SIMPLIFIED FUZZY ARTMAP AS PATTERN RECOGNIZER

By S. Rajasekaran,[1] Member, ASCE, and G. A. Vijayalakshmi Pai[2]

ABSTRACT: Pattern recognition has turned out to be an important aspect of a dominant technology such as machine intelligence. Domain specific fuzzy-neuro models particularly for the ''black box'' implementation of PR applications have been recently investigated. In this paper, Kasuba's simplified fuzzy adaptive resonance theory map (ARTMAP) has been discussed as a pattern recognizer/classifier for image processing problems. The model inherently recognizes only noise free patterns and in the case of patterns with noise or perturbations (rotation/scaling/translation) misclassifies the images. To tackle this problem, a conventional moment based rotation/scaling/translation invariant feature extractor has been employed. However, since the conventional feature extractor is not strictly invariant to most perturbations, certain mathematical modifications have been proposed that have resulted in an excellent performance by the pattern recognizer. The potential of the model has been demonstrated on two problems, namely, prediction of load from the yield patterns of elastoplastic analysis of clamped and simply supported plates and prediction of modes from mode shapes.

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

INTRODUCTION

Pattern recognition (PR) is a science that concerns the description or classification (recognition) of measurements and is often an important component of intelligent systems. Thus, PR has turned out to be a useful and rapidly developing technology with cross-disciplinary interest and contribution.

PR techniques overlap with areas such as structural modeling, optimization and estimation theory, and neuro-fuzzy systems. PR applications include seismic analysis, radar signal analysis, computer vision, face recognition, speech recognition, electrocardiogram signal analysis, and handwritten and character recognition. In the area of civil engineering, diagnostic analysis of crack patterns (Chao and Cheng 1998), structural damage assessment (Ishizuka et al. 1982; Watada 1984), and examination of soil cracking patterns are some of the problems on which PR techniques have been applied.

Although neural networks (NN) and fuzzy logic have individually presented themselves as potential candidates of several approaches well suited for pattern classification applications, hybrid systems that combine NN and fuzzy logic have also been investigated in PR. Kosko's fuzzy associative memory (1992) has been used to represent fuzzy association. Yamakawa and Tomoda (1989) proposed a simple fuzzy neuron model for application in character recognition but failed to describe the specific learning algorithm. Takagi et al. (1990) suggested a structured NN with the help of inference rules, but the network demands complicated training despite its better performance. Machada and Rocha (1992) constructed a fuzzy connectionist expert system using a combinatorial NN model. Kwan and Cai (1994) presented a four-layer feedforward fuzzy NN but with limited recognition capability particularly with regard to translation of patterns.

A general survey of the models in the literature indicate that most models are applicable or have been discussed only for monochrome (binary) images or patterns and employ either a complex architecture or adopt lengthy training sessions. Furthermore, the capability of the system to exhibit tolerance to noise and pattern perturbations leaves much to be desired.

Therefore, the aim of this paper is to build a pattern recognizer based on a fuzzy-neuro model and exhibit the following characteristics:

1. A less complex but powerful fuzzy-neuro architecture
2. Significant reduction in training
3. Recognition of both monochrome and color images
4. Tolerance to pattern perturbations (e.g., rotation, translation, and scaling)
5. Tolerance to random noise

Kasuba's simplified fuzzy ARTMAP (SFAM) (1993), which is a vast simplification of Carpenter et al.'s fuzzy ARTMAP (1992), was chosen to function as the core architecture. When compared to its predecessor, SFAM has reduced computational overhead and architectural redundancy. Also, the model employs simple learning equations with a single user selectable parameter and can learn every single training pattern with only a handful of training iterations.

Although SFAM inherently exhibits characteristics (1) and (2) listed above, experiments (Rajasekaran and Vijayalakshmi Pai 1997; Rajasekaran et al. 1997) revealed that the characteristics (4) and (5) were almost nil, and the model displayed the capability to recognize/classify patterns that are only absolutely free of noise or perturbations.

To overcome this problem, a careful choice of invariant features extracted from the patterns needs to be fed into SFAM as ''preprocessed'' input. Digital approximations of moment invariants (Schalkoff 1989) have been employed for this purpose. However, these approximations, which are invariant to translation (T) of the patterns, are not strictly invariant to rotation (R) and scaling (S) changes. The properties therefore had to be mathematically modified before being put to use.

In this paper, a neuro-fuzzy pattern recognizer is discussed. First, Kasuba's SFAM and the learning equations are presented in brief. Next, the conventional moment invariants and the modifications carried out for the successful feature extraction are discussed. Finally, the structure of the pattern recognizer and application of the architecture to two problems chosen from civil engineering, namely, prediction of load from the yield patterns of elastoplastic analysis of clamped and simply supported plates and prediction of modes from mode shapes, is discussed.

SFAM REVIEW

SFAM is essentially a two layer net containing an input and an output layer (Fig. 1).

[1]Prof. and Head, Dept. of Civ. Engrg., PSG Coll. of Technol., Coimbatore, Tamil Nadu 641 004, India. E-mail: sekaran@hotmail.com

[2]Sr. Grade Lect., Dept. of Mathematics and Comp. Applications, PSG Coll. of Technol., Coimbatore, Tamil Nadu 641 004, India. E-mail: vijipai@vsnl.com

Note. Editor: Sivand Lakmazaheri. Discussion open until September 1, 2000. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on June 24, 1998. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 14, No. 2, April, 2000. ©ASCE, ISSN 0887-3801/00/0002-0092–0099/$8.00 + $.50 per page. Paper No. 18628.

