# A Pruning Algorithm for Efficient Image Segmentation with Neighborhood Neural Networks

Siddhartha Bhattacharyya * Ujjwal Maulik † Paramartha Dutta ‡

*Abstract* —**Most of the image preprocessing techniques by existing neighborhood neural networks, suffer from the problem of false classification of the image features. This is mainly due to the redundancy in the interconnectivity patterns of the networks. The larger number of interconnections in these networks implies a larger network complexity as well.**
**A fuzzy set-theoretic pruning algorithm to refine the interconnection pattern of neighborhood neural network architectures, is presented. The algorithm is primarily focussed on the judicious selection of the participating neurons of the network topology. Efficiency of the pruning algorithm is demonstrated with the segmentation and extraction of synthetic and real life images. The universality of the algorithm is evident from its application in the binary, multilevel and color domains.**

***Keywords: fuzzy cardinality, image segmentation, neighborhood neural networks, pruning algorithms***

## 1 Introduction

Most of the image preprocessing techniques suffer from the problem of false classification of the image features owing to the methodologies adopted in the classification procedure. Enhancement and restoration of images through filtering techniques and morphological approaches [1][2], report misclassification due to misrepresentation of the information content. This problem becomes more severe in noisy environments. This is due to the fact that the probability of erroneous classification of an image feature into a noise feature and *vice versa*, increases with the degree of noise levels. However, the designing of appropriate filters and choice of morphological parameters require appreciable *a priori* information regarding the feature space and the degree of underlying noise content.

The neural networking paradigm has often been employed for the purpose of processing of noisy images, leading to faithful extraction of regions of interest in the images [3]-[5]. The multilayer self organizing neural network (MLSONN) [6] is a feedforward neighborhood neural network architecture, efficient for the extraction of objects from noisy and blurred perspectives, by self supervision of the image pixel neighborhood information.

Most of the existing neural network topologies, remain highly interconnected through the interconnections between their neurons. Thus, a larger number of neurons would mean a larger number of interconnections. The obvious fallout is a larger network complexity. Moreover, redundancy in the network interconnections also leads to the problem of false classification and recognition.

Researchers have tried out several methods for arriving at an optimized neural network topology, which would sever the redundant interconnections in the architecture, thereby evolving time and space-efficient topologies [7]-[10]. Opitz and Shavlik [11] proposed the genetic algorithm based pruning algorithm (REGENT) for refining the network topology. However, the use of genetic algorithm based search techniques leads to computational overhead, which prevents the application of these optimized networks from real time operations.

Other notable contributions in this direction include the intuitive pruning methods based on weight and unit activation values [12], magnitude-based pruning (MBP) [13] which assumes that small weights are irrelevant. A number of attempts based on approximations to the Fisher information matrix for determining the optimal number of hidden units and weights of neural networks figure in the literature. Cottrell *et al.* [16] represented the interconnection weights as functions of the information matrix. Approaches based on singular value decompositions (SVD) of the hidden unit activation covariance matrices for determining the optimal number of hidden units are reported in [17][18]. However, these techniques are restricted to weights between the hidden and output layer only. Principal component analysis (PCA) based pruning techniques [19][20] use the SVD of the Fisher information matrix to find the linear transformations of the original parameters of the network topologies. The nonprincipal

*Department of Computer Science and Information Technology, University Institute of Technology, The University of Burdwan, Burdwan 713 104, India Email: siddhartha.bhattacharyya@gmail.com

†Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, India Email: drumaulik@cse.jdvu.ac.in

‡Department of Computer and System Sciences, Visva-Bharati University, Santiniketan 731 235, India Email: paramartha.dutta@gmail.com

(Advance online publication: 20 May 2008)

components are pruned since they do not account for data variance.

Statistical analysis of the network parameters have been used often to determine the neurons to undergo pruning. Steppe *et al.* [21] used the likelihood-ratio test statistics of the interconnection weights to refine the network topology. They pruned those hidden units having weights statistically different from zero. Belue and Bauer [22] injected a noisy input parameter into a neural network model and decided upon the relative significance of the original network parameters with the injected noisy parameter. Parameters with lower significance than the noisy parameter are pruned. Prechelt [23] developed methods to determine the number of weights to be pruned.

Sensitivity analysis based pruning techniques, which model the network parameters as small perturbations to the network sensitivity, have been widely used for evolving pruned network topologies. Zurada *et al.* [25][26] removed redundant network input units by means of the sensitivity analysis of the network output function with respect to perturbations of input units. This approach was further extended in [27][28]. Ponnapalli *et al.* [29] devised a formal pruning technique for reducing redundancy in feedforward neural network architectures, based on the concept of sensitivity index proposed by Karnin [24]. In the optimal brain damage (OBD) [30], optimal brain surgeon (OBS) [31] and optimal cell damage (OCD) [32], sensitivity analysis is performed with regard to the training error, while network generalization error forms the basis of pruning mechanism in [33].

