FACE DETECTION USING NEURAL LEARNING WITH OPTIMIZED FEATURE SET

Stan Z. Li *Juwei Lu* *Kap Luk Chan*

School of Electronic and Electrical Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
szli@szli.eee.ntu.edu.sg http://markov.eee.ntu.edu.sg:8000/˜szli/
Tel:+65 790-4540 Fax: +65 791-2687

ABSTRACT

A system using feature optimization and neural learning techniques is presented for detection of upright frontal faces. The design of the system takes advantages of the existing techniques and avoid their shortcomings. Statistical pattern recognition (PR) techniques are used to optimize the feature selection. A neural network (NN) method is used to learn a complex mapping function for the classification, given the optimized feature set. The optimization of feature set reduces the burden of the subsequent NN classifier and improves its performance in learning speed and classification rates. The use of the NN for classification avoids the need for the simplification of classifier function, as practiced in the PR approach, for the mathematical tractability at the sacrifice of the performance. Experimental results show that our system produces higher detection and lower missing rates than several existing state-of-the-art face detection systems, with an average false detection rate.

Keywords: Face detection, principle component analysis, Fisher's Linear Discriminant, Neural networks.

1. INTRODUCTION

The first step in automated face recognition is face detection by which the location and size of each face is determined. Its reliability has a major influence on the performance and usability of the whole face recognition system. Various methods exist for face detection, including correlation in eigenface space [1], neural networks [2, 3, 4], probabilistic estimation [5], that hybridizing probabilistic estimation and neural network [6], labeled graphs [7], and also geometric feature based [8]. Most methods for face detection are based on normalized correlation or template matching. In such methods, the input image is windowed (with varying window sizes) from location to location, and the subimage in the window is classified into face or non-face.

Two issues are central: (i) what features to use to represent a subimage for the purpose of face detection, and (ii) how to classify the pattern in the subimage, based on the chosen features, into one of the two possibilities. Template matching methods treat face detection as an intrinsically two-dimensional (2-D) problem, taking advantage of the fact that faces are highly correlated. They assume that human faces can be described by some low-dimensional features which may be derived from a set of prototype face images. Based on the chosen features, the classification of a subimage into face or non-face is done by using a classification function involving some parameters which have to be learned.

There are two broad types of approaches for solving the problem, in terms of balance between feature extraction and classification [9]: pattern recognition (PR) based and neural network (NN) based. In the PR approach, the whole task is divided into a feature extractor, which transforms the input pattern into a low-dimensional feature vector, and a recognizer which does the feature-based classification. The feature extractor requires most design effort because it is task-specific and is often hand-crafted, while the classifier part is general-purpose and trainable. The recognition accuracy depends on the quality of the design of the feature set and the parameters tuned for the classification function.

The NN approach (of multi-layer networks) combines the two processes into one by taking the high-dimensional data as the input and training a network to learn a complex mapping for classification. This allows the designer to rely more on learning and less on detailed engineering for feature extraction. The ability of the mapping function to fit and to generalize patterns and hence the recognition accuracy depends on the design of the network architecture and the way the network is trained.

In this work, we present a face detection system that is designed based on the following ideas to take advantages of the PR and NN approaches and avoid their shortcomings (Section 2): (i) Given a set of training patterns, the feature set is optimized using well-developed PR techniques. (ii) Given the optimized feature set, a complex mapping function is learned for classification using an NN. The optimization of the feature set reduces the burden of the neural network. The use of the NN for classification avoids the need for the simplification of the classifier function, as practiced in the PR approach, for the mathematical tractability at the sacrifice of the performance.

In the optimization of feature set, the training patterns of face and non-face are clustered into a few distinct Gaussian distributions (classes), respectively, to explain a wide variety of variations such as facial appearance, pose, expression and illumination. During the clustering, principle component analysis (PCA) is used to approximate the Gaussian distributions. Then, the feature set is optimized in terms of the Fisher's linear discrimination (FLD) criterion, which maximizes the ratio of the between-class scatter and the within-class scatter [10], to provide best discrimination. The optimal feature space is of a further reduced dimension.

