SciencePG
Science Publishing Group

Soft Computing Techniques for Various Image Processing Applications: A Survey

Rahul Kher[1], Heena Kher[2]

[1]Department of Electronics & Communication Engg, G H Patel College of Engineering & Technology, Vallabh Vidyanagar, India

[2]Department of Electronics & Communication Engg, A D Patel Institute of Technology, New Vallabh Vidyanagar, India

Email address:
rahul2777@gmail.com (R. Kher), heena.adit@gmail.com (H. Kher)

Abstract: Soft computing techniques have found numerous applications in various domains of image processing and computer vision. This paper represents a survey on various soft computing methods'- fuzzy logic, neural network, neuro-fuzzy systems, genetic algorithm, evolutionary computing, support vector machine etc. - applications in various image processing areas. There are numerous applications of SC ranging from industrial automation to agriculture and from medical imaging to aerospace engineering, but this paper deals with the relevance and feasibility of soft computing tools in the area of image processing, analysis and recognition. The techniques of image processing stem from two principal applications namely, improvement of pictorial information for human interpretation and processing of scene data for automatic machine perception. The different tasks involved in the process include enhancement, filtering, noise reduction, segmentation, contour extraction, skeleton extraction etc. Their ultimate aim is to make understanding, recognition and interpretation of the images from the processed information available from the image pattern. There are many hybridized approaches like neuro-fuzzy system (NFS), fuzzy-neural network (FNN), genetic-fuzzy systems, neuro-genetic systems, neuro-fuzzy-genetic system exist for various image processing applications. Tools like genetic algorithms (GAs), simulated annealing (SA), and tabu search (TS) etc. have been incorporated with soft computing tools for applications involving optimization.

Keywords: Soft Computing, Image Processing, Fuzzy Logic, Neural Networks, Medical Images

1. Introduction

Soft Computing (SC) is an important field that comprises of techniques like fuzzy logic, neural computing, evolutionary computation, machine learning and probabilistic reasoning. Because of their inherent learning and cognitive ability and good tolerance of uncertainty and imprecision, the SC techniques have been applied in varieties of domains. Unlike conventional (hard) computing, the Soft computing is tolerant of imprecision, uncertainty, partial truth, and approximation. Thus, soft computing can be used for decision making keeping human brain as the model. The SC can be thought of comprising the following tools/ algorithms – Fuzzy Logic (FL), Neural Networks (NN), Support Vector Machines (SVM), Evolutionary Computation (EC), and – Machine Learning (ML) and Probabilistic Reasoning (PR). Soft computing techniques, basically learns from previously unseen inputs by using outputs from previous learned inputs to produce outputs. Thus, learning from experimental data is a unique property of soft computing [1].

There are numerous applications of SC ranging from industrial automation to agriculture and from medical imaging to aerospace engineering, but this paper deals with the relevance and feasibility of soft computing tools in the area of image processing, analysis and recognition. The techniques of image processing stem from two principal applications namely, improvement of pictorial information for human interpretation and processing of scene data for automatic machine perception. The different tasks involved in the process include enhancement, filtering, noise reduction, segmentation, contour extraction, skeleton extraction etc. Their ultimate aim is to make understanding, recognition and interpretation of the images from the processed information

available from the image pattern. In an image analysis system, uncertainties can arise at any phase resulting from incomplete or imprecise input information, ambiguity or vagueness in input images, ill-defined and/or overlapping boundaries among the classes [2].

This paper presents a survey on soft computing methods' applications in various fields of image processing and computer vision, for example image filtering/ de-noising, image segmentation, image classification, image compression, character/ text recognition, medical image analysis, object detection, extracting edges & skeletons.

2. Soft Computing for Image Processing

This section describes the usefulness of soft computing in various image processing and computer vision domains followed by a few example applications.

2.1. Usefulness of. Soft Computing in Image Processing Applications

Images of real scenes very frequently contain data which is ambiguous and incomplete. The conventional mathematical approaches offer very little facilities in designing a flexible model that may be, response wise, close to human visual system (HVS). To overcome the limitations, scientists have been searching for new approaches capable of modeling the characteristics close to those mentioned above for solving real world problems efficiently. Soft computing is one such outcome that has emerged in the recent past as a collection of several modes of computation which work synergetically and provide, in one form or another, the capability of flexible information processing. The objective of soft computing is to exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve tractability, robustness, low solution cost and close resemblance with human like decision making. Soft computing techniques have been found extremely useful in applications like filtering images, enhancing images, extracting features images, edge/ boundary detection from images, object/ region identification images, compressing images, image pattern classification, face identification, target & character recognition, motion analysis & estimation [2].

2.2. Image Filtering / Denoising Using Soft Computing

It is very usual that images are corrupted by various types of noise sources. Filtering of noisy images is a very challenging issue as noise has to be removed from the image data without destroying fine details and textures. *Evolutionary neural fuzzy filters* are a class of *nonlinear filters* for image processing. The structure of these filters adopts fuzzy reasoning in order to cancel noise without destroying fine details and textures. The learning method based on the Genetic Algorithms yields very satisfactory results within a few generations. Experimental results have shown that evolutionary neural fuzzy filters are very effective in removing impulse noise from highly corrupted images and significantly outperform conventional

techniques [2].