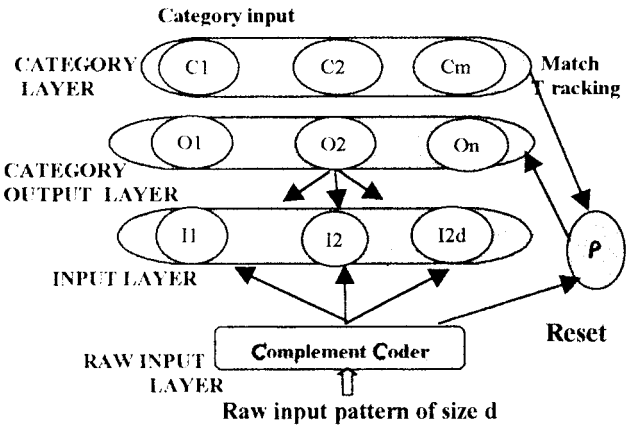

FIG. 1. Architecture of SFAM Network

Input Normalization

NN are in general very sensitive to absolute magnitude, and fluctuations in inputs may tend to swamp the performance of the network while predicting the desired outputs. Hence the need for normalization of inputs so that the inputs correspond to the same range of values.

Complement coding is a normalization rule for preserving amplitude information. Thus it represents the on-response and the off-response to an input vector. It also represents the presence of a particular feature in the input pattern and its absence. For example, if **a** is the given input pattern vector of *d* features, the complement coded vector **a** represents the absence of each feature, where $\mathbf{a}'$ is defined as

$$\mathbf{a}' = 1 - \mathbf{a} \tag{1}$$

The previous equation is valid because, just as in fuzzy logic, all SFAM input values must be within the range of 0–1. Therefore, the complement coded input vector **I** internal to SFAM is given by the 2D vector

$$\mathbf{I} = (a \cdot a') = (a_1, a_2, \ldots, a_d, a_1', a_2', \ldots, a_d') \tag{2}$$

Output Node Activation

When SFAM is presented as an input pattern whose complement coded representation is **I**, all output nodes become active to some degree. This output activation denoted by T_j for the *j*th output node, where W_j is the corresponding weight, is given by

$$T_j(I) = \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{\alpha + |\mathbf{W}_j|} \tag{3}$$

Here, $\wedge$ indicates the fuzzy AND operator defined by

$$\mathbf{a} \wedge \mathbf{b} = \min(a, b) \tag{4}$$

where **a** and **b** = fuzzy vectors. Here, α is kept as a small value close to 0, usually about 0.0000001. The winning output node is the node with the highest activation function T_j. If more than one T_j is maximal, the output node *j* with the smallest index is arbitrarily chosen to break the tie. The category associated with the winning output node is described as the network's classification of the current input pattern.

The match function given below helps determine if learning should occur

$$\frac{|\mathbf{I} \wedge \mathbf{W}_j|}{|\mathbf{I}|} \tag{5}$$

When used in conjunction with the vigilance parameter, the match function value states whether the current input is a good enough match to a particular output node to be encoded by that output node, or instead, if a new output node should be formed to encode the input pattern.

If the match function value is greater than the vigilance parameter, the network is said to be in a state of resonance. Resonance means that output node *j* is good enough to encode input *I* provided that output node *j* represents the same category as input *I*.

A network state called ''mismatch reset'' occurs if the match function is less than vigilance. This state indicates that the current output node does not meet the encoding granularity represented by the vigilance parameter and therefore cannot update its weights even if the input patterns' category is equal to the category of the winning output node.

Once a winning output node *j* has been selected to learn a particular input pattern *I*, the top down weight vector $\mathbf{W}_j$ from the output node is updated according to the equation

$$\mathbf{W}_j^{\text{new}} = \beta(I \wedge \mathbf{W}_j^{\text{old}}) + (1 - \beta)\mathbf{W}_j^{\text{old}} \tag{6}$$

where $0 < \beta \leq 1$. Once SFAM has been trained, the equivalent of a ''feedforward'' pass for an unknown pattern classification consists of passing the input pattern through the complement coder and into the input layer. The output node activation function is evaluated and the winner is the one with the highest value. The category of the input pattern is the one with which the winning output node is associated.

WORKING OF SFAM

The circle in the square problem requires a system to identify which pockets of a square lie inside and which lie outside a circle whose area equals half that of the square. This problem, specified as a benchmark problem for system performance evaluation in the defense advanced research projects agency (DARPA) artificial neural network technology program, has been used to illustrate the workings of SFAM. In this, SFAM is trained with a set of points for a definite number of training epochs. The inputs during training are the points (*x*, *y*) and the category to which they belong, namely, inside the circle (IN) or outside the circle (OUT). The training of SFAM has been illustrated in the following examples.

