SCENE CLASSIFICATION BY FUZZY LOCAL MOMENTS

H. D. Cheng and Rutvik Desai
Department of Computer Science
Utah State University
Logan, UT 84322-4205
cheng@hengda.cs.usu.edu
Tel: (801) 797-2054
Fax: (801) 797-3265

Abstract

Identification of images irrespective of their location, size and orientation is one of the important tasks in pattern analysis. Use of global moment features has been one of the most popular techniques for this purpose. We present a simple and effective method for gray-level image representation and identification which utilizes fuzzy radial moments of image segments (local moments) as features as opposed to global features. A multi-layer perceptron neural network is employed for classification. Fuzzy entropy measure is applied to optimize the parameters of the membership function. The technique does not require translation, scaling or rotation of the image. Furthermore, it is suitable for parallel implementation which is an advantage for real-time applications. The classification capability and robustness of the technique is demonstrated by experiments on scaled, rotated and noisy gray-level images of uppercase and lowercase characters and digits of English alphabet, as well as the images of a set of tools.

I. INTRODUCTION

The issue of pattern/object classification and recognition invariant to translation, rotation and scaling is very important for pattern recognition, image processing and computer vision [18]. Various shape description techniques have been developed to extract image features which represent the image accurately and are robust to noise and changes in position, size or orientation of the image. One of the most commonly used techniques is moment invariant which has many applications in the areas of pattern recognition and image processing. Hu introduced moment invariants in 1962, based on algebraic invariants [1]. If f(x, y) is a piecewise continuous therefore bounded, and has nonzero values only in the finite part of the xy plane, then moments of all orders exist and satisfy the uniqueness theorem: the moment sequence $\{m_{pq}\}$ is uniquely determined by f(x,y); and conversely, f(x,y) is uniquely determined by $\{m_{pq}\}$. This theorem is the foundation of applying moments for classification/recognition. [2] used a set of moment invariants (features) to identify six different types of aircrafts. The experimental results indicated the method had significantly lower error rate than human observers. [3] developed a character recognition system based on moments and was tested on six different machine-printed fonts. The recognition rate 98.5% to 99.7% was achieved. [19] discussed two-dimensional image moments with respect to Zernike polynominals, and the application examples of Zernike moments were also given. [20] used complex moments to derive moment invariants and to provide an analytic characterization of moment invariants as features for pattern recognition and image processing. Problems about information loss, suppression and redundancy in moment invariants were also investigated. [7] evaluated a number of moments (regular moments, Legendre moments, Zernike moments, pseudo-Zernike moments, rotational moments and complex moments) and discussed some essential issues, such as, representation ability, noise tolerance, information redundancy, etc. The authors concluded that high order moments were more vulnerable to noise and the orthogonal moments (Legendre, Zernike and pseudo-Zernike) were better than other types of moments in terms of information redundancy. In terms of overall performance, Zernike and pseudo-Zernike moments are superior to others. [21] described the derivation of moment equations and a method of moment measurement. A threshold sequence and decision rule were developed and implemented in the matching of radar to optical images. A comprehensive study of the effectiveness of different moment invariants in pattern recognition using two sets of data (handwritten numerals and aircrafts) was presented in [22]. It was showed that the new method of deriving Zernike moment invariants along with the new normalization scheme obtained the best overall performance even when the data were degraded by additive noise. [23] proposed a new corner detection method based on the principle of preserving gray and mass moments. The location, orientation and angle of the corners in a digital image can be determined at a good accuracy level. [5] studied a novel global description technique based on lower order silhouette moments. [6] described radial and angular moment invariants and expressed Hu's invariants in terms of these radial and angular moments. [24] discussed a new normalization technique for moment-based global image features and used them for identifying three-dimensional objects from a silhouette image obtained at an arbitrary range and viewpoint. [25] discussed the overall problem of shape reconstruction, normalization and recognition of 3D objects by means of 3D moments. The excellent classification rate was achieved by using only a small number of normalized moments.

Regular or geometric moments have the form of projection of $f(x,y)$ onto the monomial $x^p y^q$, where x and y are the Cartesian coordinates. A regular moment m_{pq} of order $(p+q)$ of continuous image $f(x,y)$ is defined as

$$\mathrm{m_{pq}} = \iint_{\mathrm{A}} x^p y^q f(x,y) dx dy \tag{1}$$

where A is the area of the image.