Although the feature set optimized using FLD for the best linear discrimination may not be optimal for nonlinear discrimination as performed by a multi-layer NN, it provides more compact clusters in the optimized feature space and better separability between clusters. We believe that this would help ease the training of the NN and obtain better classification performance.

Taking the optimized feature vector as the input, an NN consisting of two concatenated independent multi-layer feedforward NNs is trained to classify windowed patterns as faces or non-faces. The form and the involved parameters of highly complex classification function are thus learned from examples. Multiple NN arbitration is used to improve detection performance. Also, we propose to use a heuristic that takes the advantage of the symmetry property of faces to effectively reduce false detection.

This paper is closely related to the following work: Turk and Pentland [1] described a detection system based on eigenfaces. The input test pattern is classified as face or non-face by comparing with the training samples in their weights (coordinates of the projection in the eigenface space). A non-face is assumed at the input, if the weighted sum is less than a threshold.

Moghaddam and Pentland [5] presented a maximum-likelihood approach. Densities are estimated in high dimensional spaces using the eigenface decomposition. Two types of density estimators are derived for modeling the training data using either a multivariate Gaussian or a Mixture-of-Gaussian models. With these probabilities, one can then formulate the face detection as a maximum-likelihood problem. This work exploited the intrinsic low-dimensionality of the training imagery to form a computationally simple estimator for the complete likelihood function of the object.

Sung and Poggio [6] came up with a technique for modeling the distribution of human face patterns by means of a few view-based face and non-face clusters. PCA is used to partition the pattern space into the principle component space and the complement null space. Two types of distances are used as features for classifying an input pattern: the Mahalanobis distance from the cluster center in each principle component space and the Euclidean distance from its projection in the principle component space. A trained classifier (a multi-layer network or a perceptron) determines, based on the vector of such distances, whether or not a human face exists at the current image location. The bootstrap algorithm is used to collect meaningful non-face samples into the training set.

Rowley *et al.* [3] presented a face detection system based on retinally connected neural networks. The input to the NN is the preprocessed image pixel values directly. Simple heuristics, such as using the fact that faces rarely overlap in images are used improve the accuracy of their detection systems. Post-processing of the neural networks are performed by either ANDing/ORing the outputs or using an additional neural network to arbitrate between the outputs. The bootstrap algorithm is also employed for enlarging the training set.

Osuna *et al.* [4] applied the support vector machines algorithm to train an NN to classify face and non-face patterns. Performance is evaluated and compared with [6] using a test data set.

Experiments are conducted to compare our system with other systems [6, 3, 4] in the detection and false detection rates (Section 3). A common test set of 23 images containing 155 faces from K.K. Sung are used to obtain performance statistics. The results show that our system produces higher detection and lower missing rates than several existing state-of-the-art face detection systems, with an average false detection rate.

2. THE METHODS

A set of face and non-face patterns is given as the training set (see Fig.1:Left). Every pattern is preprocessed by window resizing into the standard size of 19×19, illumination correction, mean value normalization, histogram equalization. A preprocessed pattern is represented by a vector in the $N = 19 \times 19 = 361$ dimensional space. The subsequent processing consists of two parts: the PR-based optimal feature extraction and the NN-based learning for classification.

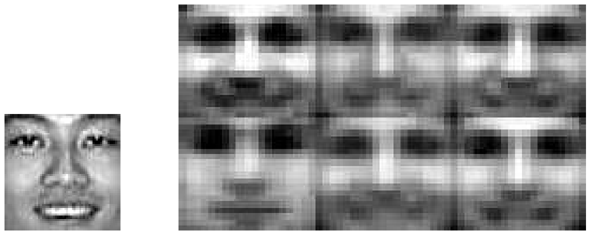

Figure 1: A example of canonical face pattern (left) and centroids of six face and six non-face clusters

2.1. PR-Based Feature Selection

2.1.1. Multi-Gaussian Approximation of Face Distribution

The distribution of training patterns is very complex because of variety of changes and high dimensionality. It is not appropriate to use a single distribution to explain all such variations. Therefore, the training patterns of faces are clustered into a few $L_f = 6$ face clusters, and the training patterns of non-faces into $L_n = 6$ non-face clusters, respectively.