Example 1

Consider the point (0.7, 0.7) as input *I* and whose category is IN.
Complement of *I* = (0.3, 0.3)
Augmented input *I* = (0.7, 0.7, 0.3, 0.3)

Because *I* is the first input seen by SFAM in the IN category, the top down weights $W_i = (W_x, W_y, W_{x^c}, W_{y^c})$ are set to the augmented input values and are therefore given by W_1 = (0.7, 0.7, 0.3, 0.3). The activation function value $T_1(I)$ of the top down weight node W_1 for the input *I* shown is set to null and is set to point to the category IN in the category layer.

Example 2

I = (0.3, 0.8) and category = IN
Augmented input *I* = (0.3, 0.8, 0.7, 0.2)
Activation function $T_1(I)$ = 0.7499 (choosing α = 0.0000001)
Match function value = $MF(I)$ = 0.75

Here, $MF(I)$ is greater than the vigilance parameter ρ (chosen as 0.5), and because the category of *I* is the same as that pointed to by W_1, W_1 is fit enough to learn the current input *I*. This is accomplished by updating the weights of W_1 as il-

J. Comput. Civ. Eng., 2000, 14(2): 92-99

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

lustrated in (6), choosing $\beta = 1$ [i.e., $W_1^{new} = (0.3, 0.7, 0.3, 0.2)$].

Example 3

$I = (0.9, 0.9)$ and category = OUT

Because this is a new category, $W_2 = (0.9, 0.9, 0.1, 0.1)$ and $T_2(I) =$ null.

Example 4

$I = (0.7, 0.9)$ and category = OUT
Augmented input $I = (0.7, 0.9, 0.3, 0.1)$

Because there are two weight nodes to decide which node is fit enough to learn the new input I, the activation function values of the two nodes are computed [i.e., $T_1(I) = 0.9333$ and $T_2(I) = 0.8999$]. The node with the highest activation function value, W_1 in this case, is chosen to learn the new input. Also, the match function of I with W_1 is greater than the vigilance parameter [i.e., $(MF(I) = 0.7) > (\rho = 0.5)$]. However, due to category mismatch, W_1 is not fit to learn the input, and hence the next node is to be considered. Before proceeding to the next node, match tracking is done by updating the vigilance parameter by a small quantity (i.e., $\rho = 0.501$). The next node W_2 gives $MF(I) = 0.9$. Because $MF(I) > \rho$ and the categories are also the same, learning occurs in W_2 given by the updating of W_2 as $W_2^{new} = (0.7, 0.9, 0.1, 0.1)$.

Example 5

$I = (0.1, 0.3)$ and category = IN
Augmented input $I = (0.1, 0.3, 0.9, 0.7)$
Activation function values of W_1 and W_2 are $T_1(I) = 0.5999$ and $T_2(I) = 0.33$

Choosing the highest activation function value, namely, W_1, the match function yields $MF(I) = 0.45$, which is less than $\rho = 0.501$, rendering the node to be unfit to learn the pattern I. Choice of W_2 also results in a similar case with $MF(I) = 0.3$, which is less than $\rho = 0.501$. In such a case, a new top down weight node W_3 pointing to IN is created with $W_3 = (0.1, 0.3, 0.9, 0.7)$.

During inference, SFAM is presented with points (x, y) alone to determine the category. In this case, the top down weight node that reports the highest activation function value for the given (x, y) is the winner, and the category pointed to by the node is the category that (x, y) belongs to.

FEATURE EXTRACTION—MOMENT BASED INVARIANTS

The classification of 2D objects from visual image data is an important pattern recognition task. This task exemplifies many aspects of a typical PR problem, including feature selection, dimensionality reduction, and the use of qualitative descriptors. Moments are extracted features that are derived from raw measurements. In practical imagery, various geometric distortions or pattern perturbations may be observed in the image to be classified. Fig. 2 illustrates some example pattern perturbations. It is therefore essential that features that are invariant to orientations be used for the classification purpose. For 2D images, moments have been used to achieve R, S, and T invariants. Properties of invariance to R, S, and T transformations may be derived using function of moments.

The moment transformation of an image function $f(x, y)$ is given by

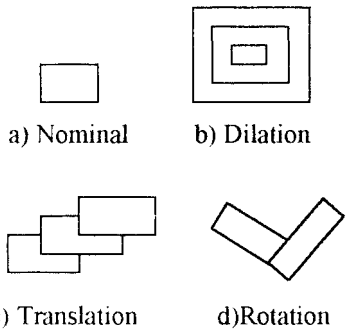

FIG. 2. Example Pattern Perturbations

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y)\, dx\, dy, \quad p, q = 0, 1, 2, \ldots, \infty \tag{7}$$

However, in the case of a spatially discretized $M \times N$ image denoted by $f(i, j)$, (7) is formulated using an approximation of double summations

$$m_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^p j^q f(i, j) \tag{8}$$

The so-called central moments are given by

$$\mu_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \hat{i})^p (j - \hat{j})^q f(i, j) \tag{9}$$

where

$$\hat{i} = \frac{m_{10}}{m_{00}}; \quad \hat{j} = \frac{m_{01}}{m_{00}} \tag{10a,b}$$

The central moments are still sensitive to R and S transformations. The S invariant may be obtained by further normalizing μ_{pq} or by forming

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{[(p+q)/2+1]}}, \quad p + q = 2, 3, \ldots \tag{11}$$

From (11), constraining $p, q \leq 3$, and using the tools of invariant algebra, a set of seven RST invariant features may be derived (Schalkoff 1989)

$$\phi_1 = \eta_{20} + \eta_{02} \tag{12}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{13}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{14}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{15}$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \tag{16}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{17}$$

$$\phi_7 = 3(\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \tag{18}$$

Thus, for a common PR application, which involves recognition of 2D images, the moment based invariant functions ϕ_1–ϕ_7 are computed from the raw measurements of every given shape. Those shapes with similar ϕ_i are classified as belonging to the same class. In other words, shapes belonging to a class are the perturbed (rotated/scaled/translated) or exact version of the given nominal shape.