If $g(r,\theta)$ represents the image function in polar coordinates, i.e., $f(x,y) \equiv g(r,\theta)$, then the *radial* and *angular* moments are defined as follows [6]:

$$\Psi_r(k, g) = \int_r r^k g(r, \theta) dr \quad (2.a)$$

where k is the order of the moment.

$$\Psi_\theta(p, q, g) = \int_\theta \cos^p(\theta) \sin^q(\theta) g(r, \theta) d\theta \quad (2.b)$$

$$\Psi(k, p, q, g) = \int_r \int_\theta r^k g(r, \theta) \cos^p(\theta) \sin^q(\theta) d\theta\, dr \quad (2.c)$$

For digital images, Eq. (2.c) can be written as

$$\Psi(k, p, q, g) = \sum_r \sum_\theta r^k g(r, \theta) \cos^p(\theta) \sin^q(\theta) \quad (2)$$

where k is the order of the radial moments and $(p + q)$ is the order of the angular moments.

Since the fuzzy sets theory was first introduced by Lotfi Zadeh in 1965 [29], it has found many applications in pattern recognition, control system, medicine, social science, etc. [30-32]. In this work we present a simple technique using the first order fuzzy radial moments of the image segments of image as the features. The membership grade of a pixel in an image is determined by the gray level of the pixel, where the standard S-function is used as the membership function. To obtain the optimal parameters for the S-function, a fuzzy entropy measure based on Shannon's function is applied. The classification capability of the technique is tested using a data set consisting of scaled, rotated and noisy gray-level images of uppercase and lowercase characters and digits of English alphabet, as well as scaled and rotated images of a set of tools. A multi-layer perceptron neural network with back-propagation training is used for classification.

II. FUZZY LOGIC, FUZZY MOMENTS AND ENTROPY

Fuzzy logic becomes more and more popular in the field of image processing and pattern recognition recently. The reason is that many image properties, such as boundaries, brightness, etc., are fuzzy in nature. In many real-world situations, we need to classify scenes in which the separation between an object in the scene and the background is not perfect. This can be due to the lighting condition or due to the inherent nature of the scene. The scene may contain various levels of intensities, some of which may be common between the object and the background. It is difficult to use feature extraction methods like moments for such scenes, because it requires one to determine if a given pixel belongs to the object to be classified or not. The concept of fuzzy logic can provide a solution to this problem by allowing pixels to belong to an object *to a certain degree*. We introduce fuzzy logic and propose fuzzy versions of moments described in the previous section to handle classification of gray-level images.

Fuzzy Logic: A fuzzy set A with its finite number of supports x_1, x_2,..., x_n in the universe of discourse U is defined as

$$A = \{\mu_A(x_i)/x_i\}$$

where the membership function $\mu_A(x_i)$ having positive value in the interval [0, 1] denotes the degree to which the element x_i belongs to A. An image X of size M $\times$ N with L gray levels ranging from L_{min} to L_{max} can be defined as an array of fuzzy singletons [8], each with a membership function value denoting its degree of brightness relative to some gray level l ($l = L_{min}, ..., L_{max}$). Thus,

$$X = \{\mu_X(x_{mn}), \quad m = 1, 2, ..., M; \quad n = 1, 2, ..., N\}$$

where x_{mn} is the gray level of the (m, n)th pixel.

Fuzzy Moments: The moments defined by Eq. (3) can be fuzzified as:

$$\Psi_X(k, p, q, g) = \sum_r \sum_\theta r^k \mu_X(g(r, \theta)) \cos^p(\theta) \sin^q(\theta) \tag{3}$$

The S-function [8] is applied as the membership function which is defined as:

$$\mu_X(x) = S(x, a, b, c)$$

$$= \begin{cases} 0 & x \leq a \\ 2 \times [(x-a)/(c-a)]^2 & a \leq x \leq b \\ 1 - 2 \times [(x-c)/(c-a)]^2 & b \leq x \leq c \\ 1 & x \geq c \end{cases} \tag{4}$$

In Fig. 1, a typical S-function is depicted. Here, $b = (a+c)/2$ is the crossover point. The portion of image X in the interval [a, c] is a fuzzy region, whereas the regions in [L_{min}, a] and [c, L_{max}] are crisp or non-fuzzy.

