

A Promising technique for feature extraction and detection of faces using Sparsity - Enforcing Method

Banurekha.K.M
Department of MCA,
AVC College of Engineering,
Mayiladuthurai, India.

Nelgadevi.M
Department of CS,
Rani Anna Govt. College,
Tirunelveli, India.

Vijayaselvarani.S
Department of CSE,
Bhajarang Engineering College,
Chennai,India

Abstract: This paper proposes a sparsity enforcing based face detection with three kinds of classifiers. The feature extraction is done on the basis of rectangular feature from the integral image of the face. Sparsity enforcing method is applied for the dimension reduction on rectangular features from the generated Integral image. The performance of the classifier such as Support Vector Machines, Bayesian and AdaBoost are analyzed with the proper testing procedure. The analysis is also made, how the classifier performs according to the training sample size. Experimental results on three widely used face databases are presented to demonstrate the efficiency of the proposed approach.

Keywords: Support Vector Machines, Sparsity – Enforcing, Bayesian, AdaBoost

1. INTRODUCTION

The rapidly expanding research in face processing is based on the premise that information about a user's identity, state, and intent can be extracted from images, and that computers can then react accordingly. In the last five years, face and facial expression recognition have attracted much attention though they have been studied for more than 20 years by psychophysicists, neuroscientists, and engineers. Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. The goal of facial feature detection is to detect the presence and location of features, such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc., with the assumption that there is only one face in an image. It is worth mentioning that many papers use the term "face detection," but the methods and the experimental results only show that a single face is localized in an input image.

While numerous methods have been proposed to detect faces in a single image of intensity or color images, many were unaware of any surveys on this particular topic. [8] discuss that in general Face recognition techniques can be broadly divided into three categories based on the face data acquisition methodology: methods that operate on intensity images; those that deal with video sequences; and those that require other sensory data such as 3D information or infra-red imagery. Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. Since these data

driven methods rely heavily on the training sets, this paper also discuss several databases suitable for this task. The classes of face and nonface images are decidedly characterized by multimodal distribution functions and effective decision boundaries are likely to be nonlinear in the image space. S.T.Gandhe,et.al have disussed about the Principal Component Analysis, Discrete Wavelet Transform Cascaded with Principal Component Analysis, Contour Matching and Isodensity Line Maps Cascaded with Hopfield Neural Network in [11].M. Meenakshi has proposed a Digital Signal Processor (DSP)-based Prototype facial recognition and verification system in [2]. In [3] Siba Shankar Rout confirms that fixed point implementation (Integer Wavelet Transform) is best candidate than floating point implementation (Classical Wavelet Transform).A Combination of PCA-SVM (Principle component analysis-Support Vector Machine) and GA (Genetic Algorithm) is proposed by Rakesh Kumar,et.al in [4]. Artificial Neural Network is been tested for face Recognition System by S.Adebayo, et.al in [1]. In [17] A non-linear support vector machine is used to determine whether or not a face is contained within the observation window.[6][12][7] talks about Feature Selection via Sparse Approximation.[5] talks about a two-in-one portable low-cost dsPIC30F3013 digital signal processing microcontroller based system for real time face recognition and signature verification. A Survey is made on Detecting Faces in Images regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is analyzed in [14].To handle profile views

and rotated faces, a decision tree is trained to determine the viewpoint class (such as right profile or rotated 60 degrees) for a given window of the image being examined in [13][9]. Asymmetric AdaBoost and a Detector Cascade for fast face detection is used in domains where the distribution of positive and negative examples is highly skewed. This is been proposed by Paul Viola and Michael Jones in [16]. Image segmentation based on skin color, features extracted from the two dimensional discrete cosine transform (DCT), and self-organizing maps (SOM) is proposed in [10],[15] deals with Feature based face detection using Gaussian derivative filters. To be effective, either classifiers must be able to extrapolate from a modest number of training samples or be efficient when dealing with a very large number of these high-dimensional training samples.