The clustering is performed by using a modified k-mean clustering algorithm, proposed by [6]. Assuming that the k-th cluster is a Gaussian distribution with mean vector $\overline{x}^k$ and covariance matrix Σ^k, the clustering is based on the Mahalanobis distance $D_M(x, \overline{x}^k)$.

For high dimensional vector spaces, modeling a distribution with a full-covariance Gaussian is often not feasible because one usually has too few training samples to recover the covariance matrix accurately. Therefore, some approximations of the Gaussian distributions are necessary. PCA can be performed on each cluster to derive the principle components of the cluster [5, 6]. A number of $M = 50$ ($\ll N$) principle components of the cluster are computed from Σ^k and then used to approximate the patterns in the cluster originally in the $N = 361$ dimensional image space. Then the k mean clustering can be performed by using the approximate M-dimensional (M-D) mean vectors and $M \times M$ covariance matrices. Fig.1 shows the means (centers) of the 6 face clusters.

The PCA partitions, for each cluster, the high (N-) dimensional feature space into the low (M-) dimensional principal component subspace and the complement $N - M$ dimensional null-space. Let Φ_i be the eigenvector of Σ^k corresponding to its i-th largest eigenvalue e_i. The first M principle components $\{\Phi_i\}_{i=1}^M$ provides the basis for the principal component subspace and the other $N - M$ components forms the null-space.

Two sources of information are derived for each training pattern x, based on the PCA of a cluster: (i) Its projection into the principal component subspace of the cluster as

$\{w_i\}_{i=1}^{M}$, where $w_i = \Phi_i^T(x - \overline{x}^k)$ are the coordinates, normalized as $\{w_i/e_i\}_{i=1}^{M}$. (ii) The normalized Euclidean distance between x and its projection is $D_E(x, \overline{x}^k) = \frac{1}{\rho}(\left\|x - \overline{x}^k\right\|^2 - \sum_{i=1}^{M} w_i^2)$ where ρ is best estimated as $\rho^* = \frac{1}{N-M}\sum_{i=1+M}^{N} e_i$ [5]. Both the coordinates $W = [w_1/e_1, \ldots, w_M/e_M]$ and the distance D_E provide useful information for classification [6]. Now, x is represented, with respect to this cluster, by W and D_E, with $M + 1 = 51$ components.

2.1.2. *Optimal Linear Discriminant Feature Selection Using FLD*

Considering each of the $L_f + L_n$ clusters of the training patterns as an individual class, FLD transform is performed on the W's and D_E's of all the $L_f + L_n = 12$ clusters. This is to obtain the optimal linear discriminant feature set for the classification into one of the $L_f + L_n$ classes. The FLD also provides a further reduction of the dimensionality which is necessary: There are a total number of $(L_f + L_n) \cdot (M+1) = 612$ feature components for the 12 clusters; a multi-layer NN with 612 input units would be too large to train.

The FLD finds a set of basis vectors in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. Let S_{BTW} and S_{WTH} be the between- and within-class scatter matrices, respectively. The new basis vectors, denoted ϕ_k, is found by maximizing $\left|\phi^T S_{BTW}\phi\right| / \left|\phi^T S_{WTH}\phi\right|$ where $\phi = [\phi_1, \ldots \phi_m]$ and m is the reduced dimension. Assuming that S_{WTH} is non-singular, the basis vectors ϕ_k correspond to the eigenvectors of $S_{WTH}^{-1} S_{BTW}$ associated with the largest eigenvalues.