The values of parameters a, b and c can seriously affect the quality of the solution, because these parameters determine the degree of membership of a given pixel in the image. For different images, different values of a, b and c should be determined, depending on the nature of the image. We propose an automated way to determine the optimal values for these parameters for a given image using minimization of entropy, discussed below.

Entropy and Fuzzy Sets: When uncertainty is involved in the system under consideration, the concept of entropy can play an important role in the formulation of a model of the system. In probabilistic systems, entropy under the measure of probability is used. On the other hand, when the system is characterized by vagueness or fuzziness as opposed to randomness, e.g., lack of a crisp boundary between regions of an image, entropy should be based on fuzziness. The entropy of a fuzzy set is an indication of the ambiguity in the set. For a fuzzy set X, the entropy [9, 11, 12] can be measured by:

$$H(X) = \frac{1}{MNln2} \sum_{m=1}^{M} \sum_{n=1}^{N} S(\mu_X(x_{mn})) \tag{5}$$

where $S(\cdot)$ is the Shannon's function, given by

$$S(\mu_X(x_{mn})) = -\mu_X(x_{mn}) ln(\mu_X(x_{mn})) - (1-\mu_X(x_{mn})) ln(1-\mu_X(x_{mn})) \quad (6)$$

$$\mathrm{m} = 1, 2, ..., \mathrm{M}; \mathrm{n} = 1, 2, ..., \mathrm{N}$$

If we use the gray level x instead of the intensity of (m, n)th pixel x_{mn}, then the fuzzy entropy is computed by

$$H(X) = \frac{1}{MNln2} \sum_{x=L_{min}}^{L_{max}} S(\mu_X(x, a, b, c)) h(x) \quad (7)$$

where $h(x)$ is a function indicating the number of pixels with a given intensity x of the image, and $x \subseteq \{L_{min}...L_{max}\}$.

Low entropy of an image indicates less ambiguity and better separation between the various regions of the image. In a classification task where images consist of object(s) and the background, a low entropy means better separation between the object(s) and the background, which facilitates a more accurate feature extraction. In this work, we select the location of the fuzzy region of the S-function $((c-a) = 2(b-a) = 2\Delta)$ so as to minimize the entropy of the image. For a selected width of the fuzzy region, entropy is computed for all possible values of the parameter a and the value of a which produces the minimum entropy is selected. If entropy of a fuzzy set representing an image X is denoted by H(X, a, b, c) where a, b and c are the parameters of the S-function, and $a < b < c$, then the optimum values a^{opt}, b^{opt} and c^{opt} of the parameters a, b and c respectively are obtained from

$$H_X^{min}(X, a^{opt}, b^{opt}, c^{opt}) = min\{H(X, a, b, c) | a = 0, 1, ..., (\mathrm{L_{max}} - 2\Delta)\} \quad (8)$$

This approach has the advantage that while respecting the inherent fuzziness of the image, it tries to optimize the membership function so as to achieve minimum ambiguity and better feature extraction. How to select the fuzzy region and determine the fuzzy membership function can be found in [26 - 28].

III. IMAGE DESCRIPTION SCHEME

First, the centroid of the image (x_0, y_0) is found, using the zero-th order and the first order moments as shown below:

$$x_0 = \frac{m_{10}}{m_{00}}, \quad y_0 = \frac{m_{01}}{m_{00}} \tag{9}$$

where $\mathrm{m_{pq}}$ is obtained from Eq. (1). Next, the image is divided into n sectors formed by drawing n radial lines from the centroid at an angle θ to each other such that $n\theta = 360^0$. Furthermore, m concentric circles are drawn with the centroid of the image as center such that each "band" or "ring" between any two consecutive circles has equal area [5]. This is accomplished by computing the radius $\mathrm{r_j}$ of the jth circle as follows:

