ISSN:2249-5789
KesavaRao Seerapu et al , International Journal of Computer Science & Communication Networks,Vol 2(5), 584-589

Face Recognition using Robust PCA and Radial Basis Function Network

KesavaRao Seerapu[1] R. Srinivas[2]

[1](M.Tech Student, Dept. of CSE, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh
[2](Associate Professor, Dept. of CSE, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh

***Abstract*— Face detection and recognition is challenging due to the Wide variety of faces and the complexity of noises and image backgrounds. In this paper, we propose a neural network based novel method for face recognition in cluttered and noisy images. We use a Modified radial basis function network (RBFN) to distinguish between face patterns and non face patterns. The complexity RBFN is reduced by RobustPCA as it gives good results even in different illumination environments and highly un-susceptible to occlusion when compared with Classical PCA (Principal component analysis). RobustPCA is applied on Images to get the eigen-vectors. These eigen-vectors are given as input to RBFN network as the inputs for training and recognition. The proposed method has good performance good recognition rate.**

Keywords: RobustPCA, Neural networks, Radial basis function network, Face recognition, Eigen vectors, PCA

1. Introduction

Face Recognition is one of the key areas under research. It has number of applications and uses. Many methods and algorithms are put forward like, 3D facial recognition etc. Face recognition comes under Biometric identification like iris, retina, finger prints etc. The features of the face are called biometric identifiers. The biometric identifiers are not easily forged, misplaced or shared hence access through biometric identifier gives us a better secure way to provide service and security.We can also develop many intelligent applications which may provide security and identity. we propose a novel method for face recognition using RobustPCA and RBFN. These systems can be well incorporated into mobile and embedded systems efficiently and can be utilized on larger scale.Face recognition becomes challenging with varied illumination and pose conditions. This method over comes the varied illumination problem and detection in noisy environments.

2. System Overview

The procedure for Face recognition is as follows.

1. Pre processing: The image is rescaled and the noise is reduced, contrast was normalized with histogram equalization..
2. RobustPCA: The images then are applied with RobustPCA for reduction in dimensionality and there by reducing complexity.
3. Modified RBFN: The outputs of Robust PCA are applied to RBFN for classification , separation of faces and non faces and for training.

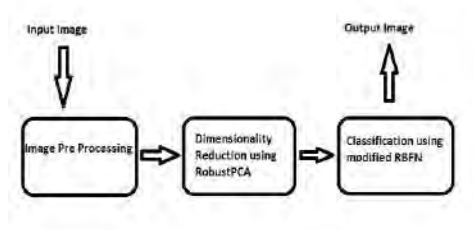

Fig 2.1 Face Recognition system diagram.

3. Classical PCA

PCA[1] is a statistical technique that is useful for dimensionality reduction Let $\mathbf{D} = [\mathbf{d}_1\ \mathbf{d}_2 \dots \mathbf{d}_n] = [\mathbf{d}^1\ \mathbf{d}^2 \dots \mathbf{d}^d]^T$ be a matrix $\mathbf{D} \in \Re^{d \times n}$ 1, where each column $\mathbf{d}_i$is a data sample, *n* is the number of training images, and d is the number of pixels in each image. We assume that training data is zero[7][8] mean, otherwise the mean of the entiredata set is subtracted from each column $\mathbf{d}_i$ Previous formulations assume the data is zero mean. In the least-squarescase, this can be achieved by subtracting the mean from the training data. For robust formulations, the "robust mean" must be explicitly estimated along with the bases.