Figure 1 (a) shows a typical 3x3 mask that may be used to filter an image. Figure 1 (b) represents a basic structure of a neural fuzzy filter for removing a salt and pepper noise from an image. The filter is formed by two symmetrical subnetworks that aim at detecting positive and negative noise pulses, respectively. Squares denote nodes that perform fuzzy set-based operations. Circles denote minimum and maximum operators. There are four layers- Layer 1: fuzzification layer, Layer 2 and 3: hidden layers, Layer 4: output layer. The output layer evaluates a correction term which, added to $x(n)$, yields the resulting pixel value $y(n)$ in the output image. In order to increase the ability to remove noise pulses, the filter is recursively applied to the image data, i.e., the new value $y(n)$ is assigned to $x(n)$ at the end of the processing.
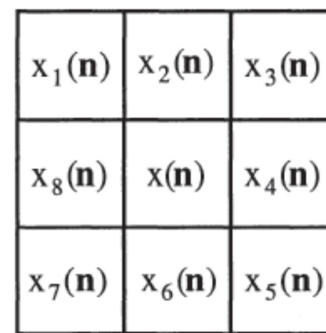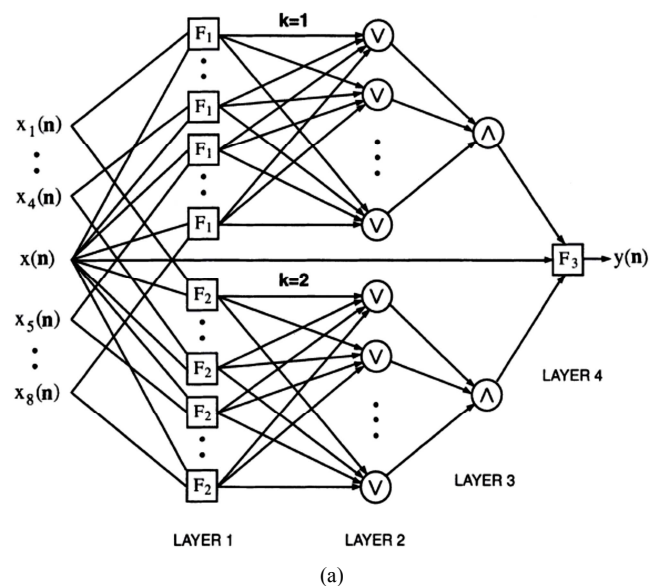

(a)

(b)

Figure 1. *(a) Pixels belonging to a 3x3 neighbourhood (b) Network structure of an evolutionary neural fuzzy filter [2]. © Springer-Verlag.*

When the image data are highly corrupted by noise, a larger neighborhood as shown in Figure 2, should be considered in order to take care of many possible patterns of adjacent noise pulses. The network structure of a neuro-fuzzy filter operating on such a mask will have an upper symmetrical half containing $x_1(n)$ to $x_9(n)$ and lower half containing $x_1'(n)$ to $x_{16}'(n)$.

$\dot{x}_1$ (n)	$\dot{x}_2$ (n)	$\dot{x}_3$ (n)	$\dot{x}_4$ (n)	$\dot{x}_5$ (n)
$\dot{x}_{16}$(n)	x_1 (n)	x_2(n)	x_3 (n)	$\dot{x}_6$ (n)
$\dot{x}_{15}$(n)	x_8(n)	x(n)	x_4 (n)	$\dot{x}_7$ (n)
$\dot{x}_{14}$(n)	x_7 (n)	x_6 (n)	x_5 (n)	$\dot{x}_8$ (n)
$\dot{x}_{13}$(n)	$\dot{x}_{12}$(n)	$\dot{x}_{11}$(n)	$\dot{x}_{10}$(n)	$\dot{x}_9$ (n)

Figure 2. 5x5 mask for filtering a highly corrupted image [2]. © Springer-Verlag.

2.3. Edge Extraction Using Fuzzy Reasoning

Images of real scenes very frequently contain data which is ambiguous and incomplete. In particular, the problem of determining what is and what is not an edge in an image is confounded by the fact that edges are often partially hidden or distorted by various effects such as uneven lighting. Furthermore, images frequently contain data with edge-like characteristics, but a confident classification of such data can best be resolved when high level constraints are imposed on the interpretation of an image. The problem of detecting edges in images can be characterized as a fuzzy reasoning problem. The edge detection problem is divided into three stages: filtering, detection, and tracing. Images are filtered by applying fuzzy reasoning based on local pixel characteristics to control the degree of Gaussian smoothing. Filtered images are then subjected to a simple edge detection algorithm which evaluates the edge fuzzy membership value for each pixel, based on local image characteristics. Finally, pixels having high edge membership are traced and assembled into structures, again using fuzzy reasoning to guide the tracing process. An additional process called *joining* is applied in order to have the continuity in the detected edges. Figure 3 shows the typical edge extraction result obtained by fuzzy reasoning [2].

Algorithms based on Genetic Algorithm and Exhaustive search have also been used for edge extraction.

2.4. Soft Computing for Image Segmentation

Image segmentation methodologies can be viewed as efforts to classify distinct regions in an image by recognition of embedded patterns using various criteria including similarity measures, decision rules, and cluster validity measures among many others. Image segmentation allows mapping of similar regions in a scene leading to recognition of distinct objects by high-level vision systems. Therefore, an efficient technique of clustering is a natural choice for image segmentation. Recognition of similar patterns embedded in image data is the basis of clustering subregions in an image. Efforts to develop algorithms for adaptive and less computationally complex classification of data have led to

implementation of statistical classifiers in artificial neural networks (of both supervised and unsupervised categories). Such neural network architectures are ways to achieve autonomous processing of patterns but are not considered to incorporate intelligent decision processes offered by various models of fuzzy clustering. Integration of fuzzy membership values of samples into neural network processing generates more powerful models for autonomous and intelligent pattern recognition algorithms. Efficient object extraction for image segmentation as well as vector quantization for image coding can be achieved by neuro-fuzzy clustering algorithms [2].
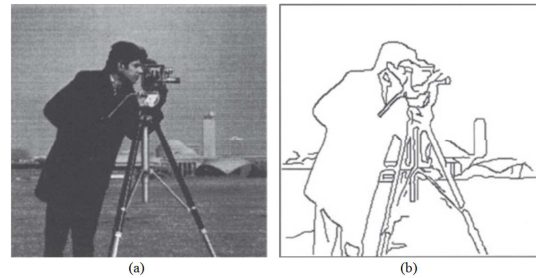

