Morphological Shared Weight Neural Network: A method to improve fault tolerance in Face Recognition

Sumedha, Monika Jenna

Abstract— In this paper, the focus is on the investigation of face recognition using morphological shared-weight neural network. Being nonlinear and translation-invariant, the MSNN can be used to create better generalization during face recognition. The MSNN is a heterogeneous network that produces high order features based on local features extracted by morphological operations. 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.

Index Terms— Fault Tolerance, Pattern Recognition, Neural Networks, Back-Propagation Neural Network, Morphological operations

I. Introduction

Fault-tolerant computing is the art and science of building computing systems that continue to operate satisfactorily in the presence of faults. A fault-tolerant system may be able to tolerate one or more fault-types including -- i) transient, intermittent or permanent hardware faults, ii) software and hardware design errors, iii) operator errors, or iv) externally induced upsets or physical damage. An extensive methodology has been developed in this field over the past thirty years, and a number of fault-tolerant machines have been developed -- most dealing with random hardware faults, while a smaller number deal with software, design and operator faults to varying degrees. A large amount of supporting research has been reported. Fault tolerance and dependable systems research covers a wide spectrum of applications ranging across embedded real-time systems, commercial transaction systems, transportation systems, and military/space systems -- to name a few. The supporting research includes system architecture, design techniques, coding theory, testing, validation, proof of correctness, modeling, software reliability, operating systems, parallel processing, and real-time processing. These areas often involve widely diverse core expertise ranging from formal logic, mathematics of stochastic modeling, graph theory, hardware design and software engineering.

Manuscript received March, 2012.

***Sumedha**, Computer Science, Amity School of Engineering & Technology, Noida, India, Phone/Mobille No. +919899240428,*

***Monika Jenna**, Computer Science, Amity School of Engineering & Technology, Noida, India, Phone/Mobille No. +919810248237,*

Basic Concepts

Hardware Reliability -- The majority of fault-tolerant designs have been directed toward building computers that automatically recover from random faults occurring in hardware components. The techniques employed to do this generally involve partitioning a computing system into modules that act as fault-containment regions. Each module is backed up with protective redundancy so that, if the module fails, others can assume its function. Special mechanisms are added to detect errors and implement recovery.

Software Reliability -- Efforts to attain software that can tolerate software design faults (programming errors) have made use of static and dynamic redundancy approaches similar to those used for hardware faults. One such approach, N-version programming, uses static redundancy in the form of independently written programs (versions) that perform the same functions, and their outputs are voted at special checkpoints. Here, of course, the data being voted may not be exactly the same, and a criterion must be used to identify and reject faulty versions and to determine a consistent value (through inexact voting) that all good versions can use. An alternative dynamic approach is based on the concept of recovery blocks. Programs are partitioned into blocks and acceptance tests are executed after each block. If an acceptance test fails, a redundant code block is executed.

Pattern Recognition/Image Processing

Automatic (machine) recognition, description, classification, and grouping of patterns are important problems in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, artificial intelligence, and remote sensing. A pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal. Given a pattern, its recognition/classification may consist of one of the following two tasks: 1) supervised classification (e.g., discriminate analysis) in which the input pattern is identified as a member of a predefined class, 2) unsupervised classification (e.g., clustering) in which the pattern is assigned to a hitherto unknown class. [16] The recognition problem here is being posed as a classification or categorization task, where the classes are either defined by the system designer (in supervised classification) or are learned based on the similarity of patterns (in unsupervised classification).These applications include data mining (identifying a "pattern", e.g., correlation, or an outlier in millions of multidimensional patterns), document classification (efficiently searching text documents), financial forecasting, organization and retrieval of multimedia databases, and biometrics. The rapidly growing and available computing power, while enabling faster processing of huge data sets, has also facilitated the use of elaborate and diverse methods for data analysis and classification. At the same time, demands on automatic pattern recognition systems are rising enormously due to the availability of large databases and stringent performance requirements (speed, accuracy, and cost). The design of a pattern recognition system essentially involves the following three aspects: 1) data acquisition and pre-processing, 2) data representation, and 3) decision making. It is generally agreed that a well-defined and sufficiently constrained recognition problem (small interclass variations and large interclass variations) will lead to a compact pattern representation and a simple decision making strategy. Learning from a set of examples (training set) is an important and desired attribute of most pattern recognition systems. The four best known approaches for pattern recognition are: 1) template matching, 2) statistical classification, 3) syntactic or structural matching, and 4) neural networks. [17]