Let the first *k*principal components of**D** be **B** = $[\mathbf{b}_1, \dots, \mathbf{b}_k] \in \Re^{d \times k}$. The columns of **B**are the directionsof maximum variation within the data. The principal components maximize max $\sum_{i=1}^{n} ||\mathbf{B}^T\mathbf{d}_i||_2^2 = \mathbf{B}^T\mathbf{\Gamma}\mathbf{B}$ with the constraint $\mathbf{B}^T\mathbf{B} = \mathbf{I}$, where $\mathbf{\Gamma} = \mathbf{D}\mathbf{D}^T = \sum_i \mathbf{d}_i\mathbf{d}_i^T$ is the covariance matrix. The columns of**B** form an orthonormal basis that spans the principal subspace. If the effective rank of**D** is much less than *d* and we can approximate the column space of **D**withk << d principal components. The data$\mathbf{d}_i$can beapproximated by linear combination of the principal components as $\mathbf{d}_i^{rec} = \mathbf{B}\mathbf{B}^T\mathbf{d}_i$ where $\mathbf{B}^T\mathbf{d}_i = \mathbf{c}_i$ are the linear coefficients obtained by projecting the training data onto the principal subspace; that is, $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \dots \mathbf{c}_n] = \mathbf{B}^T\mathbf{D}$.

A method for calculating the principal components that is widely used in the statistics and neural network community formulates PCA as the least-squares estimation of the basis images **B**that minimize:

$$E_{pca}(\mathbf{B}) = \sum_{i=1}^{n} e_{pca}(\mathbf{e}_i) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{B}^\mathbf{T}\mathbf{d}_i||_2^2 = \sum_{i=1}^{n}\sum_{p=1}^{d}(d_{pi} - \sum_{j=1}^{k} b_{pj}c_{ji})^2 \quad (1)$$

Where $c_{ji} = \sum_{t=1}^{d} b_{tj}d_{ti}$, $\mathbf{B}^T\mathbf{B} = \mathbf{I}$, $||.||_2$ denotes L_2norm $\mathbf{e}_i = \mathbf{d}_i - \mathbf{B}\mathbf{B}^T\mathbf{d}_i$is the reconstruction error vector, and $e_{pca}(\mathbf{e}_i) = \mathbf{e}_i^T\mathbf{e}_i$is the reconstruction error of $\mathbf{d}_i$

Alternatively, we can make the linear coefficients an explicit variable and minimize

$$E_{pca_2}(\mathbf{B},\mathbf{C}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \mathbf{B}\mathbf{c}_i||_2^2. \quad (2)$$

One approach for estimating both the bases, **B**, and coefficients, **C**, uses the Expectation Maximization (EM) algorithm. The approach assumes that the data generated by a random process and computes the subspace spanned by the principal components when the noise becomes infinitesimal and equal in all the directions. In that case, the EM algorithm can be reduced to the following coupled equations:

$$\mathbf{B}^T\mathbf{B}\mathbf{C} = \mathbf{B}^T\mathbf{D} \quad \text{(E-step)}. \quad (3)$$

$$\mathbf{B}\mathbf{C}\mathbf{C}^T = \mathbf{D}\mathbf{C}^T \quad \text{(M-step)}. \quad (4)$$

EM alternates between solving for the linear coefficients**C**(Expectation step) and solving for the basis**B**(Maximization step).

In the context of computer vision, Shum et al. solve the PCA problem with known missing data by minimizing an energy function similar to (2) using a weighted least squares technique that ignores the missing data. The method is used to model a sequence of range images withocclusion and noise and is similar to the method of Gabrieland Zamir .

4. Robust PCA

The above methods for estimating the principal components are not robust to outliers that are common in training data and that can arbitrarily bias the solution This happens because all the energy functions and the co- variance matrix are derived from a least-squares (L2 norm) framework. While the robustness of PCA methods in computer vision has received little attention, the problem has been studied in the statistics and neural networks literature, and several algorithms have been proposed.