2. Face Extraction and Detection

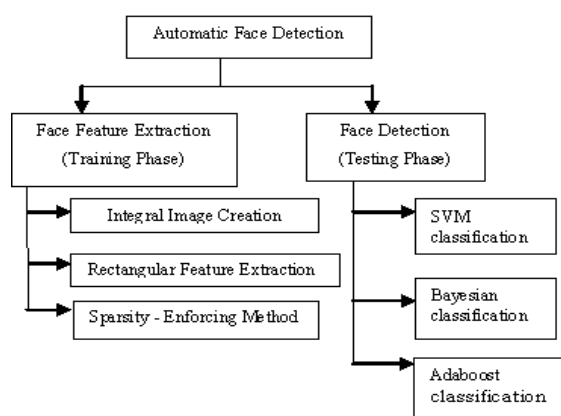


Figure. 1 Architecture Design of face extraction and detection

2.1 Integral Image

Using the integral image any rectangular sum can be computed in four array references. Clearly the difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

Rectangle features can be computed very rapidly using an intermediate representation for the image which is the integral image. The integral image at location x, y contains the sum of the pixels above and to the left of x, y , inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

(where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$) the integral image can be computed in one pass over the original image.

Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

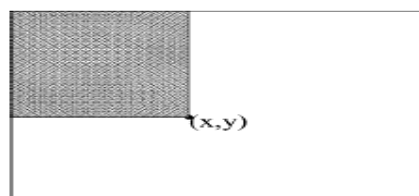


Figure 2: The value of the integral image at point (x, y) is the sum of all the pixels above and to the left.

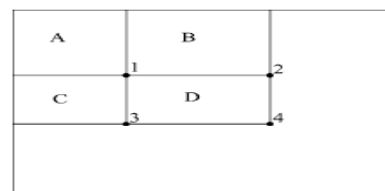


Figure 3: The sum of the pixels within rectangle D can be computed with four array references.

The value of the integral image at location 1 is the sum of the pixels in rectangle A .

The value at location 2 is $A + B$, at location 3 is $A + C$, and at location 4 is $A + B + C + D$.

The sum within D can be computed as $4 + 1 - (2 + 3)$.

In the case of convolution, if the derivative operator is applied both to the image and the kernel the result must then be double integrated:

$$f * g = \iint (f' * g')$$

Convolution can be significantly accelerated if the derivatives of f and g are sparse (or can be made so). A similar insight is that an invertible linear operation can be applied to f if its inverse is applied to g :

$$(f') * \left(\iint g \right) = f * g$$

Viewed in this framework computation of the rectangle sum can be expressed as a dot product,

i-r

where i is the image and r is the box car image (with value 1 within the rectangle of interest and 0 outside). This operation can be rewritten

$$i * r = \left(\iint i \right) \cdot r^N$$

The integral image is in fact the double integral of the image (first along rows and then along columns). The second derivative of the rectangle (first in row and then in column) yields four delta functions at the corners of the rectangle. Evaluation of the second dot product is accomplished with four array accesses.

2.2 Sparsity Feature Extraction

The number of features is much larger than the dimension of the training set, so that the system is hugely under-determined. Because of the redundancy of the feature set, the collinearities responsible for severe ill-conditioning is considered. Both difficulties call for some form of regularization and can be obviated by turning the problem. A Function is Sparse if most coefficients is zero.

The face detection procedure classifies images based on the value of simple features. There are many motivations for using features rather than the pixels directly. The most common reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. For this system there is also a second critical motivation for features: the feature-based system operates much faster than a pixel-based system.

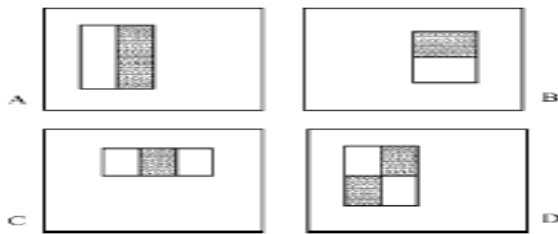


Figure 4: Example rectangle features shown relative to the enclosing detection window.