However, though the set of invariant moments shown are

J. Comput. Civ. Eng., 2000, 14(2): 92-99

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

invariant to T, despite their being computed discretely, the moments cannot be expected to be strictly invariant under R and S changes due to the ''non 1:1 nature of general discrete geometric transformations (including rotation and scale changes)'' (Schalkoff 1989).

An investigation by the writers revealed that in the definition of μ_{pq}, the contribution made by a pixel had been overlooked. The modified μ_{pq} definitions will be discussed in the succeeding section.

μ_{pq} REDEFINED

Visual image data can be either binary (for monochrome images) or real (for color images). For example, in Fig. 3 the region R appears in the binary image as region of intensity ''1.'' If we represent an image function as $f(x, y)$, then $f(x, y)$ is either 0 or 1 depending on whether the pixel is dark or bright. On the other hand, in Fig. 4 the intensity is represented by various shades with $0 \leq f(x, y) \leq 1$, indicating that the density lies anywhere between the ends of a spectrum: very dark to very bright. This is how color images are represented. However, the image function $f(x, y)$ is constant over any pixel region.

Moments are descriptive techniques with an intuitive basis in the study of mechanics of bodies. Consider the real image

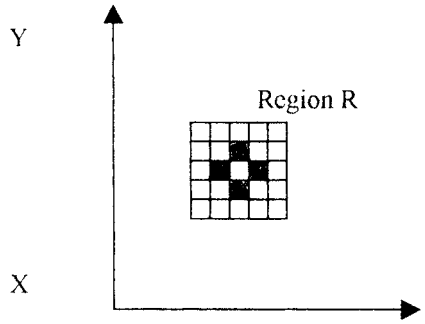

FIG. 3. Monochrome Image

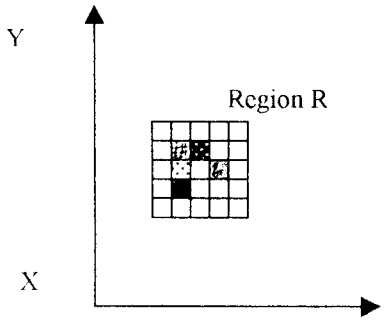

FIG. 4. Color Image

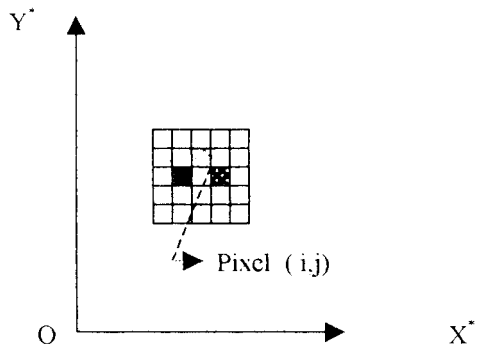

FIG. 5. Pixel (i, j) in Image

as shown in Fig. 5. Here X^*, Y^* are the axes passing through the origin O. The dimension of any pixel is taken as a 1×1 square. Considering the unit thickness of the lamina, the image function is analogous to mass density function. Total mass of an image is given as

$$M = \int \int f(x^*, y^*)\, dx^*\, dy^* \tag{19}$$

and the mass of pixel

$$(i, j) = f(x_i^*, y_j^*) \tag{20}$$

$$M = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i^*, y_j^*) \tag{21}$$

assuming any figure is represented in a square region of $n \times n$ pixels. In general, the moment transform of an image function $f(x, y)$ is given by

$$m_{pq} = \int \int x^{*p} y^{*q} f(x^*, y^*)\, dx^*\, dy^* \tag{22}$$

m_{10} which represents the first moment of the mass of the image about the Y^*-axis is given by

$$m_{10} = \int \int x^* f(x^*, y^*)\, dx^*\, dy^* \tag{23a}$$

$$m_{10} \text{ of pixel } (i, j) = f(x_i^*, y_j^*) \int \int x^*\, dx^*\, dy^* \tag{23b}$$

From Fig. 6

$$x^* = x_i^* + x \tag{24}$$

where x_i^* is the distance of the center of gravity (CG) of the pixel (i, j) from the Y^*-axis. Because $dx^* = dx$

$$m_{10} \text{ of pixel } (i, j) = f(x_i^*, y_j^*) \int (x_i^* + x)\, dx \tag{25a}$$

$$m_{10} \text{ of pixel } (i, j) = f(x_i^*, y_j^*) x_i^*, \quad \because \int x\, dx = 0 \tag{25b}$$

In the discrete case

$$m_{10} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i^*, y_j^*) x_i^* = M\bar{x}_C^* \tag{26}$$