One approach replaces the standard estimation of the covariance matrix, Γ, with a robust estimator of the covariance matrix . This approach is computationally impractical for high dimensional data such as images. Alternatively, Xuand Yuille[2] have proposed an algorithm that generalizes the energy function (1), by introducing additional binary variables that are zero when a data sample(image) is considered an outlier. They minimize

$$E_{xu}(\mathbf{B},\mathbf{V}) = \sum_{i=1}^{n}[V_i\|\mathbf{d}_i - \mathbf{B}\mathbf{B}^{\mathbf{T}}\mathbf{d}_i\|_2^2 + \eta(1-V_i)]$$
$$= \sum_{i=1}^{n}\left[V_i(\sum_{p=1}^{d}(d_{pi} - \sum_{j=1}^{k} b_{pj}c_{ij})^2) + \eta(1-V_i)\right] \quad (5)$$

Where $c_{ij} = \sum_{t=1}^{d} b_{tj}d_{ti}$. Each V_i in $\mathbf{V} = [V_1, V_2, \ldots, V_n]$

variable. IfV_i= 1the sample$\mathbf{d_i}$is taken

into consideration, otherwise it is equivalent to discarding$\mathbf{d_i}$as an outlier. The second term in (5) is a penalty term, or prior, which discourages the trivial solution where allV_i are zero. Given$\mathbf{B}$, if the energy

$e_{pca}(\mathbf{e}_i) = \|\mathbf{d}_i - \mathbf{B}\mathbf{B}^{\mathbf{T}}\mathbf{d}_i\|_2^2$ is smaller than a threshold, then the algorithm prefers toSetV_i= 1considering the sample as $\mathbf{d_i}$ an inlier and 0 if it is greater than or equal to ŋ

Minimization of (5) involves a combination of discrete and continuous optimization problems and Xu and Yuille [2] derive a mean field approximation to the problem which, after marginalizing the binary variables, can be solved by minimizing:

$$E_{xu}(\mathbf{B}) = -\sum_{i=1}^{n}\frac{1}{\beta}f_{xu}(\mathbf{e}_i,\beta,\eta) \quad (6)$$

The above techniques are of limited application in computer vision problems as they reject entire images as outliers. In vision applications, outliers typically correspond to small groups of pixels and we seek a method that is robust to this type of outlier yet does not reject the "good" pixels in the data samples. Gabriel and Zamir proposed a weighted Singular Value Decomposition (SVD) technique that can be used to construct the principal subspace. In their approach, they minimize:

$$E_{gz}(\mathbf{B},\mathbf{C}) = \sum_{i=1}^{n}\sum_{p=1}^{d} w_{pi}(d_{pi} - (\mathbf{b}^p)^T\mathbf{c}_i)^2 \quad (7)$$

where, recall, $\mathbf{b}^p$is a column vector containing the elements of the p-th row of$\mathbf{B}$This effectively puts a weight,wpi on every pixel in the training data. They solve the minimization problem with "criss-cross regressions" which involve iteratively computing dyadic (rank 1) fits using weighted least squares. The approach alternates between solving for $\mathbf{b}^p$ or$\mathbf{C_i}$while the other is fixed; this is similar to the EM approach but without a probabilistic interpretation.

5. Modified Radial Basis Function Network

RBF networks[3][4] has its own techniques in multidimensional data interpolation. For classification purpose, RBFN models the class distributions by local basis functions while MLP seperates the classes by hidden units which form hyper-planes in input space. RBFN and MLP play very similar roles in that they both provide techniques for approximating arbitrary non-linear functional mappings between multidimensional spaces.Compared with MLP the advantage with RBFN is possibility to choose proper parameters for their hidden units without using full non-linear optimization techniques.

Specifically, RBFN has feed forward network architecture with an input layer, a hidden layer and an output layer. The input layer of network is set of d units, which accept d-dimensional input vector.The input units are connected to hidden units and they in turn are fully connected o the output units. Since face detection is two class classification problem, the output has single out put unit to determine whether it is face or not.