Neural Network

Neural network techniques are powerful tools that prove their efficiency in real-world applications, where problems are badly defined or difficult to formalize. Some applications, especially those involving images, require a huge number of operations and an enormous reduction of the dataflow from input to output data. For sorting objects by vision or image analysis, for example, inputs are images at video rate; outputs are correcting values, names of objects, or locations of specific objects in a picture.

The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data. The most commonly used family of neural networks for pattern classification tasks is the feed-forward network, which includes multilayer perceptron networks. Another popular network is the Self-Organizing Map (SOM), or Kohonen-Network, which is mainly used for data clustering and feature mapping. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification/clustering task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge and due to the availability of efficient learning algorithms for practitioners to use [13].

II. LITERATURE SURVEY

Techniques from statistical image recognition have, since the revival of neural networks, obtained a widespread use in digital image processing. Initially, pattern recognition problems were often solved by linear and quadratic discriminates [3] or the (non-parametric) k-nearest neighbour classi1er and the Parson Density estimator [4]. In the mid-eighties, the PDP group together with others introduced the back-propagation learning algorithm for neural networks. This algorithm for the 1st time made it feasible to train a non-linear neural network equipped with layers of the so-called hidden nodes. In their 1993 review article on image segmentation, Pal and Pal predicted that neural networks would become widely applied in image processing [5].

The different problems addressed in the field of digital image processing can be organised into an image processing chain as shown in figure (1) [6].

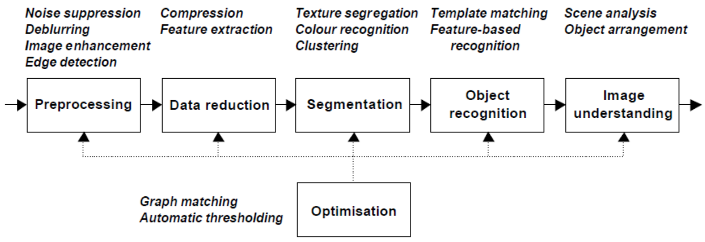

Fig1 An image processing chain [6]

Image reconstruction problems often require quite complex computations and a unique approach is needed for each application.. Meyer and Heind [7] used regression feed-forward networks (that learn the mapping E(y/x), with x the vector of input variables and y the desired output vector) to reconstruct images from electron holograms. Wang and Wahl trained a Hopfield ANN for reconstruction of 2D images from pixel data obtained from projections [8].

The majority of applications of ANNs in pre-processing can be found in image restoration [9]. In general, one wants to restore an image that is distorted by the (physical) measurement system. The system might introduce noise, motion blur, out-of-focus blur, distortion caused by low resolution, or any kind of fault etc. Restoration can employ all information about the nature of the distortions introduced. In the most basic image restoration approach, noise is removed from an image by simple filtering. One of the most important applications of removed noise is feature extraction and then perform classification. The feature is extracted by morphological operations and classification is done by back-propagation neural network. For both these steps, ANNs have been used. The conclusion must be that ANNs can play a role in image processing, although it might be a role as a supporting tool rather than a major one. ANNs are useful in image processing as either non-parametric classi1ers, non-linear regression functions, or for (un)supervised feature extraction.