In this article, a fuzzy set-theoretic neighborhood topology pruning strategy for efficient image segmentation and extraction, is proposed. Full connectivity in neighborhood topology-based neural networks for processing of neighborhood information, brings in false classification of object pixels as noise pixels and *vice versa*. This problem of misclassification gets aggravated at higher noise levels. A fuzzy cardinality estimate of the image pixels is used to prune the network architecture through a judicious selection of the relevant participating neurons. Results are reported on the segmentation of images from a noisy and noise-free background.

## 2 Relevant fuzzy set-theoretic concepts

A fuzzy set $A$, [34][35] comprises elements characterized by a certain degree of membership, $\mu_A(x)$ lying in [0, 1]. The resolution of a fuzzy set $A$ is determined by the $\alpha$-cut (or $\alpha$-level set) of the fuzzy set. It is a crisp set $A_\alpha$ that contains all those elements of the universal set $U$ with membership in $A$ greater than or equal to $\alpha$, i.e.

$$A_\alpha = \{x \in U | \mu_A(x) \geq \alpha\}, \alpha \in [0, 1] \quad (1)$$

If $A_\alpha = \{x \in U | \mu_A(x) > \alpha\}$, then $A_\alpha$ is called a strong $\alpha$-cut. The set of all levels $\alpha \in [0, 1]$ that represents distinct $\alpha$-cuts of a given fuzzy set $A$, is called a level set of $A$, i.e.,

$$\Lambda_A = \{\alpha | \mu_A(x) = \alpha, x \in U\} \quad (2)$$

For a fuzzy set $A$ with finite support, the fuzzy cardinality $(A_f)$ of the set is defined as the summation of the membership grades of all elements of $x$ in $A$. It is given by [34][35]

$$|A_f| = \sum_{\alpha \in \Lambda_A} \frac{\alpha}{|A_\alpha|} \quad (3)$$

where, $\alpha$ is the cut-off value, $A_\alpha$ is the $\alpha$-level set of the fuzzy set and $\Lambda_A$ is the corresponding level set. Since the fuzzy cardinality is directly proportional to the memberships of the constituent elements in a fuzzy set, it indicates the relative proportion of the higher and lower membership elements in the set.

The subnormal linear index of fuzziness for a subnormal fuzzy set $A_s$, $\nu_l(A_s)$ is defined as

$$\nu_l(A_s) = \frac{2}{n} \sum_{i=1}^{n} [\min\{\mu_{A_s}(x_i) - L, U - \mu_{A_s}(x_i)\}] \quad (4)$$

## 3 Problems of redundant network interconnectivity

A neighborhood topology-based neural network in a layered architecture, is a fully connected network structure comprising neurons which correspond to the pixels in an image. The neurons of a particular layer of the network architecture are connected to the corresponding neurons and their neighbors in the other layers following a neighborhood topology. These neurons self organize the cumulative intensity information of the neighborhoods of the pixels in the image.

This full connectivity ensures the propagation of the neighborhood information from one layer of the network to the other. However, in a noisy image, each object pixel is encircled by pixels which correspond to either the object or the noise. In such a scenario, full connectivity among the neurons of such networks implies contribution from both the object and the noise pixels. This leads to false classification of an object pixel into a noisy pixel and *vice versa*. As the noise level increases, this problem gets aggravated. Thus, redundancy in interconnections leads to misclassification of object pixels as noise pixels. A reduced network architecture can be evolved by pruning these redundant interconnections between the different layers, thereby enabling the network to classify object and noise pixels more efficiently. Moreover, pruning of the redundant interconnections also reduces the overall processing time of the network self organization process.

## 4 Neighborhood topology-based neural network architecture

Several pixel neighborhood systems constitute an image. A neighborhood system is made up of a candidate pixel encircled by a number of neighbors. Different orders of neighborhoods are possible. A $n^{th}$ order pixel neighborhood system comprises $2^{(n+1)}$ neighboring pixels. However, due to the fixed size of the image, the neighborhoods of the pixels on the boundaries are necessarily smaller. For an $M \times N$ image, the candidate pixels at the corners of the lattice are encircled by only three neighboring pixels. Moreover, for the candidate pixels lying on the rim of the lattice, there are only five neighbors.

A layered architecture of a neighborhood topology-based neural network comprises different neighborhood topology-based network layers of neurons. The aggregate information of the candidate pixels along with the respective neighbors are used by these networks for the purpose of object extraction. This information is propagated from the input to the other layers of the network, which in turn are activated by an activation/transfer function characteristic to the network. Subsequently, network responses or outputs are generated at the different layers depending on the nature of the characteristic activation/transfer function. Thus, the input information is used to generate outputs at the different layers of the neighborhood topology-based neural network.

An example of a layered neighborhood topology-based neural network architecture is the multilayer self organizing neural network (MLSONN) [6] architecture. It is a feedforward network architecture characterized by a neighborhood topology-based network interconnectivity. It comprises an input layer, any number of hidden layers and an output layer. The network operates in a self supervised mode featuring backpropagation of errors and feedback of outputs. The system errors are determined from the linear indices of fuzziness in the network outputs obtained. As a result, this type of network is suitable for real time operations as compared to the supervised neural network architectures, where the training time required for the training of inputs poses additional computational burdens. A detailed analysis of the architecture and operation of the MLSONN architecture can be found in [6].