$$\mathrm{r_j} = \sqrt{\frac{\mathrm{j}}{\mathrm{m}}}\,\mathrm{r_m} \tag{10}$$

where $\mathrm{r_m}$ is the distance between (x_0, y_0) and the furthest pixel of the image from (x_0, y_0). Thus, the image is divided into $n \times m$ segments. Fig. 2 shows this scheme for n=12 and m=3. Fig. 3(a) and 3(b) show images of characters 'G' and 'H' after segmenting them according to this scheme. The first order radial moment of the segment contained between angles α and $\alpha + \theta$ and band between circles of radius $\mathrm{r_j}$ and $\mathrm{r_{j+1}}$ is given by

$$\Psi_X^i = \Psi_X^{(r_j,\alpha)}(1,g) = \sum_{r_j}^{r_{j+1}} \sum_{\alpha}^{\alpha+\theta} r\, \mu_X(g(r,\theta)) \tag{11}$$

$$\text{where } \mathrm{i} = \mathrm{j\,n} + \tfrac{\alpha}{\theta} + 1$$
$$\theta = \frac{360^0}{\mathrm{n}}$$

Eq. (12) is obtained from substituting $k = 1$ (first order radial) and $p = q = 0$ (no angular moments) in Eq. (4). Moments of each segment are computed using Eq. (12) to form an $\mathrm{n} \times \mathrm{m}$ feature vector which describes the image. Thus, the image is represented by the feature vector

$$\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_{\mathrm{n}\times\mathrm{m}}]$$
$$\mathrm{x}_\mathrm{i} = \Psi^\mathrm{i}_\mathrm{X}$$

Note that all the sectors and bands have equal area, so all of $n \times m$ segments cover an equal area of the image. The reason for having segments of equal area is to ensure that all portions of the image are emphasized equally in extracting features. This enables us to apply this method "blindly" to a image classification problem, without having to tweak any parameters. This approach can be tailored easily to achieve better performance for a particular problem, by changing the size of bands so as to finely divide areas of the image which contain details important for classification.

The obtained feature vector can be transformed as described below to achieve translation, scale and rotation invariance.

A. Translation invariance:

In the above scheme, translation invariance is achieved by moving the origin to the centroid of the image. This is done by transforming the image f(x,y) into $\mathrm{f}(\mathrm{x}+\mathrm{x}_0,\mathrm{y}+\mathrm{y}_0)$ where $(\mathrm{x}_0,\mathrm{y}_0)$ is the centroid of the image and determined by Eq. (10).

B. Scale invariance:

Scale invariance can be accomplished by normalizing the feature values computed by Eq. (12). X is transformed to X^s by

$$\mathrm{X}^\mathrm{s} = \frac{\mathrm{X}}{\mathrm{X}^{\mathrm{max}}} \tag{12}$$

$$\mathrm{X}^{\mathrm{max}} = \max(\mathrm{X})$$

C. Rotation invariance:

Consider the feature vector X of the form

$$\begin{aligned} \mathrm{X} \;=\; & \{[x_1, x_2, ..., x_{M_1-1}, x_{M_1}, x_{M_1+1}, ..., x_{1\times n}] \\ & [x_{n+1}, x_{n+2}, ..., x_{M_2-1}, x_{M_2}, x_{M_2+1}, ..., x_{2\times n}] \\ & \bullet\bullet\bullet \\ & [x_m, x_{m+1}, ..., x_{M_m-1}, x_{M_m}, x_{M_m+1}, ..., x_{m\times n}]\} \end{aligned} \tag{13}$$

where

$$\begin{gathered} x_{M_i} = max\{x_j\} \\ (\mathrm{i}-1)\times \mathrm{n} < \mathrm{j} < \mathrm{r}\times \mathrm{n} \\ 1 \leq \mathrm{i} \leq \mathrm{m} \end{gathered}$$

In this representation, each subvector represents the feature values corresponding to a particular band. The orientation invariance can be achieved by transforming X into X^{R} as following:

$$\begin{aligned} \mathrm{X}^{\mathrm{R}} \;=\; & \{[x_{M_1}, x_{M_1+1}, ...x_{1\times n}, x_1, x_2, ..., x_{M_1-1}] \\ & [x_{M_2}, x_{M_2+1}, ..., x_{2\times n}, x_{n+1}, x_{n+2}, ..., x_{M_2-1}] \\ & \bullet\bullet\bullet \\ & [x_{M_m}, x_{M_m+1}, ..., x_{m\times n}, x_m, x_{m+1}, ..., x_{M_m-1}]\} \end{aligned} \tag{14}$$

In other words, each feature subvector is shifted in such a way that the maximum valued feature becomes the first element of the subvector, and the relative order of the features remains the same for that subvector. In case of more than one maximum values in a particular band, the value with the next highest value is selected as maximum. Clearly, X^{R} is independent of the orientation of the image, since only the order of the features changes with orientation of the image.