(a)   (b)

Figure 3. (a) original image, (b) after filtering, detection, tracing, and joins [2]. © Springer-Verlag.

In non-fuzzy *hard* clustering, the boundaries of different clusters are such that one pattern is assigned to exactly one cluster. On the contrary, fuzzy clustering provides partitioning resulting from additional information supplied by the cluster membership values indicating different degrees of belongingness [3, 4]. *The fuzzy c-means clustering (FCM)* has proven to be an effective algorithm for segmenting the images.

Neuro-Fuzzy algorithms retain the basic properties and architectures of neural networks and simply *fuzzify* some of their elements. In these classes of networks, a crisp neuron can become fuzzy and the response of the neuron to its activation layer signal can be of a fuzzy relation type rather than a continuous or crisp function type. Examples of this approach can be found where domain knowledge becomes formalized in terms of fuzzy sets and afterward can be applied to enhance the learning algorithms of the neural networks or augment their interpretation capabilities. Since the neural architecture is conserved, changes are made in the weights connecting the lower layer to the upper layer neurons. *The Adaptive Fuzzy Leader Clustering Network (AFLC)* developed by Newton *et al* [5] is a hybrid neuro-fuzzy system that can be used to learn cluster structure embedded in complex data sets, in a self-organizing and stable manner. *The Integrated Adaptive Fuzzy Clustering Network (IAFC)* model developed by Kim and Mitra [6] is another useful algorithm for image segmentation.

Remotely sensed images are normally poorly illuminated, highly dependent on the environmental conditions, and have very low spatial resolution. Most of the times a scene contains too many objects (or regions), and these regions are ill-defined because of both grayness and spatial ambiguities. Moreover, the gray value assigned to a pixel is the average reflectance of different types of ground covers present in the corresponding pixel area. Assigning unique class labels with certainty is thus a problem for remotely sensed images. Fuzzy set theory

provides a way of handling this problem by associating certainty factors with class labels.

The problem of segmenting remotely sensed images has been addressed in [7]-[9]. Laprade [8] presented a split-and-merge technique using F-test and a mean predicate to test the uniformity of regions and applied it to aerial photographs. A two-stage fuzzy c-means algorithm was applied on a Landsat- 4 image with six-bands to demonstrate the feasibility of the methodology for segmentation by Cannon et al. [7]. A method of evaluating the suitability of valleys as threshold has been proposed in [10] and applied to satellite image segmentation. Attempts are also / made to find road like structures [11]-[13] and man-made object identification from remotely sensed images [9]. Fuzzy c-means clustering (FCM) and hard c-means clustering (HCM) have also been used for segmenting remotely sensed images. Figure 4 shows the results of FCM and HCM applied to Calcutta image [2].
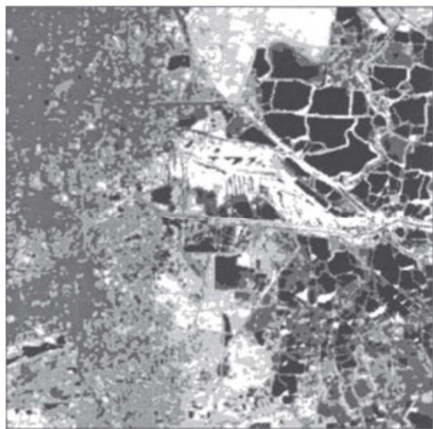
(a)
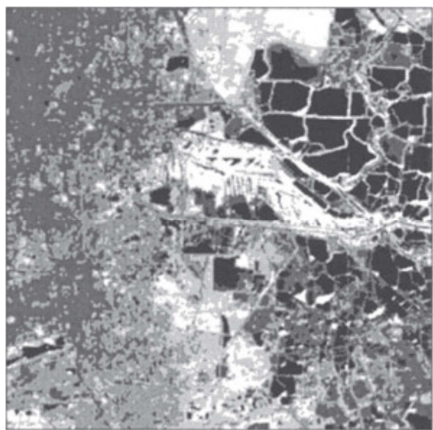
(b)

Figure 4. *Segmented Calcutta (SPOT) image using (a) FCM and (b) HCM [2]. © Springer-Verlag.*

2.5. Image Compression Using Soft Computing Tools

Image compression in this multimedia era is of utmost importance for saving the transmission time and the storage occupancy. For still images there exists JPEG compression standard which is based on the Discrete Cosine Transform (DCT). Replacing DCT by Karhunen Loeve Transform (KLT)

leads to better than JPEG results as KLT is the best in compaction of energy into coefficients which are near to the top left corner of image block. The problem is in computation time of KLT transform as no fast algorithm for KLT is known. The progress in hardware development can change the situation and the possibility to compute KLT adaptively to the image data in the frame stream can be an attractive feature [2].

