International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:9, 2008

Face Recognition Using Morphological Shared-weight Neural Networks

Hossein Sahoolizadeh, Mahdi Rahimi, and Hamid Dehghani

Abstract**—**We introduce an algorithm based on the morphological shared-weight neural network. Being nonlinear and translation-invariant, the MSNN can be used to create better generalization during face recognition. Feature extraction is performed on grayscale images using hit-miss transforms that are independent of gray-level shifts. The output is then learned by interacting with the classification process. The feature extraction and classification networks are trained together, allowing the MSNN to simultaneously learn feature extraction and classification for a face. For evaluation, we test for robustness under variations in gray levels and noise while varying the network's configuration to optimize recognition efficiency and processing time. Results show that the MSNN performs better for grayscale image pattern classification than ordinary neural networks.

Keywords**—**Face recognition, Neural Networks, Multi-layer Perceptron, masking.

I. Introduction

FACE recognition is one of the fields of biometric[4] Measurable physical or behavioral characteristics which can be used to verify the identity of an individual are called biometrics. These also primarily include fingerprints, retinal and iris scanning, hand geometry, and voice patterns, among other techniques. There are many areas in which face recognition can play a major role. Some are not necessarily high security applications, but face recognition can help to overcome a large number of unsolved identification problems, particularly in areas where instant face recognition is needed. Some examples are given in [5]:

A. Prison Visitor Systems

Visitors have to be verified so that identities may not be swapped during visits.

B. Identification of Drivers

Some drivers have fake licenses or they swap licenses among themselves to cross state borders.

C. Border Control

Face recognition can be used to identify criminals or fugitives at the airport. Face recognition has two main security applications: verification and identification. Verification is simply a one-to-one match that may be performed quickly and generates a true or false result. The system compares the features of the given image with the contents of its database, resulting in a match or no match according to predefined parameters. Identification allows the user to submit a live sample and the system attempts to identify the person by using the image library within the participating database the result may be a set of possible matches, ranked with respect to closeness to the given query.

Face recognition rose from the time when machines started to become more and more intelligent and had the advantages of filling in, correcting, or helping the lack of human abilities and senses. The subject of face recognition is as old as computer vision and has always remained a major focus of active research because of its non-invasive nature. Another reason is that it is human's instinctive method of identification. Two main approaches formed the early core techniques of facial feature analysis: the geometrical approach and the pictorial approach [6].

The geometrical approach uses spatial mapping of facial features. Faces are classified according to geometrical distances, perimeters, areas, and angles determined from point to point. We apply this technique to find the distance between the left and the right eye.

There have been active developments in face recognition systems in recent years. Here are the more common feature extraction and classification methods: Eigenfaces, Hidden Markov Model, Principal Component Analysis, Support Vector Machine, Probabilistic Decision-based Neural Network, Convolution Neural Network, and ARENA

The Eigenface algorithm [7] is widely implemented and well known for its simplicity and computational efficiency. It uses the information theory approach of coding facial images and attempts to find the top eigenvectors of the covariance matrix of an image. The coefficient vectors are then averaged for each class, after which thresholds are chosen to define the maximum allowable distance from the face class and face space for recognizing new images.

The Hidden Markov Model classifies a facial feature by the property of the Markov Chain. A sequence of random variables that takes on the respective pixel values forms a Markov Chain if the probability that the system is in state n+1

F. Hossein Sahoolizadeh is with the Young Researchers club of Arak, Islamic Azad University, Arak branch, Iran (e-mail: hosein_sahooli@yahoo.com).

S. Mahdi Rahimi is with the Young Researchers club of Arak Islamic Azad University, Arak branch, Iran (e-mail: mahdi.rahimi2008@yahoo.com).

T. Hamid Dehghani is with the Electrical Engineering Department Maleke Ashtar University of Technology Tehran, Iran (e-mail: hamid_deh@yahoo.com).

x at time n+1 depends exclusively on the probability that the system is in state n x at time n. In a Markov Chain, the transition from one state to another is probabilistic, but the production of an output symbol is deterministic. On the other hand, in a Hidden Markov Model, the output symbols are probabilistic. The result is that we have a probability distribution of all the output symbols at each state. We use this result to make comparison between two faces.