Notice that a sector of θ degrees gives perfect rotation invariance only if the image is rotated in steps of θ degrees. The error due to rotation will be maximum when the image is rotated in steps of $\theta/2$ degrees. However, if θ is sufficiently small, then the error can be ignored at the cost of higher computational time.

The computation of the moments of a segment is independent of that of other segments. Thus the computational process lends itself naturally to parallel implementation. The moments of each band of the image can be computed in parallel and the procedure for obtaining scale and rotation invariance can also be performed in parallel, therefore, the computational time can be significantly reduced. This parallel nature of the algorithm can be utilized effectively in real-time applications.

IV. CLASSIFICATION

The aim of pattern classification [4] is to assign input patterns to one of a finite number, M, of classes. Biologically inspired adaptive non-parametric artificial neural-network classifiers (or simply neural-nets) have gained popularity for classification tasks. They posses the ability to learn and build unique structure specific to a particular problem. The ability to select combinations of features pertinent to the problem, fault tolerance or graceful degradation with noise, potential for real-time parallel implementation using custom VLSI circuits are the major characteristics of neural-nets which give them an edge over the conventional statistical classifiers. Amongst several neural-net paradigms available, we use the multi-layer perceptron, discussed below.

The multi-layer perceptron: This type of network [14], illustrated in Fig. 4, has an input layer, an output layer and at least one hidden layer. Each layer comprises several processing elements (PEs/neurons). Each layer is usually fully connected to the succeeding layer. The number of units in the input layer equal the number of features plus one as the bias. The number of output neurons depends on the number of classes into which the input data are to be classified. In the *local* or *distributed* encoding of outputs [17], where each output neuron is assigned to single class, and the number of outputs equals the number of classes. As shown in Fig. 4, inputs to any processing element are multiplied by the connection weights and summed to produce a network value, net_j, as given in Eq. (16).

$$net_j = \sum w_{ji} \times O_i \tag{15}$$

Here, w_{ij} is the weight between neurons U_i and U_j, and O_i is the output of neuron i. This sum of products is transferred to the neuron output via a semi-linear a sigmoid function,

$$O_j = \frac{1}{(1 + exp(-net_j))} \tag{16}$$

The network operates in two distinct phases: training and recall [14,16]. During training, each member of the training set is presented to the network inputs. The output produced is compared with a target vector for that input and an error vector is computed. This error is then back-propagated through the network layers and the connection weights are adjusted according to the learning rule. This process continues until the r.m.s. error of the output layer for the whole training set falls below a prespecified threshold or the network stops improving. In order to determine if the neural network has stopped improving, a verification phase is often introduced. A verification set is made from the training samples. Network performance is checked periodically on the verification set, and training stops if the performance does not improve for a certain number of training cycles. This method, known as cross-validation, can prevent under-training or over-training of the neural network [33].

At the start of the training the connection weights are assigned uniformly random values between a high and low predefined limit. As traing proceeds, the weights between any two neurons U_i and Uj are adjusted using the local neuron Uj. For a given training vector, if **d** represents the desired output vector and **O** represents the actual output vector produced, then a measure of error in achieving that desired output is given by

$$E = 1/2 \sum_k (\mathbf{d_k} - \mathbf{O_k})^2 \tag{17}$$

where the subscript k indexes the components of $\mathbf{d}$ and $\mathbf{O}$. The *local error*, e_k, at each neuron in the output layer is then used to compute the required change in the value, Δw_{kj}, for each weight connecting the neurons of the output layer

$$\Delta w_{kj} = \mu e_k \mathbf{O_j} \tag{18}$$

where μ is a constant learning coefficient. By back-propagating errors through the network a similar process can be employed to adjust weights between other layers. The algorithm outlined above is known as the *error back-propagation* algorithm. One of its characteristics is long training period. Several improvements in the standard back-propagation have been suggested to reduce the training time. We have used a variation popularly known as *QuickProp*, introduced by S. Fahlman [15].