## 5 Parallel neighborhood topology-based neural network architecture

Color images are natural extensions of the binary and multilevel images. Similar to the binary images, pure color image intensity levels are manifested through two levels of intensity. These images comprise the primary component color information and their admixtures in either 0 or 255. True color images, on the contrary, exhibit intensity levels of the primary color components and their admixtures in all possible shades ranging from 0 to 255. This aspect is similar to the multilevel images, which also comprise intensity levels in the range of [0, 255]. The conventional neighborhood neural network architectures are unable to handle color image information on their own. At least three independent neighborhood neural networks are needed for processing the individual color component information. Hence an architectural modification is an obvious choice.

A parallel neighborhood topology-based neural network is a parallel extension of the neighborhood topology-based architecture. It comprises a collection of neighborhood neural networks operating in parallel. The number of such neighborhood neural networks in the collection depends on the number of color components to be processed. The parallel multilayer self organizing neural network (PSONN) [36] architecture is an example parallel neighborhood neural network architecture. The PSONN architecture is efficient in retrieving objects from a pure color noisy image. It comprises three independent three layer self organizing neural networks (TLSONN) (a subset of the MLSONN architecture) for processing of the RGB triplet information of the color image. In addition, a source layer is present in the PSONN architecture for the distribution of the RGB triplet information to the input layers of the constituent independent TLSONN architectures. Another sink layer fuses the extracted/segmented output component information obtained at the constituent TLSONN output layers, into extracted color outputs. Interested readers may refer to [36] for details.

## 6 Multilevel activation function

The main drawback of the MLSONN [6] architecture or its subset TLSONN architecture is its inability to respond to multilevel image information, i.e. information exhibited in different scales of intensity levels. This limitation is imposed by the bilevel transfer characteristics of the architecture, which only generate binary responses to the input information.

Multilevel responses can be induced in the architecture by employing a multilevel form of the characteristic sigmoidal activation function, which can generate subnormal responses in the range of [0, 1]. The multilevel form of the sigmoidal activation function can be defined as

$$f_{\text{MUSIG}}(x; \alpha_\gamma, c_\gamma) = \sum_{\gamma=1}^{K-1} \frac{1}{\alpha_\gamma + e^{-\lambda[x-(\gamma-1)c_\gamma-\theta]}} \quad (5)$$

where, $\gamma$ represents the gray scale object index ($1 \leq \gamma < K$). $c_\gamma$ represents the gray scale contribution of the $\gamma^{th}$ class. $\lambda$ controls the slope of the function and $\theta$ is a threshold/bias value. $K$ is the number of gray scale objects or classes. $\alpha_\gamma$ controls the multilevel class responses. It is related to $c_\gamma$ by

$$\alpha_\gamma = \frac{1}{c_\gamma \times C_N} \quad (6)$$

where, $C_N$ is the neighborhood gray scale contribution. Since the MUSIG activation function generates multilevel responses, an MLSONN architecture guided by this function, would be able to segment multilevel images. The network system errors would then be estimated from the subnormal linear indices of fuzziness given by equation 4. The PSONN architecture is also constrained in the segmentation of true color images owing to the aforementioned inherent limitation of the constituent TLSONN architectures. The proposed functional modification in the MLSONN architecture for inducing multilevel responses would also enable the PSONN architecture in generating multicolor responses to true color image information. Hence, if the transfer characteristics of each of the three constituent TLSONN architectures are controlled by MUSIG activation functions, the PSONN architecture would be able to segment true color images as well.

## 7 Basis of fuzzy set-theoretic pruning

A TLSONN architecture or its parallel version (PSONN) enjoys full interconnectivity between the constituent neurons of different layers. Hence, these architectures are prone to false classification of image features. Pruned networks evolved through a selective choice of the participating neighbors of the candidate neurons in these architectures, would be able to reduce the problem of misclassification.

As TLSONN neurons correspond to the pixels in the input image information, a judicious selection of the participating neurons of network neighborhood can be made by considering the similarity/dissimilarity in the intensity levels of the image pixels. Such a selection would be effective in reducing the network architecture without hampering its accuracy and performance. Obviously, the choice of the participating neurons tantamount to the selection of corresponding pixels in the pixel neighborhood systems in the input image. This selection should be essentially limited to those neighboring pixels which bear similarity to the candidate pixel in terms of the intensity level. The fuzzy cardinality estimate of a pixel neighborhood (as per equation 3), can be a guiding factor in arriving at the required decision of choosing neighboring neurons. A pruning algorithm can be devised taking into cognizance the fuzzy cardinality estimates of the pixel neighborhood regions. This is because the fuzzy cardinality estimates are reflective of the relative contributions of the higher and lower membership pixels in an image.

The interconnectivity pruning mechanism rests on the role of the neighbors of a candidate pixel in a pixel neighborhood system. Since, a fully connected neighborhood topology uses the cumulative neighborhood information, the interconnection weights between a neuron of one layer and the neighboring neurons in the previous layer are all fixed and full. This ensures equal participation of all the neighbors in the processing task. Due to this pattern, the possibility of a pixel to be classified as an object pixel increases when most of its neighbors belong to the object class. However, this also leads to misclassification of the said pixel in case some of the neighbors are corrupted with noise, since the noisy pixels also participate in the classification procedure.