Denote the input pattern as feature vector X

ISSN:2249-5789
KesavaRao Seerapu et al , International Journal of Computer Science & Communication Networks,Vol 2(5), 584-589

The output of RBFN is computed as

$$y(\mathbf{x}) = g\left(\sum_{j=1}^{l} w_j h_j(\mathbf{x}) + w_0\right)$$
$$g(a) = \frac{1}{1+\exp(-a)} \quad 8$$

Where W are connecting weights and HIJ are Activation functions(AF) of hidden units and they were taken as Gaussian.

$$h_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mu\|^2}{2\sigma_j^2}\right) \quad (9)$$

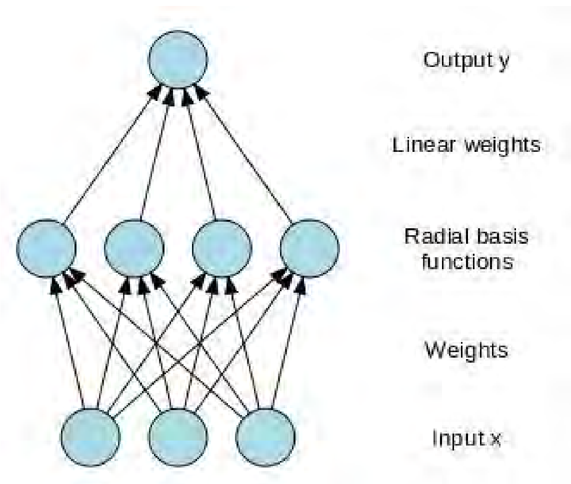

Fig 5.1 Architecture of a radial basis function network. An input vector **x** is used as input to all radial basis functions, each with different parameters. The output of the network is a linear combination of the outputs from radial basis functions.

The Gaussian AF'S are characterized by their centers and their variances. The centers can be regarded as prototype of their input vectors.

In our problem X is a vector, transformed by concatenating rows of a window image. Therefore X has high dimensionality, in order to reduce the dimensionality we used RobustPCA before we fed it into the network.

Let the mean vector of face patterns be Fa and the eigen vectors. The fatures of subspace of m(m<d) dimension can be obtained by

$$z_i = (\mathbf{x} - \bar{f}_a)^T \phi_i, \qquad i = 1, 2, \ldots, m \quad (10)$$

The features are projections of input patterns onto the sub space spanned by *m*eigen-vectors[9] while the information in complement space is omitted. RPCA extracts the eigen-vectors of feature subspace such that approximate error is minimized. However in the object detection the distance from input pattern from feature subspace (DFFS) provide useful information for discrimination

$$D_f = \|\mathbf{x} - \bar{f}_a\|^2 - \sum_{i=1}^{m} z_i^2 \quad (11)$$

It is expected that object patterns have small distance from feature subspace, while non pattern have large distance.Hence we incorporate DFFS into RBFN to further improve classification performance.

$$y(\mathbf{x}) = g\left(\sum_{j=1}^{l} w_j h_j(\mathbf{z}) + w^D D_f + w_0\right) \quad (12)$$

The modified RBFN trained by supervised learning on set of face and non face samples to reduce MSE(mean square error)

$$E = \frac{1}{2}\left(\sum_{n=1}^{N_x} [y(\mathbf{x}^n) - t^n]^2 + \lambda \sum_{j=1}^{l} w_j^2\right) = \sum_{n=1}^{N_x} E^n$$
$$E_n = \frac{1}{2}\left([y(\mathbf{x}^n) - t^n]^2 + \frac{\lambda}{N_x}\sum_{j=1}^{l} w_j^2\right) \quad \ldots(13)$$

Where t_n denotes the target output of the input pattern x_n,it takes 1 for face pattern and 0 for non face pattern. The weight decay is helpful to improve performance of neural networks.

The connecting weights are updated by stochastic gradient descent. Initially weights are randomly given and Gaussian centers are by k-means clustering on face and non-faces samples.On the input pattern the weights are updated by

$$w_j(n+1) = w_j(n) - \eta \frac{\partial E^n}{\partial w_j}, \quad j = 1, 2, \ldots, l$$
$$\mu_j(n+1) = \mu_j(n) - \eta \frac{\partial E^n}{\partial \mu_j}, \quad j = 1, 2, \ldots, l \qquad ..(14)$$