And similarly

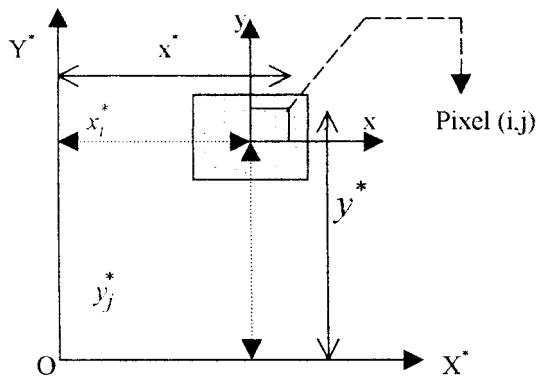

FIG. 6. Pixel (i, j) Represented in X^*, Y^*-Axes

J. Comput. Civ. Eng., 2000, 14(2): 92-99

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

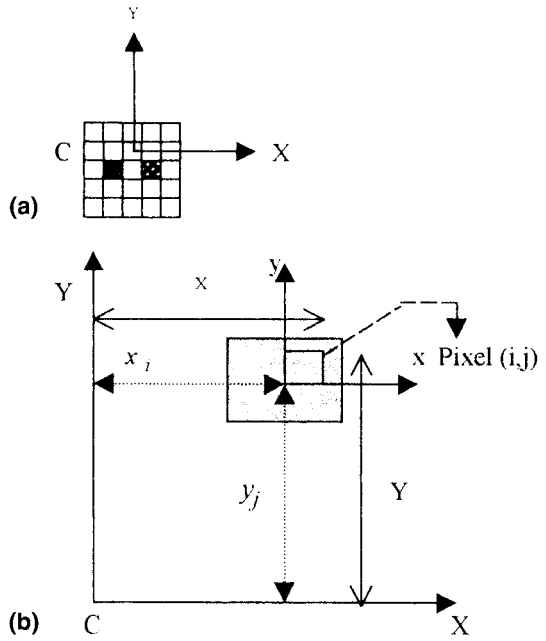

FIG. 7. (a) *XY* Axes Passing through Center of Mass (*C*) of Image; (b) Pixel (*i*, *j*) Represented in *X*, *Y*-Axes

$$m_{01} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i^*, y_j^*) y_j^* = M\bar{y}_C^* \tag{27}$$

where $\bar{x}_C^*$ and $\bar{y}_C^*$ denote the coordinates of the mass center of the image with respect to the origin. Now consider the *X*, *Y*-axes passing through the center of the mass of the image as shown in Fig. 7. Variables x_i, y_j represent the distance of the center of the pixel (i, j) from the centroidal axes. Now define the central moments μ_{pq} as

$$\mu_{pq} = \int \int X^p Y^q f(X, Y)\, dX\, dY \tag{28}$$

where $\mu_{00} = m_{00} = M$, the central mass of the image. Also, $\mu_{10} = \mu_{01} = 0$, since the axes pass through the center of mass of the image.

The derivations of μ_{pq}, $p, q \leq 3$ are presented next.

μ_{20}, μ_{02}: Derivation

$$\mu_{20} = \int \int X^2 f(X, Y)\, dX\, dY \tag{29}$$

where $X = x_i + x$.

$$\mu_{20} \text{ of pixel } (i, j) = f(x_i, y_j) \int_{-(1/2)}^{1/2} dy \int_{-(1/2)}^{1/2} (x_i + x)^2\, dx \tag{30}$$

$$\mu_{20} \text{ of pixel } (i, j) = f(x_i, y_j) \left(x_i^2 + \frac{1}{12}\right), \quad \because \int x\, dx = 0 \tag{31}$$

Thus

$$\mu_{20} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j) \left(x_i^2 + \frac{1}{12}\right)$$

and is physically expressed as the ''horizontal centralness of the mass of the image.'' Similarly

$$\mu_{02} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j) \left(y_j^2 + \frac{1}{12}\right)$$

and physically expresses the ''vertical centralness of the mass of the image.''

μ_{11}: Derivation

Now

$$\mu_{11} = \int \int XYf(X, Y)\, dX\, dY \tag{32}$$

$$\mu_{11} \text{ of pixel } (i, j) = f(x_i, y_j) \int \int (x_i + x)(y_j + y)\, dx\, dy \tag{33}$$

$$\mu_{11} \text{ of pixel } (i, j) = f(x_i, y_j)(x_i y_j), \quad \because \begin{cases} \int y\, dy = 0 \\ \int x\, dx = 0 \\ \int \int xy\, dx\, dy = 0 \end{cases} \tag{34}$$

μ_{11} physically means the ''diagonal'' indication of the quadrant with respect to the centroid where the region has more mass.

μ_{30}, μ_{03}: Derivation

$$\mu_{30} = \int \int X^3 f(X, Y)\, dX\, dY \tag{35}$$

$$\mu_{30} \text{ for a pixel } (i, j) = f(x_i, y_j) \int \int (x_i + x)^3\, dx\, dy \tag{36}$$

$$\mu_{30} \text{ for a pixel } (i, j) = f(x_i, y_j)(x_i^3) \tag{37}$$

Thus

$$\mu_{30} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i^3)$$

and represents physically the ''horizontal imbalance'' of the location of the CG with respect to half horizontal extent. Similarly

$$\mu_{03} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(y_j^3)$$

and represents the ''vertical imbalance'' of the location of the CG of mass with respect to half vertical extent.