In Principal Components Analysis (also known as the Karhunen-Loeve Transformation) [8], the data set is represented by a reduced number of “effective” features and yet retain most of the intrinsic information content of the data. PCA is always used in line with the Eigenface method. Centered subsets of eigenvectors are used as basis vectors for a subspace in which we can compare data images and novel probe images. These basis vectors are also called the principal components of the database images.

Support Vector Machines [9] are in fact binary classification methods. By constructing a support vector, an inner-product kernel can be generated between the support vector and the input vector. The support vectors consist of a small subset of training data extracted by the algorithm. The goal of an SVM is to find the particular hyper plane for which the margin of separation is maximized. In SVM, face recognition is formulated as a problem in difference space that models dissimilarities between two facial images.

Probabilistic networks [10] classify training data or vectors into their correct classes by approximating their distribution or densities in a feature space. The PNN does not make the assumption that we know the distribution from which the training data or test data come from. It all depends on the amount of training data. The larger it is, the more accurate is the approximation.

Convolution networks get their name from the cross-correlation operation performed during neural training. It has a feed forward network which uses local connections and weight sharing. For the purpose of better regionalization, images are passed through masks or kernels to detect boundaries. Kernels are square matrices whose values are designed according to the types of effects or detections desired. The variances between image values and kernel complements are used as weights for training the network. The MSNN was developed from this type of network.

ARENA [11] is a new and simple, memory-based face recognition algorithm. The method does not perform any complex feature extraction, and nor does it incorporate any face-specific information. It measures direct distance metrics between images and may use a neural network to train its output.

Neural networks [5] are widely used in face recognition in combination with the above classification methods. Neural technology simulates the way neurons work in the human brain. This is seen as the main reason for its role in face recognition. A neural network has the ability to adjust its weights according to the differences it encounters during training. As a result, it delivers high efficiency in the classification of linearly as well as nonlinearly separable classes.

II. Neural Network

A neural network is an information-processing system that has been developed as generalizations of mathematical models matching human cognition. They are composed of a large number of highly-interconnected processing units (neurons) that work together to perform a specific task. According to Haykin [2], a neural network is a massively parallel-distributed processor that has a natural prosperity for storing experimental knowledge [3]. It resembles the brain in two respects:

Knowledge is acquired by the network through a learning process;

Inter-connected connection strengths known as synaptic weights are used to store the knowledge;

Each neuron has an internal state called its threshold or activation function (or transfer function) used for classifying vectors. Neural classification generally comprises of four steps:

Pre-processing, e.g., atmospheric correction, noise suppression, band rationing, Principal Component Analysis, etc;

Training - selection of the particular features which best describe the pattern;

Decision - choice of suitable method for comparing the image patterns with the target patterns;

Assessing the accuracy of the classification.

III. Modeling the Neural Network

The MSNN is a heterogeneous network composed of two cascaded sub networks:

The feature extraction phase followed by the training and classification phase. The feature extraction layer is modeled from mathematical morphology and focuses primarily on the hit-miss transform operation. Modeling the training network involves the development of learning algorithms according to the desired behavior and the functionality of the network. While developing our MSNN model, the following issues were our main concerns:

A. Complexity

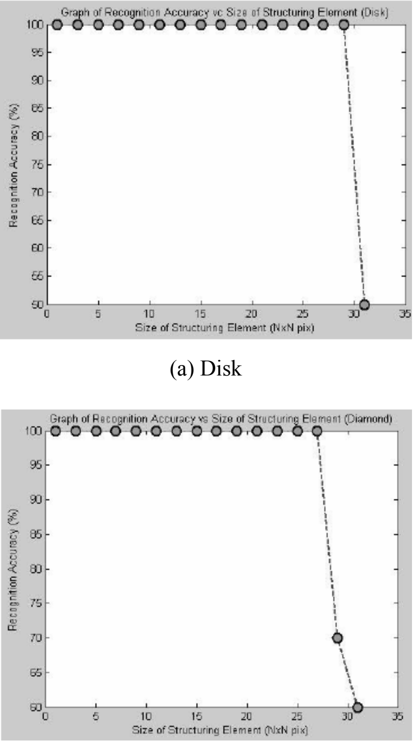
How large is our image database? How large should one image be? Should we make them smaller or larger by resizing?

B. Performance and Reliability

We need to know which neural network is reliable and learns fast. In terms of classification quality, we have to know how these various networks perform their computations. Are these techniques suitable for training digital images?