Pixel neural networks (PNN) are a special type of recurrent networks which can define 2D patterns (images). The topology of connection can be very sparse and still very good approximation of real-life pictures can be obtained. A special subclass, Fractal Blocked Pixel Neural Network (FBPNN), working in the parallel mode represents a fractal operator used for image compression. FBPNN working in sequential stochastic mode converge faster and use less memory (by about 50%) than fractal operators. Moreover, PNN networks can be used as components of a high-performance associative memory [2]. Figure 5 below shows the compression results obtained by FBPNN approach.

Figure 5. *Original Lena and images obtained using FBPNN with PSNR=21.16, 26.42, 30.35, 32.22, 32.79 [dB], respectively [2]. © Springer - Verlag.*

He et. al. [2] have proposed a Genetic Algorithm and Fuzzy Reasoning based Digital Image Compression scheme using Triangular Plane Patches (TPP). The compression method using TPP [14] can be looked upon as another version of VQ. The TPP method considers the given image as 3-dimensionalluminance

curved surface. Based on this consideration, it recursively divides an image into square blocks of variable size. For each block, 2 triangular plane patches are used to approximate the imaged luminance curved surface over the block. Each triangular plane patch is decided by the luminance values of 3 pixels at the corresponding vertices in the block. The division procedure does not stop unless the distortion on a block is less than the previously defined allowable threshold. As a result, the original image is represented by a quadtree, and every its leaf stands for a block obtained from the image division procedure. It is obvious that the compression rate, under a fixed allowable threshold, strongly depends on the number of leaves in the quadtree. Genetic Algorithm (GA) is then used to find the optimal triangular plane patches for a block. The method, called GA-TPP method, for each block that is being processed, an individual consists of 4 luminance values which represent 4 pixels of the vertices of the block. The fitness of an individual is evaluated by the distortion relevant to the individual. The inferior individuals are eliminated and those superior ones are applied for further computation. In this way, 2 optimal triangular plane patches for a block can be finally obtained. The experimental results show that this method can decrease the average distortion, avoid excess block splitting, and increase the compression rate [2].

Wang et. al. [2] have proposed an image compression system design for digital mammograms using wavelet image decomposition and vector quantization. In digital mammograms, important diagnostic features such as the microcalcifications appear in small clusters of few pixels with relatively high intensity compared with their neighboring pixels. These image features can be preserved by a compression scheme employing a suitable image transform which can localize the signal characteristics in the original and the transform domain. Image compression is achieved by using wavelet filters to decompose digital mammograms into subbands carrying different frequencies. The resulting subbands are then encoded using vector quantization. Vector quantization is based on multiresolution codebooks designed by the Linde-Buzo-Gray (LBG) algorithm and a family of fuzzy algorithms for learning vector quantization (FALVQ) [2].
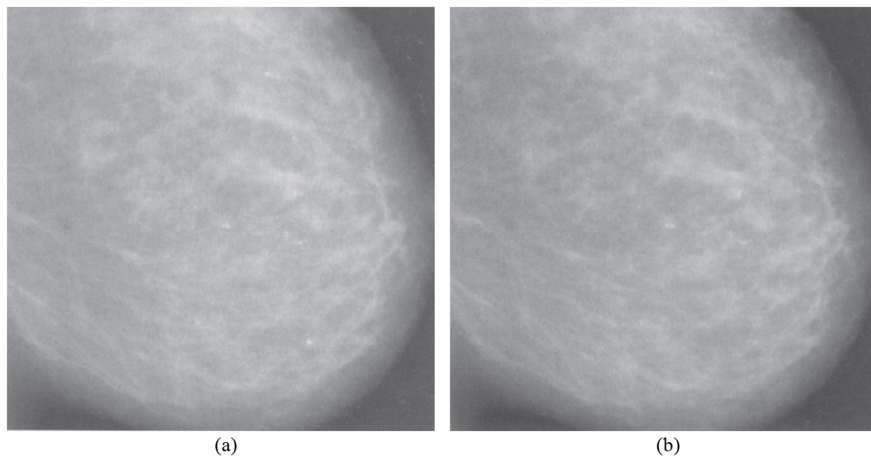

(a)   (b)

Figure 6. *(a) Original and (b) Compressed mammogram using Wavelet-FLVQ method (CR = 6.89, PSNR = 44.99 dB) [2]. © Springer- Verlag.*

3. Miscellaneous Applications

This section describes miscellaneous applications like automatic target recognition, facial analysis and processing, handwritten digit recognition, motion analysis and estimation.

3.1. Automatic Target Recognition Using ANN

An automatic target recognizer (ATR) is an algorithm that locates targets in an image and identifies the types of the targets. Most ATR designs consist of several stages [15]: At the first stage, a target detector, operating on the entire image, isolates local regions of interest (ROI) that have target-like characteristics. These ROIs are examined by another stage that attempts to reject false target-like objects (clutter) and retain targets. At the third stage, a set of features is computed. The selected features must effectively represent the target image. The fourth stage classifies each target image into one of a number of classes by using the features calculated at the previous stage. A number of learning algorithm based automatic target recognizers have been developed. The approaches differ in how features for recognition are extracted and in the architecture of the recognizer. Features are either extracted automatically by a multilayer convolutional neural network, or chosen by the designer based on experiment and previous experience. Recognizer complexity is kept low by decomposing the learning tasks using modular components or imposing an architecture that is not fully connected. Three types of recognizers have been developed- modular neural network (MNN), learning vector quantization (LVQ), and convolutional neural network (CNN) by *Nasrabadi et. al [2]*. MNN uses modular neural networks operating on local directional variances of the image. LVQ uses the Haar wavelet decomposition of the input images as features, clusters training features into templates using the K-means algorithm, and then enhances the recognition capability of the templates using learning vector quantization. CNN operates directly on input images without any preliminary feature extraction stage. The multilayer convolutional neural network simultaneously learns features and how to classify them [2].