V. EXPERIMENTS AND RESULTS

Experiment 1.

The effectiveness of the method is tested by utilizing the dataset of images of 62 characters (26 uppercase, 26 lowercase and digits 0-9). The gray level for the characters has the range from 0 to 91, while for the background has the range from 164 to 255. We apply the image transformation algorithms [34] to generate new images. Each image is rotated at 7 random angles to generate 7+1 images per character in a 64×64 resolution. Each resulted image is scaled by a factor of 0.8, 0.9, 1.0, 1.1 and 1.2. For each such image, a noise of 0%, 5%, 10%, 15%, 20% and 25% is added. Totally, $8 \times 5 \times 6 = 240$ different samples are produced for each character, and the dataset has $240 \times 62 = 14{,}880$ samples.

For segmenting an image, one has to decide the number of sectors and bands. In order to achieve rotation invariance, the sectors have to be small. But small sectors imply large dimensions of the feature vectors, and high computational time. Experimentally, we found that $\theta = 10^o$ is a good choice.

The gray-level of each pixel of noise is chosen randomly between 150 and 200. Images of characters '5', 'a' and 'T' at different orientations with scaling factors of 1.0, 0.8, 0.9, 1.1 and 1.2, and with noise of 0%, 5%, 10%, 15% and 25% are shown in Fig. 5. Radial moments of each segment are computed using Eq. (12). Transformations are performed as discussed in Section III to make this vector invariant to translation, scaling and rotation. Normalization for scale is performed for each band, which is advantageous if moments of each band are calculated in parallel. The width of the fuzzy region of the S-function is selected as 225. For each image, the membership function is optimized by Eq. (9). Scaling is done prior to rotation. Moments of various sectors of the images of character 'a' at scales of 1.0 and 0.9 and rotated by 0^o and 62^o, before and after performing transformations for scale and rotation invariance are shown in form of graphs in Fig. 6a) through 6d), respectively. Numbering of segments is similar to that shown in Fig. 2.

The neural network has an input layer with $36 \times$ n + 1 (bias) nodes (n is the number of bands used); a hidden layer with 62 nodes (for 1 band and 2 band cases), 80 nodes (for 3 band case) and 100 nodes (for 4 band case); and an output layer with 62 nodes. To train the neural-net classifier, a training set is prepared from the dataset generated as above. For each class, 3 noisy (0%, 10% and 20%), 3 scaled (0.8, 1.0 and 1.2) and 2 rotated (0^o and another angle picked randomly) samples are chosen to form a training set of $3\times3\times2$ = 18 samples per class, and there are $18 \times 62 = 1{,}116$ samples in total. Thus, less than 10% of the dataset is used for training. A verification set of the same size is prepared, for using the cross-validation method to prevent over- or under-training of the network. The rest of the samples are used as the testing set. A single hidden-layer network is used with local encoding of outputs (one node is assigned to each character, which is coded by 1 in the output vector; rest of the nodes are assigned 0). Training is stopped when there is no improvement in recognition rate on the verification set for a certain number of epochs. Neural networks in this work are implemented using the NeuralWorks Professional II/Plus package [16]. For interpreting the output of the network, we apply the winner-take-all principle, i.e., we consider the neuron with the maximum output value to be 1, and rest as 0. The network is considered to have identified the character corresponding to the node with output 1.

The results obtained using various number of bands are shown in Table I. A high accuracy is achieved even for highly noisy images, in spite of the small size of the training set. Using 3 bands gives the best performance. Lowing number of bands means that there are not enough feature elements for classification. Increasing number of bands means that each feature element is very small and the effect of the noise could surpass the one of the image. However, experimental results show that the performance does not change drastically with the number of bands which implies that the choice of the number of bands is not very critical.

It is important to notice that many of the errors are caused by the nature of the problem, rather than by an inadequacy of the technique. Uppercase versions of characters 'c', 'o', 'p', 's', 'v', 'w', 'x', 'y' and 'z' are very similar to their lowercase counterparts when scale-invariance is taken into account. Similarly, characters like 'p' and 'd' or 'b' and 'q' or 'u' and 'n' are very difficult to distinguish, even for a human observer, when they are rotated 180^o.

