}^{k} w^{(i)}\mu^{(i)}\right) = \sum_{i=1}^{k} w^{(i)} E\left(\mu^{(i)}\right) \tag{5}$$

The clutter class, being generic, may contain any image with any appearance pattern. Effectively, every pixel of a generic clutter image is a random variable which can take any value between the minimum and the maximum permitted pixel values in an image representation (N_{min} and N_{max}) with equal probability. For example, in gray-level images, $N_{min} = 0$ and $N_{max} = 255$. Therefore, the expected value of mean of pixel values within any rectangular region, $E(\mu) = 0.5(N_{max} + N_{min})$. Rewriting (5) using (2), we get (6).

$$E(f) = 0.5(N_{max} + N_{min}) \sum_{i=1}^{k} w^{(i)} = 0 \tag{6}$$

Therefore, the probability that the feature value of a HF on a clutter image to be greater than (or lesser than) 0 is 0.5. Mathematically, $\mathbb{P}(f \cdot p > 0 | \mathbf{x}^{(i)} \in \texttt{Clutter}) = 0.5$. Using the terminology introduced in (4),

$$\mathbb{P}(f_{(0,p)} > 0 | \mathbf{x}^{(i)} \in \texttt{Clutter}) = 0.5 \tag{7}$$

The clutter model in (7) can be observed from the clutter histograms shown in Fig. 2. Note that the clutter histograms are all symmetric and centered at $f = 0$.

2.2 Proposed Weak Classifier

The proposed weak classifier utilizes the clutter model in (7) by setting its threshold $\theta = 0$ so that it labels 50% of the clutter correctly. Since θ is already set, training the proposed weak classifier only involves setting an appropriate value to the polarity term (p^*) such that the training error is minimized as shown in (9). As θ need not be optimized, the training speed of the weak classifiers is much higher than the traditional ones as in (3).

$$h(\mathbf{x}) = \begin{cases} +1, & f_{(0,p)} > 0 \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

$$p^* = \arg \min_{p \in \{1,-1\}} \sum_{i=0}^{n_o} \epsilon^{(i)} \tag{9}$$

The object detectors are built by arranging weak classifiers as in (8) according to the rejection cascade architecture [2]. This architecture has been preferred for building object detectors as it is conducive for fast scanning of an image [13]. A rejection cascade, as illustrated in Fig. 3, consists of multiple nodes connected in series. Each node is a binary classifier that classifies an input sub-region as object or clutter. Each node consists of multiple weak classifiers which are

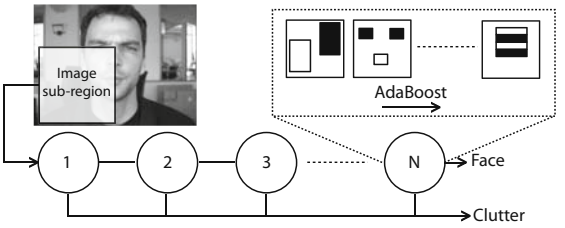

Fig. 3. A cascaded classifier consists of multiple nodes arranged in a degenerated decision tree fashion. An input image is scanned at different scales and positions for the presence of a face. If an image sub-region is classified as a face by all the sub-regions of the face, then it is labeled a face.

selected iteratively using the AdaBoost procedure [10]. The weighted decision of all the weak classifiers in a node is output as the decision of the node.

2.3 Pre-eliminating Redundant HFs

As mentioned before, HFs are an over-complete set of features, therefore, they are redundant. Conventional object detectors avoid selecting redundant features in different nodes by training each node with bootstrapped set of clutter images [13]. In other words, features selected for different nodes are suitable for classifying different subsets of clutter images. In our case, since clutter images are not used, the over-complete set of HFs need to be pruned heuristically after each node is built so that neither the previously selected features nor similar ones are selected again. Similarity between two HFs is measured by the amount of overlap between its rectangles. For example, the HFs illustrated in Fig. 2(left) and Fig. 2(middle) do not overlap at all, therefore, they are considered to make independent measurements on a clutter image. On the contrary, the HFs illustrated in Fig. 2(middle) and Fig. 2(right) have more than 50% overlap, and therefore they are considered to make redundant measurements. To build the proposed object detector, we generated a feature pool with $7,200$ HFs in which no HF in the feature pool has more than 50% overlap with the rest of the features.