One of the remedies of this problem would be to prevent the participation of those neurons which are not conducive to the processing task. These neurons can be identified from the relative contributions of the object and noise pixels in the neighborhoods. Subsequently, these neurons can be made inactive by assigning their interconnection weights with other neurons a value of 0. On the contrary, the active neurons as usual, would have their respective interconnection weights set to 1.

For an $\alpha-$cut value of 0.5 and a eight pixel neighborhood geometry, the limiting value of the fuzzy cardinality $A_{f(lim)}$ is equal to $\dfrac{0.5}{0.5 \times 8}$ i.e. $\dfrac{0.5}{4}$. This limiting fuzzy cardinality estimate indicates equal participation of background and object pixels in a pixel neighborhood. If the fuzzy cardinality of a pixel neighborhood is less than the limiting value, then the neighborhood is a brighter background perspective. On contrary, a larger limiting cardinality implies a darker background context. Hence, the MLSONN neighborhood topology can be pruned by means of a thresholding operation on the neighborhood fuzzy cardinality estimates of the input image pixel neighborhoods based on the limiting fuzzy cardinality ($A_{f(lim)}$).

The gray value ($G_C$) of the candidate pixel in the neighborhood and the gray values ($G_N$) of the neighboring pixels also play a vital role to decide the relevance/irrelevance of the candidate-neighbor connectivity strength ($C_{CN}$). The following network interconnectivity pruning algorithm can be devised taking into consideration these parameters along with the limiting cardinality estimate ($A_{f(lim)}$).

Begin
    Initialize $tot_{pixels} \leftarrow M \times N$ *(total no. of pixels)*
    Initialize $counter \leftarrow 1$
    Do
        Input $G_C$ and $G_N$
        Determine $A_f$
        If $A_f = A_{f(lim)}$ and $G_C \leq 0.5$ Then
            If $G_N > 0.5$ Then $C_{CN} = 1$
            Else $C_{CN} = 0$
            End If
        Else If $A_f = A_{f(lim)}$ and $G_C > 0.5$ Then
            If $G_N \leq 0.5$ Then $C_{CN} = 1$
            Else $C_{CN} = 0$
            End If
        Else If $A_f < A_{f(lim)}$ and $G_C \leq 0.5$ Then
            If $G_N > 0.5$ Then $C_{CN} = 1$
            Else $C_{CN} = 0$
            End If
        Else If $A_f < A_{f(lim)}$ and $G_C > 0.5$ Then
            If $G_N > 0.5$ Then $C_{CN} = 1$
            Else $C_{CN} = 0$
            End If

Else If $A_f > A_{f(lim)}$ and $G_C \leq 0.5$ Then
If $G_N \leq 0.5$ Then $C_{CN} = 1$
Else $C_{CN} = 0$
End If
Else If $A_f > A_{f(lim)}$ and $G_C > 0.5$ Then
If $G_N \leq 0.5$ Then $C_{CN} = 1$
Else $C_{CN} = 0$
End If
Else
$C_{CN} = 1$ with all $G_N$
End If
$counter \leftarrow counter + 1$
Until $counter = tot_{pixels}$
End

The pruning algorithm therefore, assigns full interconnections to a subset of the neurons of a neighborhood topology-based neural network architecture, subject to certain conditions based on the fuzzy cardinality estimates of the incident image pixel neighborhoods. Thus, the algorithm determines the active and inactive neighbors in an image pixel neighborhood. Hence, any misclassification that may arise otherwise due to fixed and full neighborhood connectivity, is avoided by a judicious choice of the active neighbors.

## 8 Implementation Results

The proposed pruning technique has been applied to a TLSONN architecture for the segmentation of noisy and noise-free binary, multilevel and color images of dimensions 128 × 128. This section summarizes the results of application of the algorithm.

### 8.1 Image segmentation with a pruned TLSONN architecture

Binary image segmentation followed by object extraction from a noisy background, has been implemented using a binary real life spanner image. A real life gray scale spanner image is used to illustrate multilevel image segmentation. Several degrees of Gaussian noise of zero mean and standard deviation of $\sigma$ = 8, 10, 12, 14 and 16 have been used to degrade the images. The noisy images with $\sigma$=14 and 16 are shown in Figure 1 (a)(b) and 1 (a)(b), respectively. The retrieved binary and gray scale images obtained with the fully connected TLSONN architectures for the different noise levels are shown in Figures 1 (a′)(b′) and 2 (a′)(b′), respectively. The corresponding retrieved images using the pruned TLSONN architecture are shown in Figures 2 (a″)(b″) and 2 (a″)(b″), respectively.
The reduction in the number of interconnections achieved through pruning, serves as a figure of merit of the fuzzy set-theoretic pruning algorithm. The total number of interconnections (*tnoc*) is 128 × 128 × 8 = 131072 in the fully connected TLSONN architecture with second order interconnectivity. The efficiency of the algorithm in terms of the proportion of reduction in the interconnections achieved, is presented in Table 1. From the table, it is evident that *tnoc* increases with an increase in the noise levels. This can be attributed to the fact that as noise levels increase, the probability of an object pixels to be surrounded by noise pixels also increases and a greater number of neighboring pixels become prohibitive.