The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-

rectangle feature. In this section, description is made on the basic algorithm on which our feature selection method is built upon. Restriction is done in the case of a linear dependence between input and output data, which means that the problem can be reformulated as the solution of the following linear system of equations:

$$g = Af \quad (1)$$

where $g=(g_1, \dots, g_n)^T$ is the $n \times 1$ vector containing output labels, $A=\{A_{ij}\}, i=1, 2, \dots, n;$

$j=1, 2, \dots, p$ is the $n \times p$ matrix containing the features j for each image i and $f = (f_1, \dots, f_p)^T$ the vector of the

unknown weights to be estimated. Since in the present context the dimensions of A are large, even if $n = p$, the usual approaches for solving the algebraic system turn out to be unfeasible. Moreover, typically the number of features p is much larger than the dimension n of the training set, so that the system is hugely under-determined. Classical regularization such as the so-called ridge regression (also referred to as Tikhonov's regularization) uses a quadratic penalty, typically the L^2 -norm of the vector f :

$$\|f\|_2 = \sum_j |f_j|^2$$

Such quadratic penalties, however, do not provide feature selection in the sense that the solution of the resulting penalized least-squares problem will typically yield a vector f with all weights f_j different from zero. This is the reason why the replacement of quadratic penalties by sparsity-enforcing penalties has been advocated in recent literature; such penalty should enforce automatically the presence of (many) zero weights in the vector f . The L^1 norm of f , which is the sum of the absolute values of the weights, i.e., $\|f\|_1 = \sum_j |f_j|$ is a suitable penalty in order to enforce sparsity while preserving convexity of the optimization problem. Hence, instead of (1), the following penalized least-squares problem is solved.

$$f_L = \arg \min_f \{ \|g - Af\|_2^2 + 2\tau \|f\|_1 \} \quad (2)$$

where T is a positive parameter regulating the balance between the data misfit and the penalty (the so-called regularization parameter).

In variable selection problems, this parameter also allows to vary the degree of sparsity (number of true zero weights) of the vector f . After problem (2) is usually referred to as "Lasso regression" (an acronym for Least Absolute Shrinkage and Selection Operator). Whereas ridge regression

solutions depend linearly on the output vector g , the L^1 -norm penalty makes the dependence of the lasso solutions on nonlinear. Hence, the computation of L^1 -norm penalized solutions is more difficult than with L^2 -norm penalties. In the special case where $A^T A$ is the identity matrix the solution of (2) is easily seen to be $f_L = S_\tau(A^T g)$ where S_τ is the following “soft-thresholder,” widely used in wavelet-based denoising schemes, and acting component wise on a vector h

$$(S_\tau h)_j = S_\tau h_j = \begin{cases} h_j - \tau \operatorname{sgn}(h_j), & \text{if } |h_j| \geq \tau \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Hence, the parameter appears as a threshold value, under which a component is set to zero. The number of selected features (nonzero weights) is controlled by the threshold T .

When $A^T A$ is different from the identity, there is no longer a closed-form expression for the minimizer (2) and several numerical strategies have been proposed in the literature to solve the corresponding and rather cumbersome nonlinear optimization problem. In this paper a simple iterative strategy, namely the following scheme is adopted

$$f_L^{(t+1)} = S_\tau[f_L^{(t)} + A^T(g - Af_L^{(t)})] \quad t = 0, 1, \dots \quad (4)$$

with an arbitrary initial vector $f_L^{(0)}$.

In the absence of soft-thresholding (or equivalently for $T=0$), this scheme is sometimes referred to as the Landweber iteration and converges to the generalized solution (minimum-norm least-squares solution) of (1).

Algorithm Specialization: It starts by considering the problem in the form (1) where A is the n by p matrix of the processed image data, the entry A_{ij} representing the j^{th} rectangle feature obtained from the image labeled by i . The data matrix is manipulated in order to center the features values around their median and then normalized by dividing its entries by a number slightly larger than its largest eigen value (this is done in order to guarantee convergence of the iterative scheme).

Since working in a binary classification setting this paper associate to each image a label $g_i \in \{-1, 1\}$. Each entry of the unknown vector f is associated to one feature.