μ_{21}, μ_{12}: Derivation

$$\mu_{21} = \int \int X^2 Yf(X, Y)\, dX\, dY \tag{38}$$

$$\mu_{21} \text{ for a pixel } (i, j) = f(x_i, y_j) \int \int (x_i + x)^2 (y_i + y)\, dx\, dy \tag{39a}$$

$$\mu_{21} \text{ for a pixel } (i, j) = f(x_i, y_j)(x_i^2 y_j) \tag{39b}$$

In the discrete case

$$\mu_{21} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i^2 y_j)$$

and similarly

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

$$\mu_{12} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i y_j^2)$$

Here, μ_{21} represents the ''vertical divergence'' indicating the relative extent of the bottom of the region compared with the top, and μ_{12} represents the ''horizontal divergence'' indicating the relative extent of the left of the region compared with the right.

Summing up, the central moments are presented below

$$\mu_{00} = M \text{ (Mass)} \tag{40}$$

$$\mu_{10} = 0 \tag{41}$$

$$\mu_{01} = 0 \tag{42}$$

$$\mu_{20} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j) \left(x_i^2 + \frac{1}{12}\right) \tag{43}$$

$$\mu_{02} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j) \left(y_j^2 + \frac{1}{12}\right) \tag{44}$$

$$\mu_{21} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i^2 y_j) \tag{45}$$

$$\mu_{12} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i y_j^2) \tag{46}$$

$$\mu_{11} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i y_j) \tag{47}$$

$$\mu_{30} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(x_i^3) \tag{48}$$

$$\mu_{03} = \sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j)(y_j^3) \tag{49}$$

The moments listed above are still sensitive to R and S transformations. Note that the S invariance may be obtained by further normalizing μ_{pq} as given in (11).

From (40)–(49) it may be observed that μ_{20} and μ_{02} are different from their conventional definitions of central moments. In the conventional definition of μ_{20}, for example, the term

$$\sum_{j=1}^{n} \sum_{i=1}^{n} f(x_i, y_j) \left(\frac{1}{12}\right)$$

has been neglected. This omission has resulted in a cascading effect rendering η_{20} and η_{02} and the functions ϕ_1, ϕ_2, and ϕ_6 incorrectly defined, leading to the misclassification of images with RST orientations.

Using the normalized central moments and tools of invariant algebra, a set of seven RST invariant features, similar to those shown in (12)–(18), may be derived. Note that ϕ_7 is actually ST invariant and changes its sign for reflection. This is because for reflection about the *Y*-axis, let us say x becomes $-x$, for odd p, η_{pq} (i.e., η_{30}, η_{12}) changes sign, and hence ϕ_7 changes sign for reflection.

STRUCTURE OF SFAM BASED PATTERN RECOGNIZER

The overall structure of the pattern recognizer is as illustrated in Fig. 8. The images (patterns) whether monochrome or color are input through the image processor. In this work, the images are engraved on a grid size of 40 × 40. Also, images may have their colors chosen from a fixed palette. Here the colors that are fixed are white, black, green, yellow, and magenta. However, any number of colors could be used to define the palette. The image processor receives the images that are to be trained (training patterns) or inferred by SFAM (inference patterns).

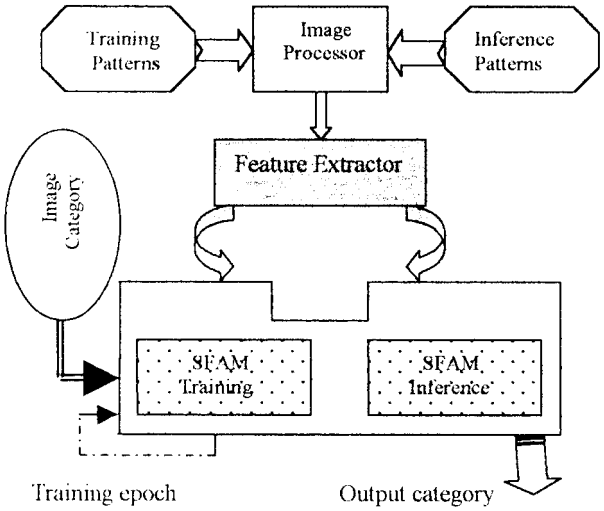

FIG. 8. Structure of SFAM Based Pattern Recognizer

The feature extractor obtains the RST invariant features for each image, whether for training or inference. From the images surrendered for training or inference, the seven moment based invariant functions, ϕ_1–ϕ_7, are extracted by the feature extractor for presentation to SFAM. Thus, SFAM works only on the feature vectors of the image and not on whole images. The SFAM activator functions as two modules: the training module (SFAM training) and the inference model (SFAM inference). The feature vectors of the training patterns and the categories to which they belong (image category) are presented to the SFAM's training module. The only user selectable parameter for the training session is the vigilance parameter ρ ($0 < \rho < 1$). Once the training is complete, the top down weight vectors represent the patterns learned. Next, the feature vectors of the images that are to be recognized/classified are presented to the inference module. SFAM now begins its classification of images by associating the feature vectors with the top down weight vectors. The category of the image is output by SFAM.