Table 1: Reductions achieved during spanner image retrieval

| $\sigma$ | Binary image | | Gray scale image | |
|---|---|---|---|---|
| | *tnoc* | % reduction | *tnoc* | % reduction |
| 8 | 123247 | 5.97% | 127676 | 2.59% |
| 10 | 113406 | 13.48% | 127702 | 2.57% |
| 12 | 102088 | 22.11% | 122678 | 6.40% |
| 14 | 95652 | 27.02% | 117661 | 10.23% |
| 16 | 90986 | 30.58% | 115938 | 11.55% |

Table 2: *pcc* and $\rho$ values for retrieved spanner image

| $\sigma$ | *pcc* values for binary image | | $\rho$ values for gray scale image | |
|---|---|---|---|---|
| | Full network | Pruned network | Full network | Pruned network |
| 8 | 99.38 | 98.82 | 0.9108 | 0.9096 |
| 10 | 99.14 | 98.69 | 0.9025 | 0.9049 |
| 12 | 98.94 | 98.34 | 0.8831 | 0.8991 |
| 14 | 97.86 | 97.41 | 0.7980 | 0.8495 |
| 16 | 96.14 | 97.04 | 0.6418 | 0.6833 |

The retrieval efficiency of the algorithm can be judged from the quality of the retrieved images. The percentage of correct classification of pixels (*pcc*) [6] is used as the figure of merit for the retrieval of the binary spanner image. The quality of the retrieved gray scale spanner image is determined by the standard correlation coefficients ($\rho$) between the retrieved and the original images. Table 2 lists the *pcc* and $\rho$ values obtained by the fully connected and pruned TLSONN architectures corresponding to the different noise levels for the spanner images. It is clear from the tables that the pruned TLSONN architectures outperform the fully connected architecture at higher noise levels. This asserts that the proposed pruning algorithm is efficient in overcoming the problems of misclassification, which arises out of the redundant interconnections in a fully connected TLSONN architecture.

### 8.2 Image segmentation with a pruned TLSONN architecture

A PSONN architecture has been applied for the segmentation of a pure color image degraded with Gaussian

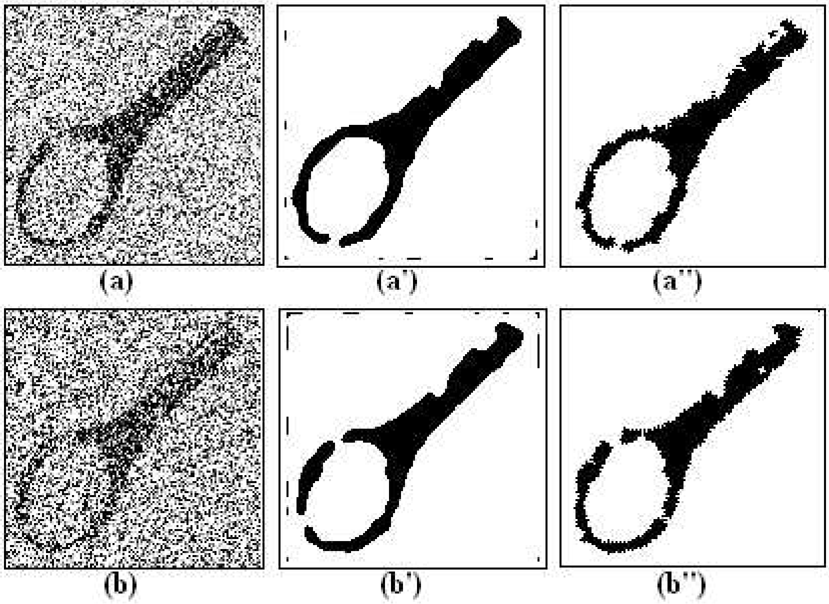

Figure 1: Noisy and retrieved real life binary spanner images at $\sigma = 14$ and 16 (a) (b) noisy images (a′)(b′) extracted with fully connected network (a″)(b″) extracted with pruned network