Performing feature selection is equivalent to looking for a

sparse solution $f = (f_1, \dots, f_p)^T$ of (1), i.e., for a

weight vector that has many zero entries. This is done through the use of the iterative algorithm (4). Features corresponding to nonzero weights f_i are retained as relevant to discriminate between the two classes. After checking empirically that the choice of the initialization vector was not a crucial issue, it chose to always initialize the weight vector f with zeros:

$$f^{(0)} = 0^T$$

Once the matrix is correctly prepared, the iterative scheme is very simple to implement, which provides a pseudo-code describing the matrix construction, centering and normalization, and the iterative scheme (4). The following stopping rule are used for the iteration according to the stability of the solution reached: at the t th iteration it evaluate

$$\|f^{(t)} - f^{(t-1)}\|_2$$

if this quantity is smaller than some prescribed threshold T (which it choose to be a fraction of the value of the norm

$$\|f^{(t)}\|_2, T = \|f^{(t)}\|_2/100)$$

during 100 consecutive iterations, then the obtained solution is stable and concluded to stop the iterative process.

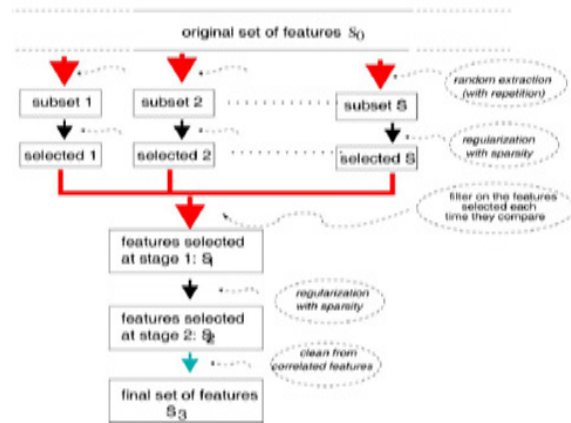


Figure 5: Flow of feature sets from the original description to the final set (dashed balloons indicate the different filtering operations performed).

A different approach was investigated and implemented, which is based on a resampling of the features set aiming at splitting the original problem into many problems of smaller size. It consists of building feature subsets by extracting each time m features from the original

set of size $p(m \ll p)$, thus obtaining smaller linear subproblems of the type

$$A_s f_s = g \quad s = 1, \dots, S$$

where A_s is a submatrix of A containing the columns relative to the features in s ; f_s is defined accordingly. The number of subproblems and their size are needed to choose. Each time extraction is done with replacement a subset of the original features. The subset size should be big enough to be descriptive, but small enough to produce an easy-to-handle matrix.

The Feature Selection Procedure

- Step 1: Find the face features for each image in the gallery.
- Step 2: Label each image (feature set) with proper class label.
- Step 3: For each image feature set, find the median value which is normalized by the iterative approach.
- Step 4: Find highest eigen values from the feature set as λ_{\max} .
- Step 5: Normalize the feature set as well as class label with λ_{\max} .
- Step 6: Apply threshold landweber for the finalized feature to represent face or non face using iterative threshold mode with the help of $f = f + A'(g - a * f)$.
- Step 6.1: Check if feature \geq threshold if so assign

$$\text{feature} = \text{feature} - \text{threshold} * \text{sign}(\text{feature})$$

otherwise set the the feature as null.

In **Feature Storing** process, the trained features are stored.