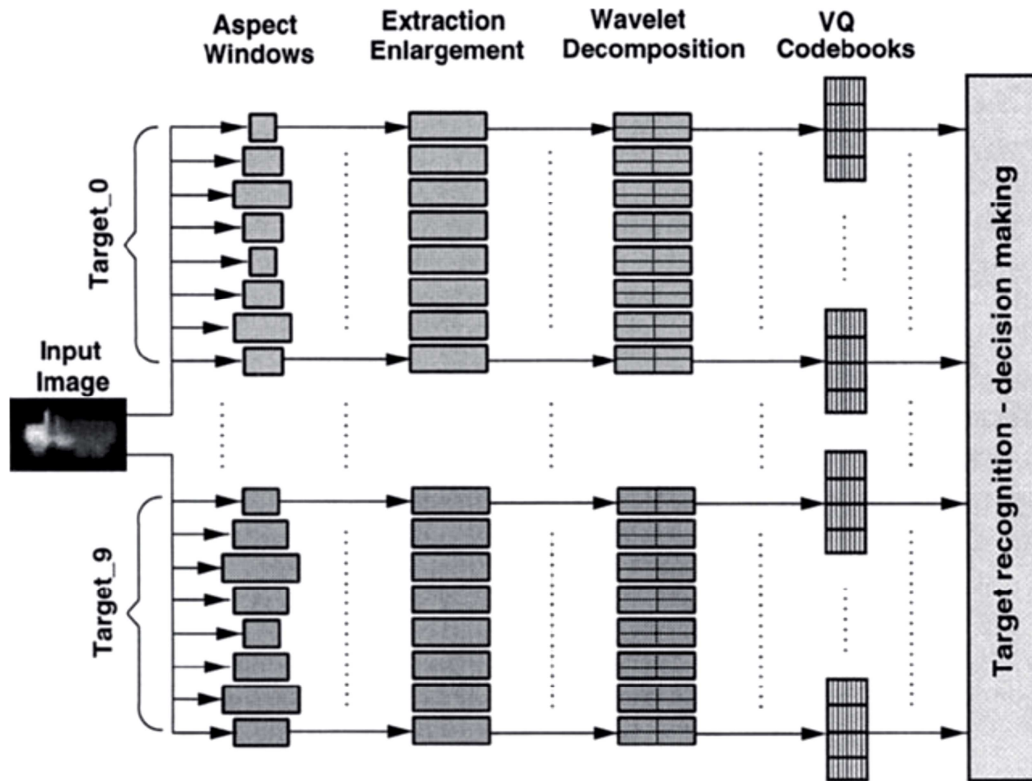Figure 7 shows a typical LVQ based ATR structure.

Figure 7. *Architecture of LVQ-based automatic target recognition classifier [2]. © Springer- Verlag.*

3.2. Hybrid Systems for Facial Analysis and Processing

Face Processing is a difficult task mostly because of the inherent variability of the image formation process in terms of image quality and photometry, geometry, occlusion, change, and disguise. Two recent surveys on face processing discuss these challenges in some detail [3, 4]. Most face processing systems available today can only perform on restricted data bases of images in terms of size, age, gender, and/'Or race, and they further assume well controlled environments. There are additional degrees of variability ranging from those assuming that the position/cropping of the face and its environment (distance and illumination) are totally controlled, to those involving little or no control over the background and viewpoint, and eventually to those allowing for major changes in facial appearance due to factors such as aging and disguise (hat and/or glasses).

As intelligent highways and multimedia applications are being developed it becomes imperative to develop robust classification and retrieval schemes. *Gutta et. al. [2]* have addressed those concerns by considering *hybrid systems* and showing their feasibility on large databases consisting of facial images. The hybrid architectures, consisting of an ensemble of connectionist networks - radial basis functions (RBF) - and inductive decision trees (DT), combine the merits of 'holistic' template matching with those of 'discrete' features using both positive and negative learning. The specific characteristics of this hybrid architecture include (a) query by consensus as provided by ensembles of networks for coping with the inherent variability of the image

formation and data acquisition process, (b) categorical classifications using decision trees, (c) flexible and adaptive thresholds as opposed to ad hoc and hard thresholds, and (d) interpretability of the way classification and retrieval is eventually achieved [2].