IV. Implementation Results

We ran experiments in which we varied the size and shape of the structuring element. The size was increased progressively from 1×1 to 31×31 pixels. Separate tests were conducted for the "disk" and the "diamond" structuring element. Results showed that the MSNN is not very sensitive to structuring element size and shape (Fig. 1). For the network that uses a "disk" structuring element, recognition accuracy remains constant at 100% until it drops abruptly at the size of 31×31 pixels; the fail size is 29×29 pixels for the network that uses a "diamond" structuring element. These findings indicate that the size of the structuring element must not get too close to the size of the input image.



(a) Disk

(b) Diamond

Fig. 1 Graph of recognition Accuracy vc size of structuring Elements

A. Learning Rate

Another test was conducted to observe training times with respect to different structuring element sizes. We found that training time is proportional to the increase in the size of the structuring element. Graph (a) in Fig. 1 shows an upward curve for the network that uses a "disk" structuring element; whereas in Graph (b), the network that uses a "diamond" structuring element has a linear timeline. This explains the nature of both structuring elements: the disk is a nonlinear detector, while the diamond is a discrete detector. The human face has a nonlinear pattern; hence, the "disk" structuring element should be used to perform hit-miss transform in the feature extraction stage.

The performance of the MSNN is very sensitive to the proper setting of the learning rate. It cannot be set too high; otherwise, the network may oscillate and become unstable. If the learning rate is too small, the algorithm will take a long time to converge.

Several trainings should be performed using a variety of learning rates before determining the optimum η.

We conducted experiments with learning rates ranging from 0.05 to 0.4, each time increasing by 0.05. Recognition accuracy increases with the learning constant until it reaches full recognition at η= 0.25, after which the performance starts to deteriorate. The recognition rate finally drops to 0% at η= 0.4 .Our observations are plotted out in Fig. 2.

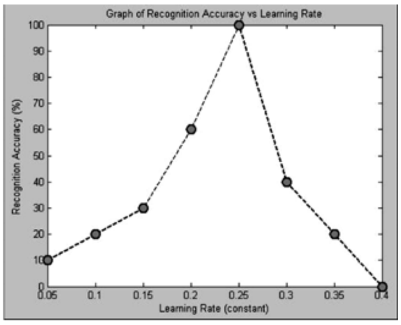


Fig. 2 Graph of recognition Accuracy vs leaning rate

B. Number of Hidden Neurons

How to determine the number of hidden neurons is always a discussion topic in neural networks. We have discussed the Baum and Haussler rule in the previous chapter, but that does not mean the equation will work for every network. The rule did not work in our case, so trial and error is still the best way to work out the optimal number of hidden units. The Baum and Haussler rule should be used as an estimator.

In our experiments, we trained the MSNN with different numbers of hidden units (5 to 40) and recorded their recognition rates. For a network with 10 to 20 hidden neurons, the recognition accuracy is 100%. Any number of hidden units beyond 20 will experience a gradual decrease in performance, eventually hitting zero recognition at 40 neurons. As the graph in Fig. 3 indicates, efficiency increases as the number of neurons increases from 5 to 10; it remains constant between 10 to 20 neurons, and drops drastically after that. Although it is true that increasing the number of hidden neurons can extract more implicit information, an excessive number may cause over fitting and high generalization error. On the contrary, the MSNN will not converge if the number of hidden units is too small under fitting results in high training and generalization error.