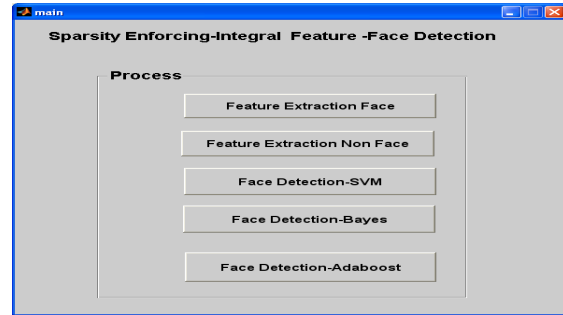


Figure 6: Overall process of feature extraction and face detection

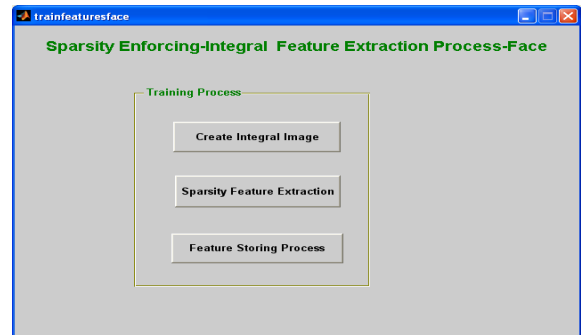


Figure 7: Feature extraction process on faces

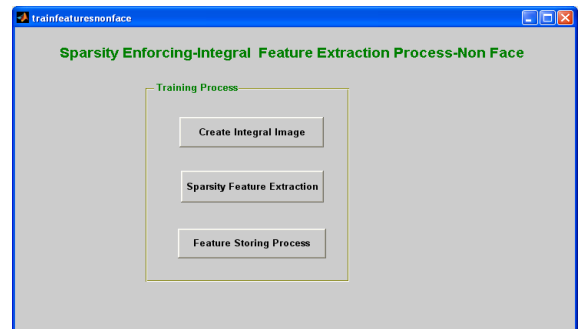


Figure 8: Feature extraction process on non-faces

2.3 Face Detection Using SVM Classifier

It is a binary classification method that finds the optimal linear decision surface based on the concept of structural risk minimization. The decision surface is a weighted combination of elements of the training set. These elements are called support vectors and characterize the boundary between the two classes. Given a set of N examples:

$$(x_1, y_1), \dots, (x_i, y_i), \dots, (x_N, y_N) \quad x_i \in \mathbb{R}^N, y_i \in \{-1, 1\}$$

In case of linear separable data, maximum margin classification aims to separate two classes with hyperplane that maximizes distance of supports vectors. This hyperplane is called OSH (Optimal Separating Hyperplane). OSH can be expressed as in equation:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (x_i^T x) + b$$



Figure 9: Face detection using SVM classifier

2.4 Face Detection Using Bayesian Classifier

Consider each data instance to be an n-dimensional vector of attribute values:

$$X = (x_1, x_2, x_3, \dots, x_n) \quad (1)$$

In a Bayesian classifier which assigns each data instance to one of m classes C_1, C_2, \dots, C_m , a data instance X is assigned to the class for which it has the highest posterior probability conditioned on X, *i.e.* the class which is most probable given the prior probabilities of the classes and the data X. That is to say, X is assigned to class C_i if and only if $P(C_i|X) > P(C_j|X)$ (2)

For the data, $m = 2$, since there are two classes. Bayes Theorem says,

$$P(C_i|X) = P(X|C_i)P(C_i)P(X) \quad (3)$$

Since $P(X)$ is a normalizing factor which is equal for all classes, only maximise the numerator $P(X|C_i)P(C_i)$ in order to do the classification. Both the values can be estimated. $P(X|C_i)$ and $P(C_i)$ are needed from the data, for building the classifier.

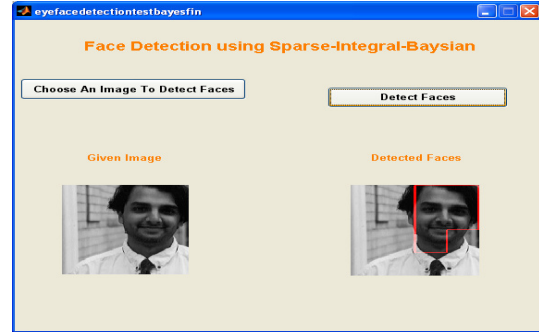


Figure 10: Face detection using Bayesian Classifier

2.5 Face Detection Using Adaboost (AB) Classifier

AB is a linear classifier with all its desirable properties. Its output converges to the logarithm of likelihood ratio. It has good generalization properties. It is a feature selector with a principled strategy (minimisation of upper bound on empirical error).