The partial derivatives are computed by

$$\frac{\partial E^n}{\partial w_j} = \begin{cases} (y(\mathbf{z}^n) - t^n)\frac{\partial y}{\partial w_j} + \frac{\lambda}{N_s} w_j, & w_j \neq w_0 \\ (y(\mathbf{z}^n) - t^n)\frac{\partial y}{\partial w_j}, & w_j = w_0 \end{cases}$$

$$\frac{\partial y}{\partial w_j} = \begin{cases} y(\mathbf{z}^n)(1 - y(\mathbf{z}^n)h_j, & w_j \neq w_0, w^D \\ y(\mathbf{z}^n)(1 - y(\mathbf{z}^n)D_f, & w_j = w^D \\ y(\mathbf{z}^n)(1 - y(\mathbf{z}^n)), & w_j = w_0 \end{cases}$$

$$\frac{\partial E^n}{\partial \mu_j} = (y(\mathbf{z}^n) - t^n))y(\mathbf{z}^n)(1 - y(\mathbf{z}^n))h_j(\mathbf{z}^n) \cdot w_j(\mathbf{x} - \mu_j)/\sigma_j^2 \qquad ..(15)$$

The training samples are repeatedly fed into Modified RBFN to update parameters until they stabilize.

6. Experimentation and Results

The experimentation is done with, 200 images from Yale database.

FIG 5.1 sample images from database.

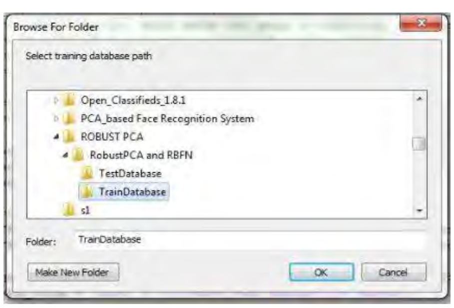

Fig 5.2 window, input image, and output image

Fig 5.3 input image, and output image

7. Conclusion.

In this paper we presented a neural network based face recognition system which can recognize even with complex backgrounds, illumination and clutter. We have used Robust PCA and Modified RBFN to achieve face recognition. The complexity and dimensionality is reduced by Robust PCA and RBFN gives better speed when compared with native neural methods, so this method gives better result in real time systems also the above system has scope for further development also can be implemented in various real time face recognition systems.

References

[1] Prof.UjvalChaudhary, ChakoliMateen Mubarak, Abdul Rehman, Ansari Riyaz, ShaikhMazhar*Face Recognition Using PCA-BPNN AlgorithmijmerVol.2, Issue.3, May-June 2012 pp-1366-1370 ISSN: 2249-6645*

[2] L. Xu and A. Yuille. Robust principal component analysis by self-organizing rules based on

[3] R. Rojas(1996), Neural Network An Introductions, Springer-Verlag, Berlin,"IEEETransactions of Neural Networks. vol.8, no.1,pp158-200

[4] „Pattern Recognition and Neural Networks‟by B.D. Ripley Cambridge University Press, 1996, ISBN 0-521-46086-7

[5] Henry A. Rowley, ShumeetBaluja, and Takeo Kanade Neural Network-Based Face Detection(PAMI, January 1998) pp 4-20

[6] SantajiGhorpade, JayshreeGhorpade, ShamlaMantri, DhanajiGhorpade "NEURAL NETWORKS FOR FACE RECOGNITION USING SOM",Dept. of Information Technology Engineering, Pune University, India, IJCST Vol. 1, Issue 2, December 2010

statistical physics approach. IEEE Trans. Neural Networks, 6(1):131–143, 1995.

[7] Kailash J. Karande Sanjay N. Talbar "Independent Component Analysis of Edge Information for Face Recognition" International Journal of Image Processing Volume (3) :Issue (3) pp: 120 -131.

[8] M. Tipping and C. Bishop. Probabilistic principal component analysis.Journal of the Royal Statistical Society B, 61,611-622, 1999.

[9] M. Turk and A. Pentland. Eigenfaces for recognition.J.Cognitive Neuroscience, 3(1):71–86, 1991.

[10] L. Xu and A. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. IEEE Trans. Neural Networks, 6(1):131–143, 1995.