Two types of FLD transforms are performed. The first type is performed on the $(L_f + L_n) \cdot M = 600$ principle components (for $W = [w_1/e_1, \ldots, w_M/e_M]$) of the $(L_f + L_n)$ clusters, resulting a subspace of $m_W = 11$ dimensions for the W features. An intermediate dimension reduction from 600 to 100 is performed by using PCA to solve the stability problem therein in the same manner as in [11, 12]. The second types is performed on the $(L_f + L_n) \cdot 1 = 12$ distances (for D_E), resulting another subspace of $m_D = 11$ dimensions for the D_E features. By this, a pattern is represented by a vector of $m_W + m_D = 22$ components.

2.2. NN-Based Classifier

2.2.1. *Architecture*

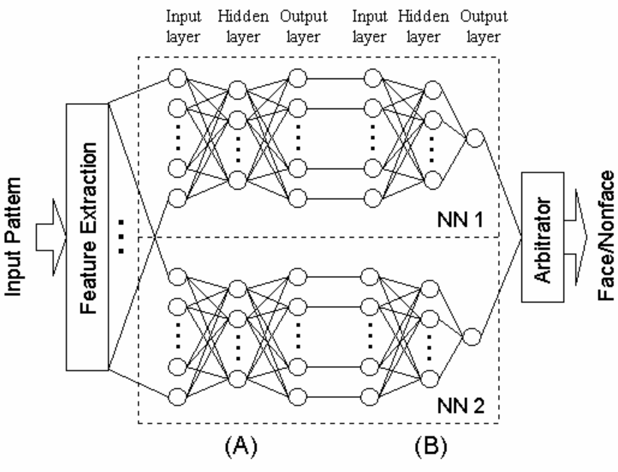

Figure 2: The NN architecture for face detector.

The NN classifier is composed of two identical NNs (two are used for multiple network arbitration, see below), each consisting of two concatenated but independent networks, A and B (see Fig.2). Each of A and B is a fully connected feed-forward multi-layer perceptron. Part A consists of a 22 unit input layer and a 12 unit output layer. It receives the optimal feature vector of 22 dimension from the FLD transforms as the input and classify it into one of the 6 face "sub-classes" and 6 non-face "sub-classes". Network B consists of a 12 unit input layer and a 1 unit output layer. It receives the output of network A as the input, and further classifies the input pattern into one of the face and non-face classes. In both A and B, a back-propagation (BP) algorithm is used to perform supervised learning.

Network A considers each of the 12 clusters of the training patterns as an individual sub-class and has 12 output neurons. Each of the 12 output neurons is supposed to produce the largest response for the corresponding sub-class. It also takes into consideration the relationship between a sub-class and its "sup-class" (face or non-face). The desired output value is 1, 0 or -1 depending on the sub-class and the sup-class labels.

Therefore, network A is trained in the following way: (i) If the output neuron corresponds to the same sup-class and sub-class as the input, the desired output is 1; (ii) if the output neuron corresponds to the same sup-class but a different sub-class , the desired output is 0; (iii) If the output neuron corresponds to the opposite sup-class, the desired output is -1.

Network B takes the output from A as its input and has one output neuron. It is trained in the following way: The desired output is 1 if the input (to network A) is of face; or -1 if the input is of non-face.

The training of A and B is done independently. If trained dependently without having the designated target output of network A, the overall architecture would be an NN with one hidden layer corresponding to the output layer of A and the input layer of B. The independent training is made possible by the meaning of the 12 output neurons of network A: they correspond to the 12 sub-classes, which is a feature different from general NNs with hidden layers.

Although network A alone is able to classify into face or non-face, for example by considering the sums of the outputs from the 6 face and 6 non-face output neurons respectively, smaller error is obtained by using the A+B architecture than by using network A alone.

2.2.2. *Mirroring Input Patterns for Verification*

The mirrored view of a human face can also be classified as a face pattern. Based on this property, a simple heuristic is used to verify the detection: If a face is detected at a location (the detection value is above the threshold), the input pattern is mirrored and passed through the face detector for another classification process. If there is still a detection with the mirrored input, the detection is confirmed. This can effectively reduce the false alarms with a very small sacrifice to the detection rate and small additional computational cost.