AB close to sequential decision making (it produces a sequence of gradually more complex classifiers).

AdaBoost is an algorithm for constructing a "strong" classifier as linear combination

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

of "simple" "weak" classifiers

2.5.1 Terminology

$h_t(x)$ is "weak" or basis classifier, hypothesis, "feature"

$H(x) = \text{sign}(f(x))$ is "strong" or final classifier/hypothesis

The $h_t(x)$ can be thought of as features. Often (typically) the set $H = \{h(x)\}$ is infinite.

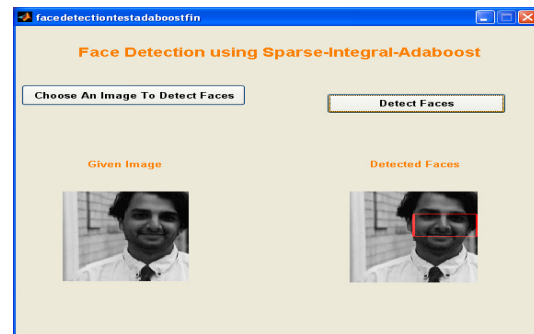


Figure 11: Face detection using Adaboost Classifier

3. RESULT ANALYSIS

The facial images from the database MIT-CMU dataset are used. From that, a training set for faces as well as for non faces of size 32 x 32 was developed. The sample faces and non faces for feature extraction in training section are shown below

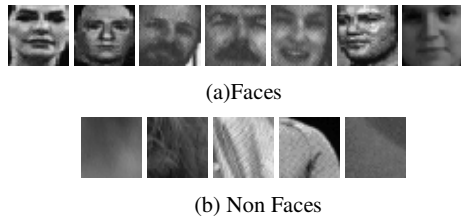
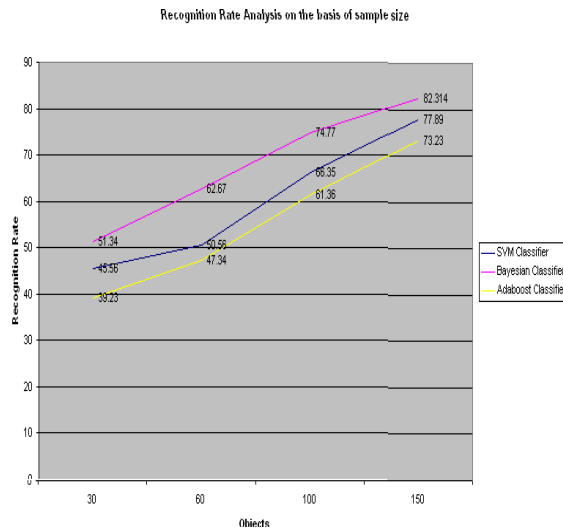


Figure 12: Sample faces and non faces for training section

3.1 Analysis on Detection Rate

From the below result, the approach of sparsity enforcing rectangular feature extraction on the basis of Bayesian classifier performs well than up to 4.424%. Bayesian classifier performs well on low and high resolution images. The performance also depends on the size of feature vector for training the detection rate which increases along the sample size and the error rate analysis is also properly maintained according to the sample size.



The performance of the Face detection using Sparsity-Enforcing Method for Learning Face Features is estimated with three modes of classifiers.

Table 1: Analysis on Detection Rate using Adaboost, SVM and Bayesian Classifier

Approaches	Classification Rate	Error Rate
Sparsity-Enforcing Approach on Rectangular Feature and Adaboost Classifier	74.21%	0.39
Sparsity-Enforcing Approach on Rectangular Feature and SVM Classifier	77.89%	0.33
Sparsity-Enforcing Approach on Rectangular Feature and Bayesian Classifier	82.314%	0.20

Table 2. Performance analysis on Face Detection using Adaboost, SVM and Bayesian Classifier with different Sample Size