3 Experimental Setup and Results

The proposed weak classifiers described above were trained for two very different object detection problems: detection of human frontal faces in photographs and detection of the human heart in short-axis cardiac Magnetic Resonance Images (MRI). For this purpose, two object databases (`face` and `heart`) were used. The `face` database was composed of 5000 images. The faces in this database exhibit an out-of-plane rotation of up to $\pm 10^\circ$ and various expressions. The `heart` database consisted of 493 short-axis MR heart images. In comparison to the images in the `face` database, the images in the `heart` database exhibit less

Table 1. Comparison of training time

Method	Number of features in the feature pool	Number of classifiers trained	Number of object images used	CPU speed (GHz)	Training time
Proposed (Face)*	7,800	3,200	5,000	3.0	96s
VJ [13]	40,000	4,297	9,500	0.4	weeks
LZZBZS [4]	n/a	6,000	2,546	0.7	weeks
WBMR [14]	40,000	3,870	5,000	2.8	13h20m
PC [7]	295,920	3,502	5,000	2.8	5h30m
Proposed (Heart)*	7,800	1,000	493	3.0	30s
VJ (Heart)*	180,000	300	493	3.0	22h

* Results from our implementation.

Table 2. True positive rate in simulated test datasets

Method	DS1[a]	DS2[b]	DS3[c]	DS4[d]	DS5[e]	DS6[f]	DS7[g]	DS8[h]
								
Proposed (Face)*†	88.0	87.4	86.5	88.0	88.0	88.0	88.0	84.7
VJ (Face)*	90.3	90.3	90.3	87.4	86.2	87.0	83.9	80.2
VJ (Face)*†	90.3	85.1	72.5	87.4	78.7	84.0	81.2	60.5
								
Proposed (Heart)*†	97.3	97.3	94.6	97.3	97.3	97.3	96.7	93.8
VJ (Heart)*	98.7	98.7	98.7	90.3	85.2	96.8	93.0	76.3
VJ (Heart)*†	98.7	81.2	63.2	90.3	20.3	69.6	35.2	0.0

* Results from our implementation. † Results without variance normalization.
[a] DS1: Original test images. [b] DS2: Intensity values are globally divided by 2.
[c] DS3: Intensity values are globally divided by 3. [d] DS4: Histogram equalized images.
[e] DS5: Gamma corrected image ($\gamma = 0.8$). [f] DS6: Gamma corrected image ($\gamma = 0.9$).
[g] DS7: Gamma corrected image ($\gamma = 1.1$). [h] DS8: Gamma corrected image ($\gamma = 1.2$).

intra-class appearance variation. The face detectors were tested on MIT+CMU frontal face database [9]. The heart detectors were tested on a set of 293 images. The speed of training of the face and heart detectors, in comparison to other methods, is tabulated in Table 1.

We tested the accuracy of the object detectors by transforming the test images artificially to simulate global illumination changes. On each of the transformed database, the accuracy of the VJ-type detector and the proposed method were measured and the results are tabulated in Table 2. We observed that, in

Table 3. Comparison of accuracy of the face and heart detectors

Method	Face		Heart	
	FD [a]	TPR [b]	FD [a]	TPR [b]
Proposed	912*	88.0*	2*	97.3*
VJ [13]	95	90.8	2*	98.7*
LZZBZS [4]	90	92.5	n/a	n/a
WBMR [14]	85	92.5	n/a	n/a
PC [7]	100	90.0	n/a	n/a
RBK [9]✠	95	89.2	n/a	n/a
SK [11]✠	65	94.5	n/a	n/a
RYA [8]✠	78	94.8	n/a	n/a