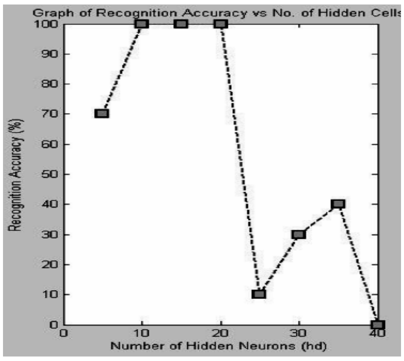


Fig. 3 Graph of recognition Accuracy vs NO of Hidden Cells

V. Conclusion

In conclusion, we have shown that the morphological shared-weight neural work can approach the robustness needed for face recognition. Although it trained slower than the normal multilayer network, it exhibited better generalization than the back propagation network in terms of accommodating noise and gray-level shifts. Both performed equally well at detecting raw images, but the MSNN maintained stability and performance under grayscale variation. We have also investigated the effects of training parameters on the MSNN. Experiments were carried out by altering the size and shape of the structuring element, the learning rate, the number of hidden neurons, and the types of sigmoid functions. The MSNN is not very responsive to structuring element size and shape; however, for training face images, the “disk” structuring element should be used, and its size should not be too large. The ideal size is 3×3 pixels for the standard ORL 112×92 image. The optimal value for our learning rate is 0.25, which produces the best performance in both face recognition and eye recognition. For the MSNN structure, the best number of hidden units is 10, while the best combination of sigmoid functions is the logistic function for both the hidden and output layers. Our MSNN borrowed the same concept created by Won [1]. He used it for target detection, whereas we used it for face recognition. Our MSNN is also a cascaded network that consists of a feature extraction layer and a feed forward network. Raw face images are under sampled to minimize computation intensity and input into the feature extraction layer. Feature extraction is performed over the entire face image, where all the pixels are mapped to a feature map using grayscale hit-miss transform. This feature map is a composite of eroded and dilated facial features. The hit weight is actually the structuring element for performing erosion; similarly, the miss weight is the structuring element for performing dilation. When the face image has passed through both these kernels separately, the eroded output and the dilated output are subtracted to obtain the difference matrix. This difference matrix forms the feature map.

Next, the feature map is fed into the back propagation network. The net sums entering the neurons are increased by adding a bias. They are then transformed into output by sigmoid. The output is subtracted by the target vector, and this error is used to calculate the correction term for the output layer. Since the network has one hidden layer, the output correction term is passed back to this hidden layer, which uses it to calculate its own correction term. Weights are then adjusted respectively at the output layer and the hidden layer. The final set of weights is then multiplied and sigmoid-transformed again with the original input to derive the final output for the entire training. This final output is later processed by another subprogram to compare against its prescribed threshold. If it falls within that range, the identification is confirmed. Image similarity computed by existing mathematical metric is not always consistent with the human perception. For instance, Euclidean distance may not effectively preserve the perceptual similarity due to the subjectivity of perceived similarity with respect to the related task and database. Therefore, designing effective feature extraction procedures is not an easy task. A good feature set makes the later part of the training and decision-making simpler and more accurate. The strength of the MSNN is in its translation-invariant extraction layer. It enables the network to learn complex patterns by extracting progressively more meaningful features from the input patterns of a face. The MSNN thus avoids being too restricted by mathematical metric in its classification process. This increases its ability to generalize.

References

[1] Y. Won et. al., Morphological Shared-Weight Networks with Applications to Automatic Target Recognition, Electronics and Telecommunications Research Institute, Daejon, South Korea, 1995.

[2] Y. Won and P. Gader, Morphological Shared-Weight Neural Network for Pattern Classification and Automatic Target Detection, University of Missouri-Columbia, 1995.

[3] D. Haun, K. Hummel, and M. Skubic, Morphological Neural Network Vision Processing for Mobile Robots, University of Missouri-Columbia, 1997.

[4] U. Uludag and A. Jain, Biometrics, International Conference on Pattern Recognition, Department of Computer Science and Engineering, Michigan State University, 1999.

[5] G. Scott and M. Skubic, Face Recognition Using Morphological Shared-Weight Neural Networks.

[6] V. Starovoitov, D. Samal, and D. Briliuk, Three Approaches for Face Recognition, The 6-th International Conference on Pattern Recognition and Image Analysis, Velikiy Novgorod, Russia, 2002.

[7] L. Aryananda, Online and Unsupervised Face Recognition for Humanoid Robot: Toward Relationship with People, A. I. Lab, MIT, 2001.

[8] K. Jung, Face Recognition Using Kernel Principal Component Analysis, Michigan State University, 2001.

[9] J. Huang, X. Shao, and H. Wechsler, Face Pose Discrimination Using Support Vector Machines, George Mason University and University of Minnesota, 1998.

[10] K. Yiu, M. Mak, and C. Li, Gaussian Mixture Models and Probabilistic Decision-Based Neural Networks for Pattern Classification: A Comparative Study, Hong Kong Polytechnic University, HK, China, 1999.

[11] T. Sim, R. Sukthankar, M. Mullin, and S. Baluja, Memory-based Face Recognition for Visitor identification, The Robotics Institute, Carnegie Mellon University, 2000.