Approaches	30	60	100	150
Sparsity-Enforcing Approach on Rectangular Feature and Adaboost Classifier	39.23	47.34	61.36	73.23
Sparsity-Enforcing Approach on Rectangular Feature and SVM Classifier	45.56	50.56	66.35	77.89
Sparsity-Enforcing Approach on Rectangular Feature and Bayesian Classifier	51.34	62.67	74.77	82.31

4. CONCLUSION

In this paper, the process of face detection using sparsity enforcing approach on face features with three kinds of analyzer are used to evaluate its performance on classification accuracy. The process of detecting face is done through three stages. In the first stage, for computational reasons, to split a big problem into a number of smaller problems involving randomized blocks of features was proposed in order to select a smaller but still highly redundant set of features; this stage could be further optimized by exploiting parallel computation. The second stage is based on applying again the selection algorithm to the features resulting from the first stage. The third stage is based on finding a very small set of uncorrelated features that is suitable for real-time processing to the price of a very limited decrease in performance. The latter description on a real-time face detection system was tested based on a cascade of SVM, Bayesian and AdaBoost classifiers. Texture features based detection and detection on the skin color based approach for the color image analysis can also be an enhancement to this work.

5. REFERENCES

- [1] S.Adebayo,Daramola,O.Sandra Odeghe,“Efficient Face Recognition System using Artificial Neural Network” , International Journal of Computer Applications Volume 41, No.21, 2012.
- [2] M. Meenakshi,“Real-Time Facial Recognition System Design, Implementation and Validation”, Journal of Signal Processing Theory and Applications, vol.1, pg: 1-18,2012.
- [3] Siba Shankar Rout,“Integer Wavelet-based PCA for Face Recognition”,International Journal of Computer Applications ,Volume 54, No.7, 2012.
- [4] Rakesh Kumar Yadav,A. K. Sachan,D. Rai,“Research and Applications of Optimal Face Recognition System”, International Journal of Computer Applications ,Volume 44, No.3, 2012.
- [5] Ibiyemi,T.S, Ogunskin,J, Daramola S.,“A Bi-Modal Biometric Authentication by Face Recognition and Signature Verification”,International Journal of Computer Applications,Volume 42,No.20,2012.
- [6] Yixiong Liang, Lei Wang,Yao Xiang, and Beiji Zou ,“Feature Selection via Sparse Approximation for Face Recognition”,IEEE TRANS. PATTERN ANAL. MACH. INTELL., JULY 2010.
- [7] Augusto Destrero,Christine De Mol,Francesca Odone, and Alessandro Verri,“A Sparsity- Enforcing Method For Learning Face Features”2009.
- [8] Rabia Jafri and Hamid R. Arabnia,“A Survey of Face Recognition Techniques”,Journal of Information Processing Systems,Vol.5, No.2, June 2009.
- [9] Margarita Osadchy,Yann Le Cun,Matthew L. Miller ,“Synergistic Face Detection and Pose Estimation with Energy-Based Models”2007.
- [10] Abdallah S. Abdallah, A. Lynn Abbott, and Mohamad Abou El-Nasr ,“A New Face Detection Technique using 2D DCT and Self Organizing Feature Map”2007.
- [11] S.T.Gandhe, K.T.Talele, A.G.Keskar,“Intelligent Face Recognition Techniques: A Comparative Study” International Journal on Graphics,Visio and Image Processing ,Volume 7, Issue 2, August, 2007.
- [12] Allen Y. Yang, John Wright,Yi Ma, and S. Shankar Sastry,“ Feature Selection in Face Recognition: A Sparse Representation Perspective”IEEE TRANS. PAMI, AUGUST 2007.
- [13] Jones M, Viola P , “Fast Multi-view Face Detection”, 2003.
- [14] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, “Detecting Faces in Images: A Survey” IEEE transactions on pattern analysis and machine intelligence, VOL. 24, NO. 1,2002.
- [15] Feature based face detection using Gaussian derivative filters, 2002.
- [16] Paul Viola and Michael Jones, “Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade”, 2002.
- [17] Sami Romdhani, Philip Torr,Bernhard Scholkopf, “Andrew Blake Computationally Efficient Face Detection”, 2001.