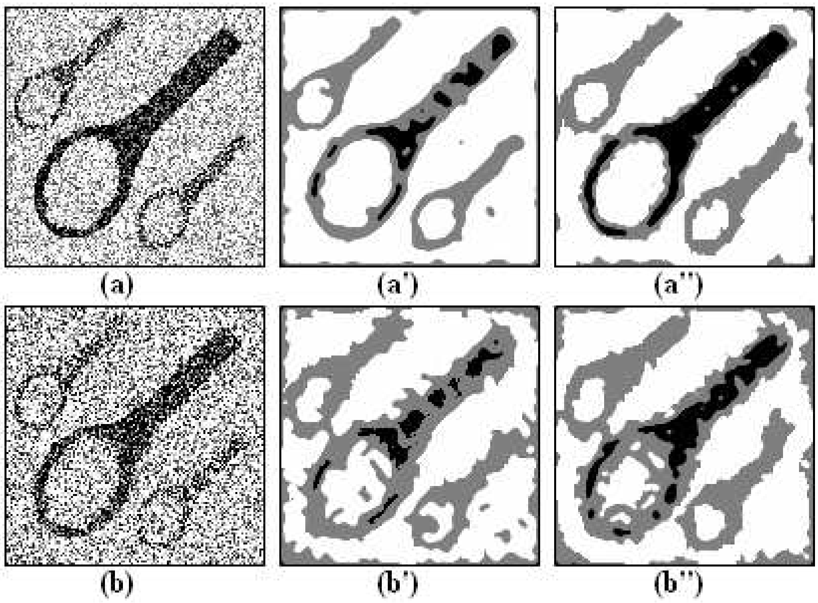

Figure 2: Noisy and retrieved real life gray scale spanner images at $\sigma = 14$ and 16 (a) (b) noisy images (a′)(b′) extracted with fully connected network (a″)(b″) extracted with pruned network

Table 3: Reductions achieved for pure color synthetic image

| | Pruned PSONN processing units | | | | | |
|---|---|---|---|---|---|---|
| | Red component | | Green component | | Blue component | |
| $\sigma$ | *tnoc* | % reduction | *tnoc* | % reduction | *tnoc* | % reduction |
| 8 | 3699 | 97.18% | 3819 | 97.09% | 3751 | 97.14% |
| 10 | 9913 | 92.44% | 9881 | 92.46% | 10136 | 92.27% |
| 12 | 13838 | 89.44% | 13765 | 89.49% | 13764 | 89.49% |
| 14 | 15999 | 87.79% | 15975 | 87.81% | 16026 | 87.77% |
| 16 | 16120 | 87.70% | 16137 | 87.69% | 16141 | 87.69% |

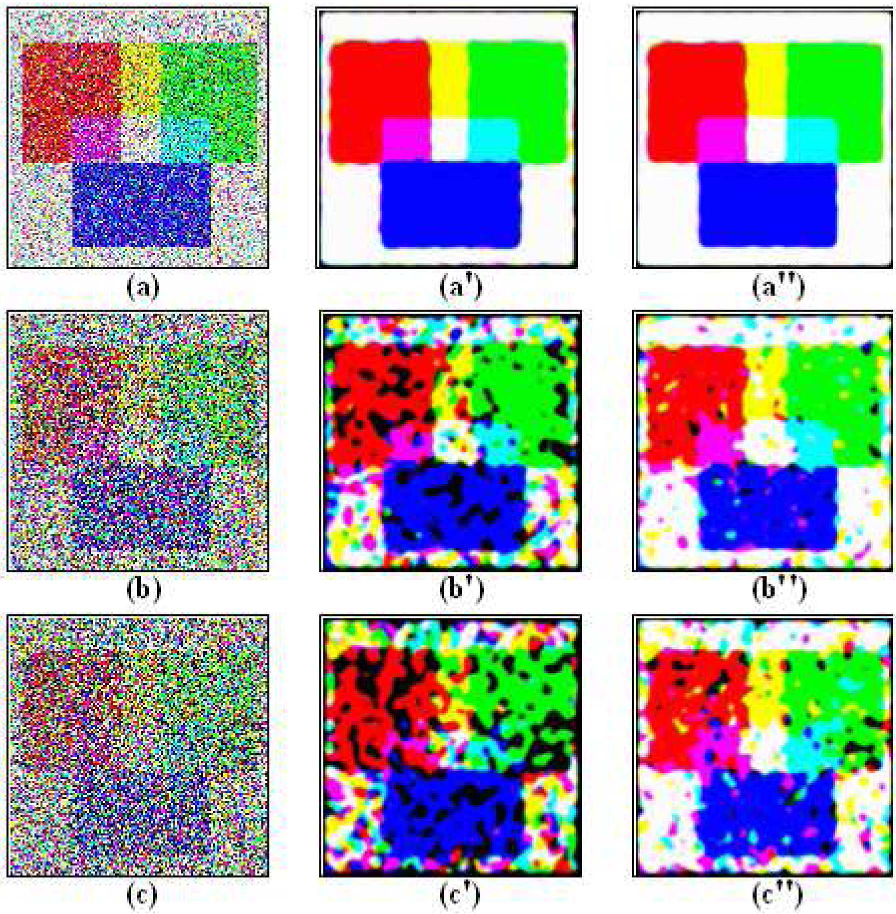


Figure 3: Results of pure color synthetic image segmentation (a)(b)(c) Noisy images at $\sigma = 12$, 14 and 16 (a′)(b′)(c′) Segmented images using fully connected PSONN architecture (a″)(b″)(c″) Segmented images using pruned PSONN architecture