[a] FD: Number of false detections. [b] TPR: True positive rate.
* Results from our implementation. ✠ Methods not based on HFs.

contrast to VJ detector, the proposed detector performed consistently to all the monotonic image transformations applied to the test images. This is because, the detector uses weak classifiers that make decision based on the sign of the feature value of a HF, and not based on the magnitude of the feature value of the HF. In theory, the accuracy of the proposed detector should not change if any monotonic transformations are applied to images. However, we see that the accuracy decreases in DS3 and DS8. This is because, two image patches (with different original intensities) might end up have the same average intensities after image transformation, and therefore, not satisfy (3) because of saturation of intensity values (as in the case of DS8) or because of rounding errors in the division process (as in the case of DS3). The results of the VJ-type detector with and without variance normalization are also tabulated in Table 2. The proposed detector does not require variance normalization procedure as the sign of the feature value of any HF is not affected by the variance normalization process. Thus, the computation of the integral image square and the computation of standard deviation of each image sub-region can be avoided during the detection process, which adds to the speed of detection. The time required to process all the images in the test set by the face and the heart detectors were 41s and 12s. This includes the time to read the image, computation of integral image(s), the scanning, and the clustering process to merge multiple detections. Our implementation of VJ procedure (with variance normalization) took 62s and 14s, respectively. The testing times were measured on a 3 GHz CPU.

The number of false detection by the the face and the heart detectors, along with the state-of-the-art methods, is listed in Table 3. The face detector achieved a false positive rate of 9.2×10^{-5} (912 false detections), which is approximately 10 times worse than the state-of-the-art detectors. However, the number of false detections by the heart detector was only 2, which represented a false positive rate of 3.3×10^{-6}.

4 Conclusions

In this paper, we have presented a novel training procedure for object detection systems and compared its performance, both during training and testing phases, with the state-of-the-art techniques. The advantages of adopting proposed technique include fast training in the order of seconds, global illumination invariance and real-time detection speed. The disadvantage of this method is that it produces more false positives with respect to the state-of-the-art.

The quick training and testing speed of the proposed technique makes it ideal for content based image retrieval systems - where a user makes a query (an image patch), and asks the system to automatically find similar patches in a huge database of images. The existing methods, by the virtue of being slow to train, cannot be used in such scenarios.

References

1. OpenCV library, `http://sourceforge.net/projects/opencvlibrary/`
2. Baker, S., Nayar, S.K.: Pattern rejection. In: CVPR 1996, pp. 544–549 (1996)
3. Huang, J., Mumford, D.: Statistics of natural images and models. In: CVPR 1999, pp. 541–547 (1999)
4. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.: Statistical learning of multi-view face detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
5. McCane, B., Novins, K.: On training cascade face detectors. In: IVCNZ 2003, pp. 239–244 (2003)
6. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: ICCV 1998, pp. 555–562 (1998)
7. Pham, M.-T., Cham, T.-J.: Fast training and selection of Haar features using statistics in boosting-based face detection. In: ICCV 2007, pp. 1–7 (2007)
8. Roth, D., Yang, M., Ahuja, N.: A SNoW-based face detector. In: NIPS 2000, pp. 855–861 (2000)
9. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE TPAMI 20(1), 23–38 (1998)
10. Schapire, R.E.: A brief introduction to boosting. In: IJCAI 1999, pp. 1401–1406 (1999)
11. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: CVPR 2000, pp. 746–751 (2000)
12. Stojmenovic, M.: Pre-eliminating features for fast training in real time object detection in images with a novel variant of AdaBoost. In: CIS 2006, pp. 1–6 (2006)
13. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)
14. Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE TPAMI 30(3), 369–382 (2008)