The effect is demonstrated in Fig.3. Two false alarms are eliminated by the verification using the mirrored input. Two false alarms remain, but are reasonable since the local images contained in these two locations emerged symmetrically.

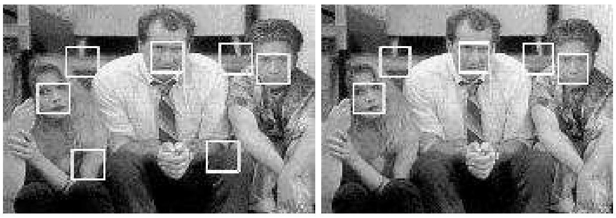

Figure 3: Effect of mirroring input to eliminate false alarms. Left: before mirroring. Right: after.

2.2.3. *Multiple Network Arbitration*

It is well known that a BP algorithm only converges to one of local minima. The performance of a thus trained network depends heavily on the initial condition of the training and even the order of the training samples. A multiple network arbitration strategy [3] is used for improving the detection as follows.

Several networks (two in our case, see Fig.2) are trained in a similar manner, but with different initial weights generated randomly and permutations of the order of presentation of the training images. It is observed that well-trained networks tend to detect roughly the same set of faces, but generally make different false positive errors. Therefore, the system arbitrates between the output of multiple networks to obtain a more reliable class prediction for each window pattern and reduce the number of false positives.

2.2.4. *Resolving Multiple Detections*

A face is initially detected by an NN if the output greater than a given threshold. Most face may be detected several times at close distances or multiple scales, while false detection often occur with less consistency. The observation leads to a heuristic which can resolve the ambiguities and eliminate many false detects through thresholding and merging on the multiple detects. As a result, the number of the multiple detects can be used as an effective indication for the existence of a face at a certain position.

3. EXPERIMENTS AND COMPARISONS

Two sets of experiments are presented to evaluate the performance of our system in terms of the detection and false detection rates. The first examines the effects of several heuristics on the performance. The second compares our system with other systems [6, 3, 4] using a common test data set.

Our training set consists of about 3000 face and 4000 non-face patterns (thanks to K.K. Sung for providing nearly half of them). The face patterns are of up-right frontal view possibly with small in-plane angular rotations. Later on after running the partially trained detector, additional non-faces are added by applying a bootstrap algorithm [6, 3]. No face patterns in the subsequent test sets are included in the training set.

The system is tested on a wide variety of images including 2 CMU databases used in [3] and 1 MIT database used in [6]. The test sets contain over one hundred images with different backgrounds, illumination and faces in all scales. Some detection results are shown in Figs.4 To save computational time, the detection was carried out using a few (normally 1 to 4) scales estimated manually.

3.1. Effect of Heuristics

This compares the effect of various heuristics on the performance. Four face detectors are designed: (i) that uses only the strategy of mirroring the input pattern on detection, (ii) that uses only the strategy of ANDing two NNs, (iii) that uses only the strategy of thresholding on multiple detection, and (iv) that combines all the three strategies. The comparison is conducted on a test set of 30 images.

Table 1: Effect comparison based on different heuristic strategies

Method	(i)	(ii)	(iii)	(iv)
Correct detection	85	81	82	80
Missed detection	12	16	15	17
Detection rate (%)	87.6	83.5	84.5	82.5
False detections	10	8	13	3

Table 1 tabulates the detection results on a typical test image of four detectors. Of interest is the numbers of false detections. Not surprisingly, because all the three heuristics help to reduce the false detection, detector (iv) tops on all the other networks with only 3 false detection. This is achieved by a slightly decrease in the detection rate.