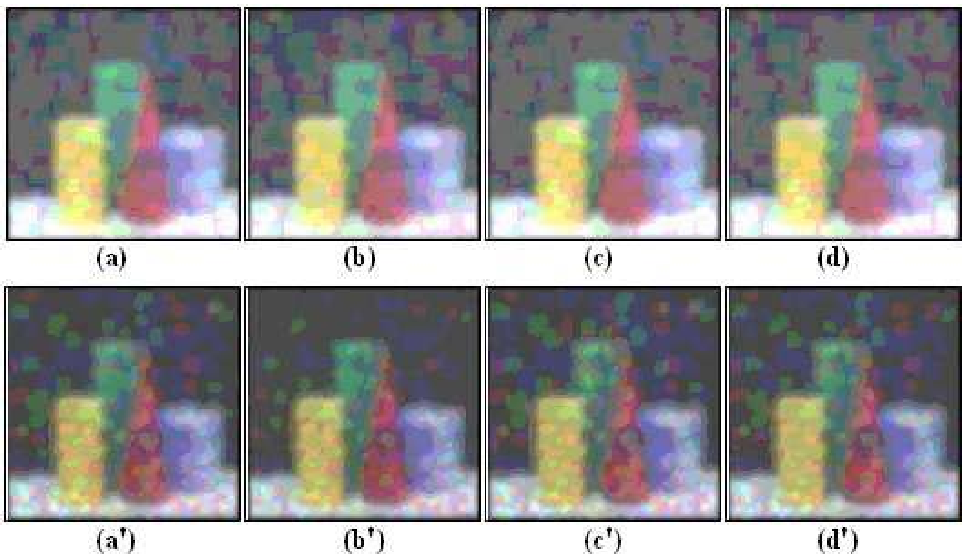


Figure 4: Results of true color cube image segmentation for four sets of $\kappa_\gamma$ parameter (a)(b)(c)(d)(e) Segmented images at $\sigma = 8$ using fully connected PSONN architecture (a′)(b′)(c′)(d′)(e′) Segmented images at $\sigma = 8$ using pruned PSONN architecture

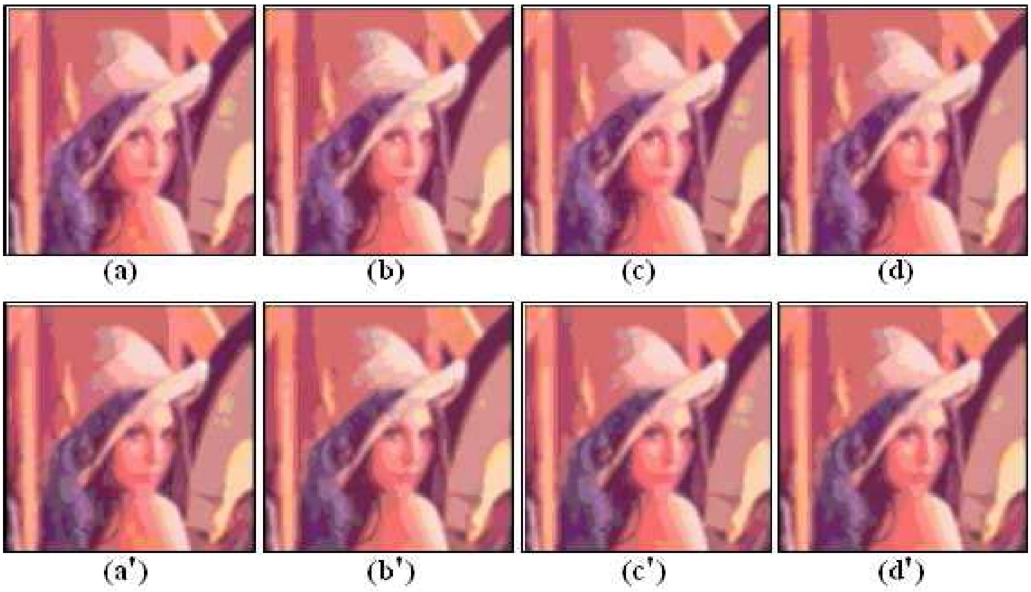


Figure 5: Segmented true color Lena images using sets (a) $s_1$ (b) $s_2$ (c) $s_3$ (d) $s_4$ using fully connected PSONN architecture (a′) $s_1$ (b′) $s_2$ (c′) $s_3$ (d′) $s_4$ using pruned PSONN architecture

Table 4: Reductions achieved for true color cube image

| | Pruned PSONN processing units | | | | | |
|---|---|---|---|---|---|---|
| | Red component | | Green component | | Blue component | |
| $\sigma$ | *tnoc* | % reduction | *tnoc* | % reduction | *tnoc* | % reduction |
| 8 | 12724 | 90.29% | 12832 | 90.21% | 13931 | 89.37% |
| 10 | 15366 | 88.28% | 15413 | 88.24% | 15477 | 88.19% |

Table 5: Reductions achieved for true color Lena image

| Pruned PSONN processing units | | | | | |
|---|---|---|---|---|---|
| Red component | | Green component | | Blue component | |
| *tnoc* | % reduction | *tnoc* | % reduction | *tnoc* | % reduction |
| 4149 | 96.83% | 4530 | 96.54% | 5336 | 95.93% |