The system can handle both symmetric and asymmetric patterns. However, in the case of symmetric patterns it is essential that only distinct portions of the images be trained. This is because in the case of doubly symmetric patterns, their RST invariant feature vectors $\boldsymbol{\phi}_2$–$\boldsymbol{\phi}_7$ acquire values that are very close to 0, and ϕ_1 tends to 1. This consequently results in feature vectors, which are similar, leading to misclassification of patterns.

APPLICATION 1: PREDICTION OF LOAD FROM YIELD PATTERNS OF ELASTOPLASTIC, CLAMPED, SIMPLY SUPPORTED PLATES

Benjamin Whang (1969) developed a finite-element displacement method for elastoplastic analysis of bilinear strain hardening orthotropic plates and shells, also considering elastic unloading. The two basic approaches adopted by Whang in the elastoplastic analysis are the initial stiffness approach and tangent stiffness approach in conjunction with the Huber-Mises yield criterion, and the Prandtl-Reuss flow rule in accordance with the strain hardening yield function.

Fig. 9 shows the uniformly loaded isotropic plates, one with clamped edges and the other with simply supported edges. The following numerical values are used:

$t = 1.0$, $E = 30{,}000.0$, $\nu = 0.3$, $E_p = 300.0$, and $\sigma_0 = 30.0$

J. Comput. Civ. Eng., 2000, 14(2): 92-99

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

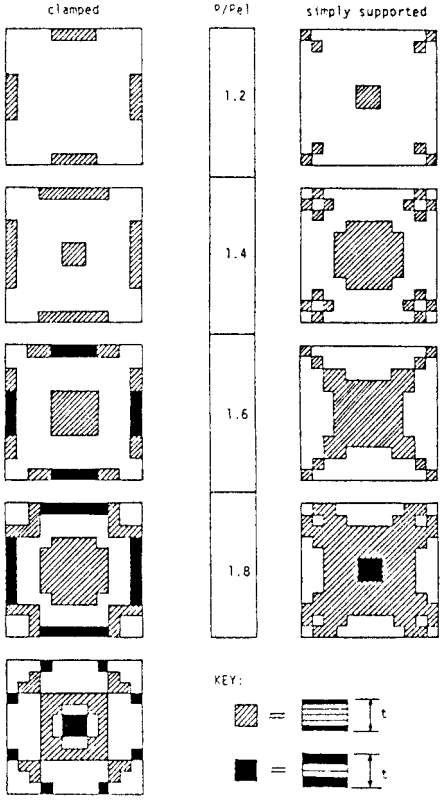

FIG. 9. Isotropic Plates: Clamped versus Simply Supported

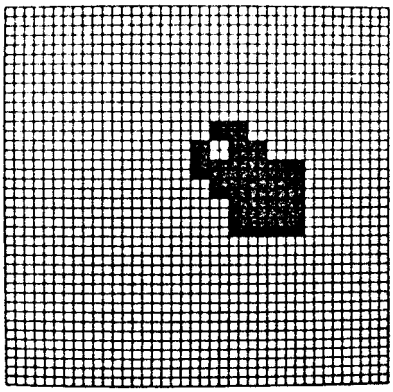

FIG. 10. Pattern as Presented to SFAM (Prediction of Load from Yield Patterns)

where t = thickness; E = Young's modulus; ν = Poisson's ratio; and σ_0 = yield stress. The formation of a plastic zone with respect to loading is also shown in the figure.

SFAM is trained with the patterns representing the plastic zones and their corresponding loading and tested for its inference capability. Considering the doubly symmetric nature of the patterns, only a quarter of the image is presented to SFAM for training. Fig. 10 illustrates a sample pattern for SFAM training. Observe that it is a quarter of the pattern corresponding to a loading of 1.8 of a simply supported plate. The prediction of load by SFAM is done quite correctly.

APPLICATION 2: PREDICTION OF NATURAL MODE SHAPES OF MULTISTORIED BUILDING FRAMES

In this case, the patterns representing the peak values of displacements for a five-story shear frame are presented as inputs to SFAM. Fig. 11 illustrates the peak values of displacements. A sample pattern presented to SFAM has been shown in Fig. 12. The model presented one exemplar for every