III. RESEARCH METHODOLOGY

It is generally believed that we human beings put different emphasis on different parts of a face e.g. eyes, nose, cheeks, forehead and other remaining parts. The existing approaches put same emphasize on all the parts of a face resulting in low recognition rate. In the approach, four different observers are selected – two eyes, nose, mouth and remaining portion of the face assuming that the eye coordinates are known. We then pass these patches (except the eyes) observed by different viewers through low pass (Gaussian) filter so as to smooth several parts of the image and reduce the effect of noises. The observed patches by different viewers are then combined into a single image vector. This image vector is then used as an input to the Artificial Neural Network and network is trained to recognize all the faces in image databases. There are many efficient algorithms using which can implement ANN, but in this paper our approaches to vision processing, the morphological shared weight neural network. The modal consist of two network stages: the first stage extracts feature using morphological operations; the second stage performs classification of outputs from the last feature extraction layer. this combination gives MSNN the ability to learn feature extraction and perform classification at the same time.

Morphological operation is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull [2].

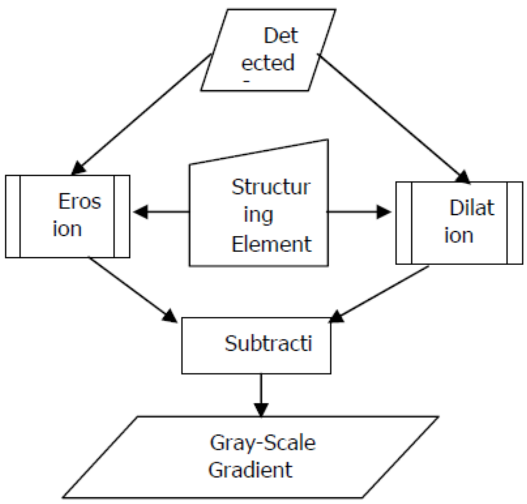

Fig2: Diagram of feature extraction

It is about adding or removing pixels from a binary image according to certain rules depending on neighborhood patterns. Dilation, erosion, closing, and opening are the more common morphological operations. As the names indicate, a dilation operation enlarges a region, while erosion makes it smaller.

Erosion and Dilation: Let Bx denote the translation of B so that its origin is located at x. The erosion of X by B is defined as the set of all points x such that Bx is included in *X*[1]:

$$X \ominus B = \{x \mid Bx \subset X\}$$

At each position where every 1-pixel of the structuring element (B) covers a 1-pixel of the binary image(*X*), the binary image pixel corresponding to the origin of the structuring element is ORed to the output image.

Let Bx denote the translation of B so that its origin is located at x. The dilation of X by B is deined as the set of all points x such that Bx hits X - that is, they have a nonempty intersection

$$X \oplus B = \{x \mid Bx \cap X \neq \Phi\}$$

Each time the origin of the structuring element (B) touches a binary 1-pixel (X), the entire translated structuring element shape is ORed to the output image which has been initialized to all zeros.

Dilation and erosion can be combined to achieve a variety of effects. For instance, subtracting an eroded image from its dilated version produces a "morphological gradient" which is a measure of local gray level variation in the image

$$\text{Morphological gradient} = X \oplus B) - (X \ominus B)$$

Opening and Closing: Dilation and erosion are often used in combination to detect sub-images or image components. The definition of a morphological opening of an image is erosion followed by dilation, using the same structuring element for both operations [1]:

$$X_B = (X \ominus B) \oplus B$$

The morphological closing of an image is the reverse: it consists of a dilation followed by an erosion with the same structuring element

$$X^B = (X \oplus B) \ominus B$$

Hit-and-Miss Operation: The hit-and-miss operation [1] is another form of dilation-erosion-based convolution. It is a morphological shape detector that can be used to look for particular patterns of foreground and background pixels on an image. This logical process ANDs an image to a structuring element. A hit-and-miss operation has the following equation [1]:

$$F \otimes K = (F \ominus K1) \cap (F^C \ominus K_2)$$
$$K_1 \cap K_2 = \Phi, K_1 \in K, K_2 \in K$$

Image *F* is first eroded with a structuring element K_1. The image's complement is then eroded with a second structuring element K_2. This operation preserves pixels whose neighborhood fits the shape of K_1 but doesn't fit the shape of K_2, K_1 and K_2 represent the shape of interest and the background respectively.