The hybrid classifier architecture for facial analysis and processing tasks is shown in Figure 8. The motivation for this architecture comes from (a) the apparent need to process imagery at different levels of granularity, like those provided by connectionist and symbolic approaches, and (b) integration of local and global processes. As it was discussed earlier, face recognition starts through the detection of a pattern as a face and boxing it, proceeds by normalizing the face image to account for geometrical and illumination changes using information about the box surrounding the face and/or eyes location, and finally it identifies the face using appropriate image representation and classification algorithms.. The hybrid classifiers consist of ensemble of connectionist networks - radial basis functions (RBF) - and inductive decision trees (DT). The reasons behind using RBF are its ability for clustering similar images before classifying them and the potential for developing in the future hierarchical rather than linear classifiers where faces will be iteratively discriminated in terms of gender, race, and age, before final recognition would take place. Decision trees (DT) implement the symbolic stage using RBF outputs because they provide for flexible and adaptive thresholds, and can interpret ('explain') the way classification and retrieval are eventually achieved [2].
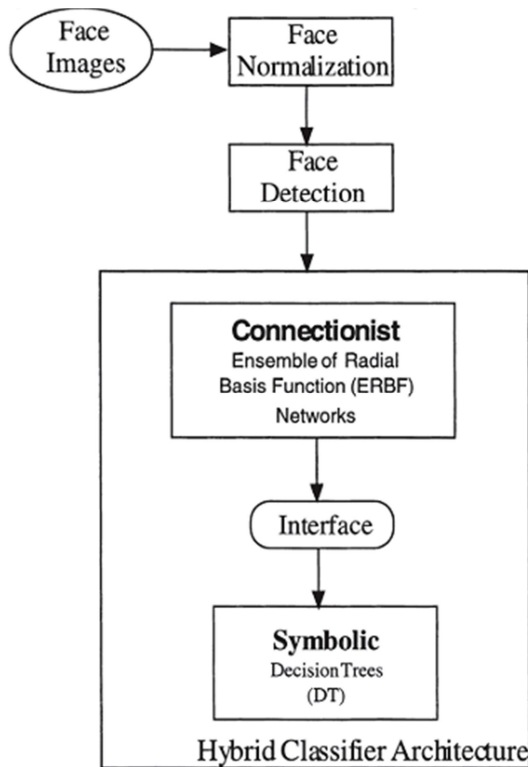
Figure 8. Hybrid classifier architecture [2]. © Springer- Verlag.

3.3. Handwritten Digit Recognition Using Soft Computing Tools

V. Susheela et. al. [2] have used the nearest neighbor classifiers and fuzzy classifiers for handwritten digit recognition. Machine recognition of handwritten text is one of the challenging areas of research for those working in the field of automatic pattern recognition and classification. This is because handwritten text yields complicated shapes and patterns that are among the most difficult to classify accurately. In fact, this is a good example of an area where the native ability of humans is far superior to that of any machine existing today. The automatic recognition of handwritten text is a problem well worth solving because a solution would enable us to design and use a more convenient and flexible communication interface between man and machine. The different categories of handwriting recognition can be listed in an increasing order of complexity as follows:

- recognition of individual alphanumeric characters
- recognition of words in a language
- recognition of personal signatures

A simple real-world example for the least complex problem in the above list is the recognition of the ZIP or PIN code as written on envelopes sent through the post office. In this example, a typical component of the problem is the recognition of the individual digits comprising the PIN code. The ZIP or PIN code must be segmented into individual digits and then these digits must be classified and labelled before a complete recognition of the code is achieved. Segmentation is a difficult problem. Even the seemingly simpler problem of individual digit recognition is an unsolved problem. The handwritten digit recognition problem may be approached in one of two modes: online or offline. Online recognition systems make use of a special hardware to obtain dynamic information and use it for classification. However, they are not attractive for massive use because of special hardware requirement. In offline recognition, the digits written on a conventionally used material like paper or envelope are scanned, digitized and stored on a machine. This data is used for recognizing the digits. Here, the recognition system does not have access to dynamic information of the digit like the number of strokes, the speed of writing, and the pressure applied in writing. So, classification using the offline data is more difficult. However, offline recognition is more popular because of its pragmatic viability.

V. Susheela et al. [2] have examined the role of a variety of neighborhood classifiers including the Nearest Neighbor classifier (NNC), K-Nearest Neighbor Classifier (KNNC), and a modified KNNC [3] in the classification of handwritten digits. These classifiers make use of all the patterns in the training data set, thus requiring a large amount of classification time. Condensed Nearest Neighbor Classifier (CNNC) is useful in this context as it selects an appropriate subset [16] of the training data set and uses this data set for classification. We present the results obtained using these classifiers on the chosen data sets. Even though CNNC reduces the size of the training data set, it is order-dependent and as a consequence, the condensed data set generated by it may not be the smallest possible in size. Different methods for selecting a subset of the training data set, called the prototype data set, have been examined. The handwritten digit recognition problem has been formulated as an optimization problem and genetic algorithms (GAs), simulated annealing (SA), and tabu search (TS) have been used to solve this optimization problem. The k-means algorithm (KMA) and fuzzy c-means algorithm (FCMA) have been used to obtain the prototype patterns. Tables 1 and 2 show the results obtained by NNC classifier results for handwritten digit recognition using 192 and 48 features, respectively.

Table 1. NNC Results Using 192 Features [2]. © Springer- Verlag.

Number of Training Patterns	Accuracy (%)		
	NNC	KNNC	MKNNC
6670 (all)	91.50	91.60	92.57
6670 (pre-proc)	92.17	92.50	93.10
1715 (condensed)	86.20	87.19	90.60

Table 2. NNC Results Using 48 Features [2]. © Springer- Verlag.

Number of Training Patterns	Accuracy (%)		
	NNC	KNNC	MKNNC
6670 (all)	85.00	86.20	88.18
6670 (pre-proc)	92.71	92.62	93.49
1715 (condensed)	76.33	81.55	82.84

3.4. Motion Analysis and Estimation Using Neural / Neuro-fuzzy Systems

Motion analysis (MA) refers to the analysis and interpretation of movements of objects over time. In the

computer vision domain, MA has been emerging actively over the years due to the advancement of video camera technologies and the availability of more sophisticated computer vision algorithms in the public domain. Video surveillance is one of the most important real-time applications of motion analysis. Apart from that, MA also contributed in video retrieval, sports analysis, healthcare monitoring, human-computer interaction and so on [17].