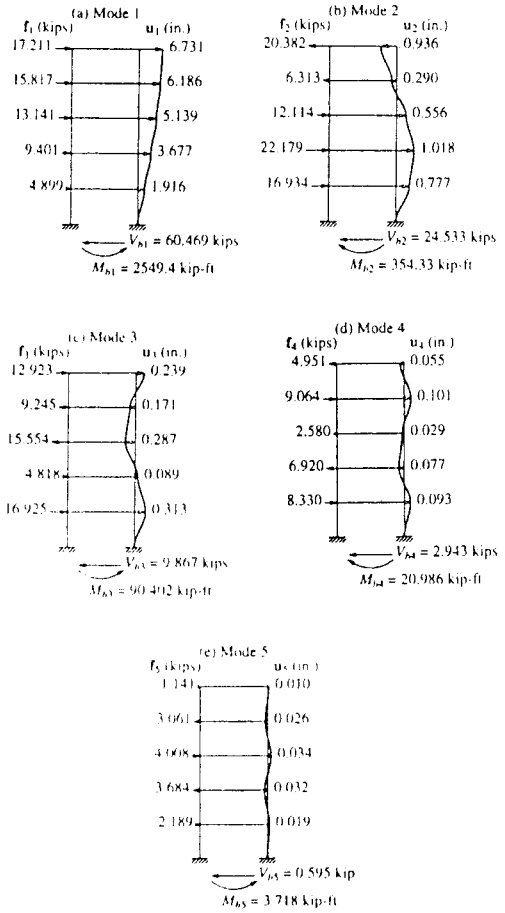

FIG. 11. Peak Values of Displacements

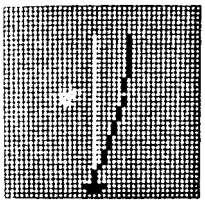

FIG. 12. Pattern as Presented to SFAM (Prediction of Natural Mode Shapes)

mode. The system is tested for noisy and noise free patterns. In all cases, the predictions are quite correct.

CONCLUSIONS

In this paper, a pattern recognizer based on SFAM has been discussed. The model employs a moment based RST invariant feature extractor, which has been modified to accommodate strict invariance to R, S, and T changes.

For classification of asymmetric patterns, the moment based feature extractor extracts unique feature vectors. But for doubly and multisymmetric patterns, the feature vectors turn out to be similar resulting in misclassification. Though the problem has been tackled by considering only a portion (quarter) of the pattern, further investigations could be undertaken to look for a feature extractor that is insensitive to symmetricity of patterns as well as their perturbations.

ACKNOWLEDGMENTS

The writers express their thanks to the management and Dr. P. Radhakrishnan, Principal, of PSG College of Technology, for providing the facilities to carry out this research work.

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.

APPENDIX I. REFERENCES

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B. (1992). ''Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps.'' *IEEE Trans. Neural Networks*, 3(5), 698–712.

Chao, C.-J., and Cheng, F.-P. (1998). ''Fuzzy pattern recognition model for diagnosing cracks in RC structures.'' *J. Comp. in Civ. Engrg.*, ASCE, 12(2), 111–119.

Ishizuka, M., Fu, K. S., and Yao, J. T. P. (1982). ''A rule based inference with fuzzy set for structural damage assessment.'' *Approximate reasoning in decision process*, M. M. Gupta and E. Sanchez, eds., North-Holland, Amsterdam, 261–275.

Kasuba, T. (1993). ''Simplified fuzzy ARTMAP.'' *AI Expert*, Nov., 18–25.

Kosko, B. (1992). *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence.* Prentice-Hall, Englewood Cliffs, N.J.

Kwan, H. K., and Cai, Y. (1994). ''A fuzzy neural network and its application to pattern recognition.'' *IEEE Trans. on Fuzzy Systems*, 2(3), 185–191.

Machada, R. J., and Rocha, A. F. (1992). ''A hybrid architecture for fuzzy connectionist expert systems.'' *Hybrid architecture for intelligent systems*, A. Kandel and G. Langholz, eds., CRC, Boca Raton, Fla.

Rajasekaran, S., and Vijayalakshmi Pai, G. A. (1997). ''Application of simplified fuzzy ARTMAP to structural engineering problems.'' *Proc., All India Seminar on Application of Neural Networks in Sci., Engrg. and Mgmt.*, Bhubaneswar.

Rajasekaran, S., Vijayalakshmi Pai, G. A., and Jesmon, P. G. (1997). ''Simplified fuzzy ARTMAP for determination of deflection in slabs of different geometry.'' *Proc., Nat. Conf. on Neural Networks and Fuzzy Sys.*, Chennai, 107–116.

Schalkoff, R. (1989). *Digital image processing and computer vision.* Wiley, New York.

Takagi, H., Konda, T., and Kojima, Y. (1990). ''Neural network design on approximate reasoning and its application to the pattern recognition.'' *Proc., Int. Conf. of Fuzzy Logic and Neural Networks*, 671–674.

Watada, J., Fu, K. S., and Yao, J. T. P. (1984). ''Linguistic assessment of structural damage.'' *Rep. CE-STR-84-30*, Purdue University, West Lafayette, Ind.

Whang, B. (1997). ''Elasto-plastic orthotropic plates and shells.'' *Proc., Symp. on Application of Finite Element Methods in Civ. Engrg.*, ASCE, New York, 481–515.

Yamakawa, T., and Tomoda, S. (1989). ''A fuzzy neuron and its application to pattern recognition.'' *Proc., 3rd Int. Fuzzy Sys. Assn. Congr.*, 30–38.

APPENDIX II. NOTATION

The following symbols are used in this paper:

$f(x, y)$ = image function;
m_{pq} = moments;
$T_j(I)$ = activation function of jth node for input I;
$\mathbf{W}_j$ = top down weights associated with jth output node;
η_{pq} = normalized central moments;
μ_{pq} = central moments;
ρ = vigilance parameter; and
ϕ_i = RST invariant moment function.

Downloaded from ascelibrary.org by UFPE - Universidade Federal de Pernambuco on 10/02/21. Copyright ASCE. For personal use only; all rights reserved.