Back propagation network: The back-propagation consists of two passes in every layer of the neural network, a feed forward and a backward pass. In the feed forward pass, the input is applied to the neural network and is passed through the different layers. At this stage the weights are fixed and do not change. At the neural output, an error is recorded, which is the difference weight the output and desired response.

During the backward pass the weight and biases are adjusted in accordance with error-correction rules. The error signal is propagate backward and the parameters are adjusted layer by layer until all the layers are covered [12][13].

Apply the Forward Propagation with current w_{ji} & w_{kj}.

- Determine the output neuron k using the following step.

For $j = 0$ to *number of neuron at Hidden layer*
  $net_j = 0$
  For $i = 0$ to *number of neuron at Input layer*
    $net_j = net_j + (w_{ji} \times O_i + \theta_j)$

$$net_j = \sum_i w_{ji} O_i + \theta_j \qquad \text{.....3.1}$$

  Next i
  $O_j = 1 / 1 + e^{-net_j}$

$$O_j = f(net_j) = \frac{1}{1 + e^{-net_j}} \qquad \text{......3.2}$$

Next j

- Determine the output neuron k using the following step.

For $k = 0$ to *number of neuron at Output layer*
  $net_k = 0$
  For $j = 0$ to *number of neuron at Hidden layer*
    $net_k = net_k + (w_{kj} \times O_j + \theta_k)$

$$net_j = \sum_j w_{kj} O_j + \theta_k \qquad \text{......3.3}$$

  Next j
  $O_k = 1 / 1 + e^{-net_k}$

$$O_k = f(net_k) = \frac{1}{1 + e^{-net_k}} \qquad \text{......3.4}$$

Next k

For $k = 0$ to *number of neuron Output layer*
  $E_k = T_k - O_k$
Next k 3.5

Fig.3. Forward Propagation Algorithm

During the feed forward, each input neuron receives an input signal. Each input neuron transmits the signal to each hidden neuron (equation 3.1) that applies the activation function and passes it to each output layer (equation 3.2). Each output layer applies the activation function to obtain the network output (equation 3.3 and 3.4). The network finally compared the target value and the error is obtained (equation 3.5). The algorithm of forward propagation is shown in Figure 3.

- Compute the error signal between the output & hidden layer

For $k = 0$ to *number of neuron at Output Layer*
  $\delta_k = O_k (1 - O_k)(t_k - O_k)$
Next k ...4.1

- Compute the error signal through layer i :

For $j = 0$ to *number of neuron at Hidden layer*
$temp = 0$
  For $k = 0$ to *number of neuron at Output layer*
    $temp = temp + (\delta_k \times w_{kj})$

  Next k
$\delta_j = O_j (1 - O_j) temp \qquad \delta_j = O_j (1 - O_j) \sum_k \delta_k w_{kj}$
Next j ...4.2

Fig.4. Backward Propagation Algorithm

During back-propagation method back-propagation of the associated error until the network is generalized acceptable, the previous error (equation 3.5) is obtained. If the error unacceptable, the error signal at output layer (equation 4.1) and input layer (equation 4.2) is determine. Since the backward propagation is applied, the previous weights are updated simultaneously to minimize the network error.