3.4.1. Neural Systems for Motion Analysis

Recently, dynamic scene analysis in the machine vision field has been given increasing attention. The input to a dynamic scene analysis system is a series of images, each representing the scene at a particular time instant. The images can be produced by disparate sensors, giving information about each time step in multiple channels. One of the goals of the analysis is the derivation of three-dimensional translational and rotational motion parameters. These motion parameters can then be used to derive the orientation of surfaces in the images (structure from motion) for further operations such as model based matching. Point correspondence across frames is done through a color segmentation step [18], followed by region matching based on

shape, size and position [19]. Motion analysis was done on the centroids of the regions after correspondence was established over more than two frames. In their system, the type of motion was classified using an error measure based on the hypothesis that the motion is pure translation [20]. Subsequent steps in their analysis refine this guess using a hierarchical generate and test scheme. Some real-world sequences were investigated in their study, and an error analysis comparison for noise sensitivity to other methods indicated that the technique was very robust [2].

One of the most comprehensive artificial neural network studies of motion perception is being pursued at the Centre for Adaptive Systems at Boston University. The frameworks chosen for their models are biologically motivated and have succeeded in replicating a wide range of visual phenomena including optical illusions and most motion paradigms. The study by Grossberg and Rudd extended the static Boundary Contour System (BCS) to motion processing [21]. *Huntsberger et. al. [2]* used Jordan-Elman neuron model [22], [23] for estimating the motion in a travelling car sequence. Figure 9 shows the results produced by applying this model on the given car sequence.
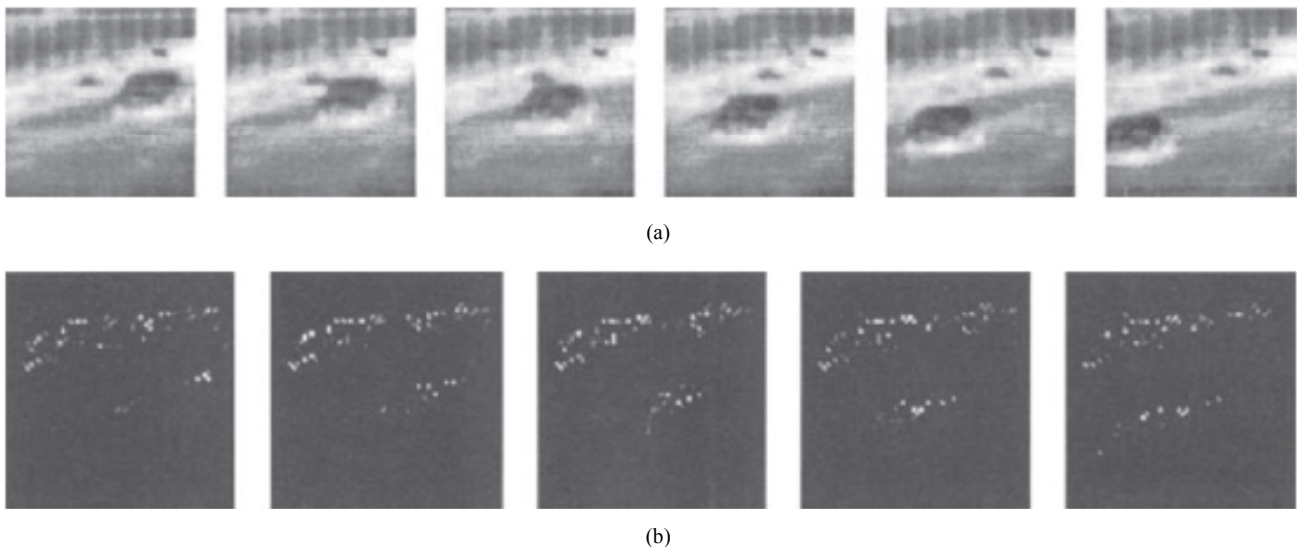

(a)

(b)

Figure 9. A six-frame subsequence showing a moving car (a), The output of units sensitive to movement from bottom left to upper right (b) [2]. © Springer-Verlag.

3.4.2. Motion Estimation and Compensation Using Neuro-fuzzy Systems

Motion estimation and compensation are key parts of video compression. They help remove temporal redundancies in images. Most motion estimation algorithms neglect the strong temporal correlations within the motion field. The search windows stay the same through the image sequences and the estimation needs heavy computation. *Kim and Kosko [2]* proposed a novel algorithm- *neural fuzzy overlapped block motion compensation (FOBMC)* scheme for motion estimation and compensation. The motion estimator uses unsupervised neural "competitive" learning to estimate motion vectors. The motion compensator uses an adaptive fuzzy system to

compensate for overlapped block motion. Motion estimation and compensation help compress video images because they can remove temporal redundancies in the image data. But motion estimation schemes often neglect the strong temporal correlations within the motion field. The search windows remain the same through the image sequences and the estimation may need heavy computation. Below we design an unsupervised neural system that uses the temporal correlation of the motion field both to estimate the motion vectors and to reduce the entropy of source coding. Motion-compensated video coding uses the motion of objects in the scene to relate the intensity of each pixel in the current frame to the intensity of some pixel in a prior frame. It predicts the value of the entire current block of pixels as the value of a displaced block from

the prior frame. It also assumes that each block of pixels moves with uniform translational motion. This assumption often does not hold and can produce block artifacts. The designed neural-fuzzy system that uses motion vectors of neighbouring blocks to improve the compensation accuracy.