Experiment 2.

The problem of classification of images of seven tools shown in Fig. 7 is used for testing the proposed method. The gray level ranges for foreground, background and noise are the same as the previous experiment. The image size is 100×100. The experiment is carried out in a manner similar to that of the previous experiment. Dataset consisting of 40 images per class is prepared by scaling images by factors of 0.8, 0.9, 1.0, 1.1 and 1.2 and by rotating them at 7 random angles. Then images are segmented by 36 sectors and 2 bands. The neural network has an input layer with $36 \times 2 + 1$ (bias) nodes, a hidden layer with 50 nodes and an output layer with 7 nodes. We experimented with several numbers of input nodes and found that 50 was a good choice to handle the high-dimensional input vectors. The training and verification sets consist of one unrotated and unscaled image of the tool, and one other image picked randomly. The rest of the samples are used for the testing set. A neural network with a single hidden layer is trained with cross-validation. A 100% recognition rate has been achieved on the testing set.

VI. DISCUSSIONS AND CONCLUSIONS

A simple, effective moment-based technique for gray-level image description and recognition has been proposed. The technique has been tested on two different problems and found to be robust to scale and orientation changes as well as noise. A method for automatic determination of the optimal membership function using minimization of image entropy has also been proposed. Like all methods relying on the center of gravity of the image, this technique is sensitive to certain type of noises, e.g., cases where some portions of the image are completely missing. It would probably not be advisable to use this method in such cases. However, it is tolerant to a high degree of salt-and-pepper type of noise and can be used in a variety of situations where invariance is required. It does not require boundary detection or contour following and thus is insensitive to discontinuous boundaries. The feature extraction method is task-independent, therefore, it can be applied to a variety of pattern classification problems. There is no translation, scaling or rotation of the image involved, which reduces the computation time significantly. Furthermore, the algorithm is suitable for parallel implementation.

This method utilizes the first order radial moments which are simple to compute. However, it has been shown in studies comparing several types of moments (regular, complex, Legendre, Zernike, pseudo-Zernike, etc.) that Zernike moments are superior to other types of moments in terms of image representation capability, information redundancy and noise sensitivity [7]. They can be used instead of radial moments at an additional computational cost. The high dimensionality of the feature vector could be a drawback of this method, since it usually increases the training time of the neural network. This problem is alleviated to some degree by the small size of training set required. Radial basis function classifiers have been shown to significantly reduce the training time, while typically providing error rates similar to those provided by back-propagation classifiers [13]. They can be used instead of back-propagation classifiers to eliminate the problem. Local output coding method has been used for encoding neural-net outputs in this work. However, another encoding scheme known as the *error-correcting* coding can be used which has been shown to improve performance of back-propagation classifiers[17].

REFERENCES

1. M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory,* Vol. IT-8, pp. 179-187, Feb. 1962.

2. S. A. Dudani, K. J. Breeding and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, Vol. C-26, No. 1, pp. 39-45, Jan. 1983.

3. G. L. Cash and M. Hatamian, "Optical character recognition by the method of moments", *Computer Vision, Graphics, and Image Processing*, **39**, 291-310, 1987.

4. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Ed. Academic Press, 1990.

5. H. K. Sardana, M. F. Daemi, A. Sanderes and M. K. Ibrahim, "A novel moment-based shape description and recognition technique," *IEE The 4th Int. Conf. on Image Processing and its Applications*, pp. 147-150, Apr. 1992.

6. S. S. Reddi, "Radial and angular moment invariants for image identification," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-3, pp. 240-242, Mar. 1981.

7. C. H. Teh and R. T. Chin, "On image analysis by methods of moments," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. PAMI-10, No. 4, pp. 496-513, July 1988.

8. S. K. Pal and D.K. Dutta, *Fuzzy Mathematical Approach to Pattern Recognition,* John Wile & Sons, 1986.

9. N. R. Pal and S. K. Pal, "Entropy: A new definition and its applications," *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-21, No. 5, pp. 1260-1270, Sept./Oct. 1991.

10. L. A. Zadeh, "Calculus of fuzzy restrictions," In: L. A. Zadeh et. al., Eds, *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, Academic Press, 1975.