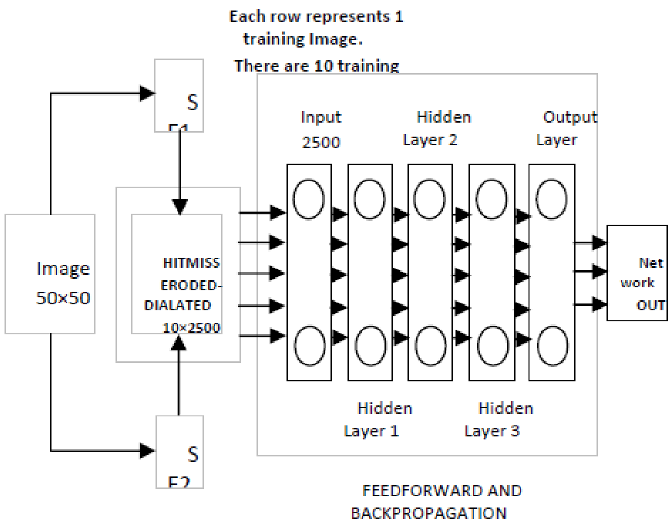

Fig5: Combination of morphological process and back propagation algorithm

In the initial stage, the MSNN performs a combination of grayscale erosion and dilation known as the hit-miss transform. Each input image is eroded by a hit structuring element and dilated by a miss structuring element separately. Both outputs are then subtracted to derive their overlapping difference. The result from this process forms the feature map, which becomes the direct input to a back propagation network as shown in fig5. 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.

IV. CONCLUSION

So, from the above discussion it has been concluded that the use of Morphological shared weight neural network is effective and efficient for face recognition. 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. Since the digital image processing has grown up as a most preferable topic for research now-a-days, so there is lot of scope & possibility of further improvements. The work can be extended to include other recognition procedures such as face from a group photograph.

REFERENCES

[1] Rafael C. Gonzalez and Richard E Woods, “Digital Image Processing”, Person Education Asia,2005.
[2] Rafael C. Gonzalez, Richard E. Woods, Stevens L. Eddins, “Digital Image Processing using MATLAB”.
[3] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the microstructure of Cognition, Vol. I, MIT Press, Cambridge, 1986, pp. 319–362.
[4] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990.
[5] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognition 26 (9) (1993) 1277– 1294.
[6] M. Egmont-Petersen, D. de Ridder, H. Handels, Image processing with neural networks—a review, Pattern Recognition 35 (2002) 2279–2301
[7] R.R. Meyer, E. Heindl, Reconstruction of oI-axis electron holograms using a neural net, J. Microsc. 191 (1) (1998) 52–59.
[8] Y.M. Wang, F.M. Wahl, Vector-entropy optimizationbased neural-network approach to image reconstruction from projections, IEEE Trans. Neural Networks 8 (5) (1997) 1008–1014.
[9] D. de Ridder, R.P.W. Duin, P.W. Verbeek et al., The applicability of neural networks to non-linear image processing, Pattern Anal. Appl. 2 (2) (1999) 111–128.
[10] 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.
[11] Latha, Dr.L.Ganesan & Dr.S.Annadurai,”Face Recognition using Neural network”
[12] “ Face Recognition using PCA and Backpropagation Neural Network”march2011.
[13] S.Haykin, “Neural Networks-A Comprehensive Foundation”,prentice Hall Inc.,New jersey,1999
[14] Pete Mc Collum, “An Introduction to Back Propagation Neural Networks”, The Newsletter of the Seattle Robotics Society.
[15] Andrew Blais and David Mertz, “An Introduction to Neural Networks – Pattern Learning with Back Propagation Algorithm”, Gnosis Software, Inc., July 2001.
[16] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990.
[17] M. Egmont-Petersen, D. de Ridder, H. Handels, Image processing with neural networks—a review, Pattern Recognition 35 (2002) 2279–2301

Sumedha student of Amity University Noida, pursuing Masters in Computer Science & Technology. having 8 years of experience in the field of academics.

Monika Jenna working as faculty in department of computer science & technology at Amity College of Engineering & Technology, Noida. Presently doing research work in the field of computer science from Amity University.