The neural quantizer system proposed by *Kim and Kosko [2]* uses the first-order and second-order statistics of the motion vectors to give ellipsoidal search windows. This method reduces the search area and gives clustered motion fields. It reduces the computation for motion estimation and decreases the entropy that the system needs to transmit the entropy-coded motion vectors. Fuzzy systems use *if-then* rules to map inputs to outputs. Neural fuzzy systems learn the rules from data and tune the rules with new data. The FOBMC estimates each pixel intensity using the block-based motion vectors available to the decoder. The fuzzy system uses the motion vectors of neighbouring blocks to map the prior frame's pixel values to the current pixel value. The 196 rules come from the prior decoded frame. The neural fuzzy system tunes its rules as it decodes the image. The fuzzy system defined a nonlinear "black box" function approximator that improved the compensation accuracy. The appendix derives the supervised neural-like learning laws that tune the parameters of the fuzzy system [2].

4. Conclusion

In this paper, we have attempted to compile the available soft computing tools and algorithms for various image processing applications. These applications include image filtering/ enhancement, edge extraction, segmentation, object/ face/ target recognition, compression, handwritten digit recognition, motion estimation etc. The soft computing tools used for these applications include neural networks (NN), pixel neural networks (PNN), Fuzzy c-means clustering (FCM), hard c-means clustering (HCM), modular neural network (MNN), learning vector quantization (LVQ), and convolutional neural network (CNN), Nearest Neighbor classifier (NNC), K-Nearest Neighbor Classifier (KNNC), neural fuzzy systems etc. Tools like genetic algorithms (GAs), simulated annealing (SA), and tabu search (TS) etc. have been incorporated with soft computing tools for applications involving optimization.

References

[1] Kurhe A. B., Satonkar S. S., Khanale P. B. and Shinde Ashok, "Soft computing and its applications", BIOINFO Soft Computing, Volume 1, Issue 1, 2011, pp-05-07.

[2] Sankar K. Pal, Ashish Ghosh, and Malay K. Kundu, "Soft Computing for Image Processing", Part of the Springer book series *Studies in Fuzziness and Soft Computing*, 2000, Volume 42, ISBN: 978-3-7908-2468-1. DOI: https://doi.org/10.1007/978-3-7908-1858-1.

[3] J. Bezdek and S. K. Pal Editors. *Fuzzy Models for Pattern Recognition*. IEEE Press, Boca Raton, 1992.

[4] Y. H. Pao. *Adaptive Pattern Recognition and Neural Networks*. Addison Wesley, New York, 1989.

[5] S. Newton, S. Pemmaraju, and S. Mitra. Adaptive fuzzy leader clustering of complex data sets in pattern recognition. *IEEE Trans. Neural Networks*, 3 (5): 794-800, 1992.

[6] Y. Kim and S. Mitra. An adaptive integrated fuzzy clustering model for pattern recognition. *Fuzzy Sets and Systems,* 65: 297-310, 1994.

[7] Cannon, R. 1., Dave, R., Bezdek, J. C, and Trivedi, M. (1986) Segmentation of a thematic mapper image using the fuzzy c-means clustering algorithm. *IEEE Transactions on Geoscience and Remote Sensing,* 24, 400-408.

[8] Laprade, R. H. (1988) Split-and-merge segmentation of aerial photographs. *Computer Vision, Graphics and Image Processing,* 44, 77-86.

[9] MandaI, D. P., Murthy, C. A and Pal, S. K. (1994) Utility of multiple choices is detecting ill-defined roadlike structures. *Fuzzy Sets and Systems,* 64, 213-228.

[10] Sahasrabudhe, S. C. and Dasgupta, S. C. (1992) A valley-seeking threshold selection technique. In *Computer Vision and Image Processing,* L. Shapiro and A. Rosenfeld eds., (Boston: Academic Press), 55-65.

[11] Barzohar, M. and Cooper, D. B. (1993) Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 459-464.

[12] Hu, J., Sakoda, B. and Pavlidis, T. (1992) Interactive road finding for aerial images. In *Applications of Computer Vision, 56-63.*

[13] Zlotnick, A. and Carnine (Jr.), P. D. (1993) Finding road seeds in aerial images. *Image Understanding,* 57, 307-330.

[14] I. Yamasaki, M. Hasegawa, S. Ikarashi and S. Okada (1992) Data compression of digital images with grey level using triangular plane patches. Trans. of IEICE, Vol. J75-D-II, No. 6, 1038-1047.

[15] Bhanu, B. (1986): Automatic target recognition: state of the art survey. IEEE Trans. Aerospace Elect. Systems, 22 (4), 364-379.

[16] Hart, P. E. (196S): The Condensed Nearest Neighbour Rule. *IEEE Trans. on Information Theory* IT-14, 515-516.

[17] Chern Hong Lim, Ekta Vats and Chee Seng Chan, "Fuzzy human motion analysis: A review", Pattern Recognition, Vol. 48, Issue 5, May 2015, 1773-1796.

[18] Oblander, R., K. Price, and D. R. Reddy (1978). Picture segmentation using a recursive region splitting method. Comput. Graphics Image Processing 8, 313-333.

[19] Faugeras, O. D. and K. E. Price (1981). Semantic description of aerial images using stochastic labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 3, 633-642.

[20] Shariat, H. and K. E. Price (1990). Motion estimation with more than two images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 417-434.

[21] Grossberg, S. and M. E. Rudd (1989). A neural architecture for visual motion perception: Group and element apparent motion. Neural Networks 2 (6), 421-450.

[22] Jordan, M. J. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In Proc. Eighth Ann Conf. of Cognitive Science Society, pp. 531-546.

[23] Ellman, J. (1990). Finding structure in time. Cognitive Science 14, 179-211.