3.2. Performance Comparison with Other Systems

This compares our system (of detector iv described above) with six other systems proposed by [6, 3, 4]: (1) Rowley 1: using one method of [3] with heuristics (two NNs AND $\rightarrow$ threshold $\rightarrow$ overlap elimination), (2) Rowley 2: that with (two NNs threshold $\rightarrow$ overlap elimination $\rightarrow$ AND), (3) Rowley 3: that with (threshold $\rightarrow$ overlap $\rightarrow$ OR $\rightarrow$ threshold $\rightarrow$ overlap); (4) Sung 1: a system of [6] with a multi-layer network, (5) Sung 2: that using perceptron (6) Sung 3: that using a multi-layer network and the same training set to (iv); and (7) Osuna: the system of [4] using Support Vector Machine network. To our knowledge, the Rowley systems have produced the best results reported so far. A common test set of 23 images containing 155 faces from K.K. Sung are used to obtain the performance statistics.

Table 2: Comparison with other systems

Systems	Missed faces	Detect rate	False detects
(iv)	11	92.9%	9
Rowley 1	39	74.8%	0
Rowley 2	24	84.5%	8
Rowley 3	15	90.3%	42
Sung 1	36	76.8%	5
Sung 2	28	81.9%	13
Sung 3	33	78.7%	9
Osuna	39	74.2%	20

The performance comparison in the detection and false detection rates is given in Table 2. Our system has a higher detection rate and a lower missing rate than all the other systems and a lower false detection rate than Rowley 3 and Osuna, though its false detection rate is higher than Rowley 1, Rowley 2, Sung 1 and Sung 2. It performs better in both detection and false detection rates than Rowley 3, the latter having the highest detection rate reported to date.

A comparison of training sets used by the compared systems is meaningful: As mentioned earlier, our system is trained with about 3000 face images and 4000 non-face examples, including fewer than 2000 non-faces added by bootstrap. In comparison, Sung's system [6] is trained with 4150 face patterns and 43166 non-face patterns; Rowley's system [3] is trained with 16000 face images and 9000 non-face images including more than 8000 obtained using the bootstrap strategy. As generally recognized, the performance of a trained NN increases with the size of training samples. We expect our system to have lower false detection rate if the training set is enlarged.

4. CONCLUSIONS

A face detection system using an efficient combination of pattern recognition (PR) and neural network (NN) techniques is presented. Based on our engineering knowledge about the optimal feature extraction, we are able to come up with an optimal linear discriminant feature set without resorting to the NN learning. The optimization of feature set reduces the burden of the subsequent NN classifier and improves its performance in learning speed and classification rates.

A multiple NN architecture is used to learn a complex function for adequate classification given the feature set. This is advantageous over the statistical PR approach in that a PR-based classifier takes a functional form which is often too simplified for mathematical tractability at sacrifice of classification performance.

The effectiveness of the above strategies is demonstrated by experimental results. The results show that our system produces a higher detection rate than several existing state-of-the-art face detection systems, with an average false detection rate. However, our system is still young. We expect it to achieve better performance as it is trained with more examples.

5. REFERENCES

[1] Matthew A. Turk and Alex P. Pentland, "Face recognition using eigenfaces.", in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hawaii, June 1991, pp. 586–591.

[2] G. Burel and D. Carel, "Detection and localization of faces on digital images", *Pattern Recognition Letters*, vol. 15, no. 10, pp. 963–967, 1994.

[3] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural network-based face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–28, 1998.

[4] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection", in *CVPR*, 1997, pp. 130–136.

[5] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 696–710, July 1997.

[6] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[7] Norbert Krüger, Michael Pötzsch, and Christoph von der Malsburg, "Determination of face position and pose with a learned representation based on labeled graphs", *Image and Vision Computing*, August 1997.

[8] G. Z. Yang and T. S. Huang, "Human face detection in a complex background", *Pattern Recognition*, vol. 27, pp. 53–63, 1994.

[9] Y. LeCun and Yoshua Bengio, "Pattern recognition and neural networks", in *The Handbook of Brain Theory and Neural Networks*, Michael A. Arbib, Ed., pp. 711–715. MIT Press, Cambridge, Massachusetts, 1995.

[10] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, Boston, 2 edition, 1990.

[11] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 831–836, 1996.

[12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

Figure 4: Face detection results