11. S. K. Pal and A. Rosenfeld, "Image enhancement and thresholding by optimization of fuzzy compactness," *Pattern Recognition Letters*, Vol. 7, pp. 77-86, Feb. 1988.

12. Xueqin Li, Zhiwei Zhao, H. D. Cheng, C. M. Huang and R. W. Harris, "A fuzzy logic approach to image segmentation," *The 12th International Conference on Pattern Recognition*, Jerusalem, Israel, Oct. 9-13, 1994.

13. R. Lippmann, "Pattern Classification Using Neural Networks," *IEEE Comm. Magazine*, Vol. 27, pp. 47-62, Nov. 1989.

14. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986.

15. S. E. Fahlman, "Faster-Learning Variations on Back-Propagation: An Empirical Study," *Proc. of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton and T. Sejnowski, eds., pp. 38-51, San Mateo, CA: Morgan Kauffman, 1989.

16. *NeuralWorks Professional II/Plus Users Manual.*

17. T. Dietterich and G. Bakiri, "Error-Correcting Output Codes: A General Method for Improving Multiclass Inductive Learning Programs," *Proc. of the Ninth National Conf. on Artificial Intelligence*, Vol. 2, pp. 572-577, 1991.

18. J. L. Mundy and A. Zisserman, *Geometric Invariance in Computer Vision*, The MIT Press, 1992.

19. M. R. Teague, "Image analysis via the general theory of moments", *J. Opt. Soc. Am.*, Vol. 70, No. 8, 920-930, 1980.

20. Y. S. Abu-Mostafa and D. Psaltis, "Recognitive aspects of moment invariants", *IEEE Trans. Pattern Anal. Machine Intell*, Vol. 6, No. 6, 698-706, Nov. 1984.

21. R. Y. Wong and E. L. Hall, "Scene matching with invariant moments", *Computer Graphics and Image Processing*, **8**, 16-24, 1978.

22. S. O. Belkasim, M. Shridhar and M. Ahmadi, "Pattern recognition with moment invariants: a comparative study and new results", *Pattern Recognition*, Vol. 24, No. 12, 1117-1138, 1991.

23. S. T. Liu and W. H. Tsai, "Moment-preserving corner detection", *Pattern Recognition*, Vol. 23, No. 5, 441-460, 1990.

24. A. P. Reeves, R. J. Prokop, S. E. Andrews and F. P. Kuhl, "Three-dimensional shape analysis using moments and Fourier descriptors", *IEEE Trans. Pattern Anal. Machine Intell*, Vol. 10, No. 6, 937-943, Nov. 1988.

25. J. M. Galvez and M. Canton, "Normalization and shape recognition of three-dimensional objects by 3D moments", *Pattern Recognition*, Vol. 26, No. 5, 667-681, 1993.

26. X. Li, Z. Zhao and H. D. Cheng, "Fuzzy entropy threshold approach to breast cancer detection", *Information Sciences, Applications, An International Journal*, Vol. 4, No. 1, 49-56, 1995.

27. H. D. Cheng and J. R. Chen, "Automatically determine the membership function based on the maximum entropy principle", *the Second Annual Joint Conference on Information Sciences (JCIS)*, Wrightsvill Beach, NC, Sept. 28 - Oct. 1, 1995.

28. H. D. Cheng and Y. M. Lui, "An iterative approach for automatic bandwidth selection of fuzzy membership functions" *the Second Annual Joint Conference on Information Sciences (JCIS)*, Wrightsvill Beach, NC, Sept. 28 - Oct. 1, 1995.

29. L. A. Zadeh, "Fuzzy sets", *Information and Control*, Vol. 8, No. 3, 338-353, 1965.

30. G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice-Hall, 1995.

31. J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition*, IEEE Press, 1992.

32. T. Terano, K. Asai and M. Sugeno, *Fuzzy Systems Theory and Its Applications*, Academic Press, 1992.

33. S. Amari, N. Murata, K. R. Muller, M. Finke and H. Yang, "Asymptotic statistical theory of overtraining and cross-validation", METR 95-06, Dept. of Mathematical Engineering and Information, Physics, Univ. of Tokyo, Japan, Aug. 1995.

34. H. D. Cheng and S. Nho, "Transformation of gray level and color images", accepted by *Information Sciences, An International Journal*, 1995.