Table 7: $\rho$ values for true color cube image and Lena image for different sets of values of $\kappa_\gamma$

| Set | Cube Image | | | | Lena Image | |
|---|---|---|---|---|---|---|
| | $\sigma$=8 | | $\sigma$=10 | | | |
| | Full Network | Pruned Network | Full Network | Pruned Network | Full Network | Pruned Network |
| $s_1$ | 0.8884 | 0.9050 | 0.8184 | 0.8286 | 0.8214 | 0.8282 |
| $s_2$ | 0.8968 | 0.8986 | 0.8065 | 0.8119 | 0.8141 | 0.8287 |
| $s_3$ | 0.8737 | 0.9017 | 0.8082 | 0.8332 | 0.8139 | 0.8253 |
| $s_4$ | 0.8798 | 0.8993 | 0.8119 | 0.8272 | 0.8086 | 0.8245 |

Table 6: *pcc* values for the pure color synthetic image

| $\sigma$ | Full network | Pruned network |
|---|---|---|
| 8 | 97.96 | 98.32 |
| 10 | 92.86 | 96.77 |
| 12 | 92.89 | 96.65 |
| 14 | 54.71 | 85.50 |
| 16 | 29.17 | 76.99 |

noise of zero mean and standard deviation, $\sigma$ = 8, 10, 12, 14 and 16. The noisy images with $\sigma$=12, 14 and 16, are shown in Figure 3 (a)(b)(c). The corresponding images retrieved with the conventional fully connected and pruned architectures, are shown in Figure 3 (a′)(b′)(c′) and (a″)(b″)(c″), respectively.
The performance of the pruning algorithm is also illustrated with the segmentation and retrieval of true color images. The number of target classes are taken to be $K = 8$. $\lambda$=1 decides the slope of the MUSIG activation. Experiments have been conducted with four different sets ($s_1$, $s_2$, $s_3$, $s_4$) of $\kappa_\gamma = \frac{1}{\alpha_\gamma}$ parameter of the multilevel MUSIG activation function.
A true color cube image affected with Gaussian noise (with $\sigma$=8, 10) is used for the object extraction procedure. The images segmented with the fully connected and pruned PSONN architectures for $\sigma = 8$ are shown in Figure 4. Segmentation results are further demonstrated with an 8-class segmentation of a true color Lena image. The segmented output images with the conventional and pruned PSONN architectures, are shown in Figure 5. An analysis of the performance of the pruning algorithm can be made from the reductions achieved in the PSONN interconnections. Tables 3, 4 and 5 show the *tnoc* and reduction level in the *tnoc* for the independent pruned color component (RGB) processing TLSONN units of the PSONN architecture for the test images.
The efficiency of the algorithm can also be adjudged from the quality of the segmented pure and true color images. Table 6 compares the *pcc* values obtained by the fully connected and pruned PSONN architectures.

The qualities of the segmented true color images have been assessed by the standard correlation coefficients ($\rho$) between the outputs obtained and the original images. The $\rho$ values obtained for the true color cube and Lena images are shown in Table 7. From Table 7 it can be surmised that the pruned PSONN architecture outperforms its fully connected counterpart. This is due to the successful prevention of false classifications that is common with the fully connected version.

## 9 Discussions and Conclusion

A fuzzy set-theoretic pruning algorithm for reducing false classification of images through refinement of a neighborhood neural network architecture, is presented. Pruning of irrelevant and redundant network interconnections is carried out by estimation of the fuzzy cardinality estimates of the input image pixel neighborhoods.
Results of applications of the algorithm to the segmentation and retrieval of objects from images are illustrated with a second order neighborhood topology-based three layer self organizing neural network and its parallel version. The efficiency and scalability of the algorithm is justified from its applicability in all of the binary, multilevel and color intensity domains.

## References


[1] Banham, M.R., Katsaggelos, A.K., "Digital Image Restoration," *IEEE Signal Processing Magazine*, V14, N2, pp. 24-41, 97.

[2] Gonzalez, R.C., Woods, R.E., *Digital Image Processing*, Second Edition, Pearson Ed., 2002.

[3] Forrest, B.M., *et al.*, "Neural network models," *Parallel Computing*, V8, pp. 71-83, 88.

[4] Leondes, C.T., *Neural Network Techniques and Applications (Image Processing and Pattern Recognition)*, Academic Press, 1998.

[5] Egmont-Petersen, M., Ridder, D. de, Handels, H., "Image processing using neural networks - A review," *Pattern Recognition*, V35, N10, pp. 2279-2301, 02.

[6] Ghosh, A., Pal, N.R., Pal, S.K., "Self-organization for object extraction using multilayer neural network and fuzziness measures," *IEEE Transactions on Fuzzy Systems*, V1, N1, pp. 54-68, 93.

[7] Reed, R., "Pruning Algorithms - A survey," *IEEE Transactions on Neural Networks*, V4, pp. 740-747, 93.

[8] Zeng, X., Chen, X.-W., "SMO-Based Pruning Methods for Sparse Least Squares Support Vector Machines," *IEEE Transactions on Neural Networks*, V16, N6, pp. 1541-1546, 05.

[9] Kruif, B.J.de, Vries, T.J.de, "Pruning error minimization in least squares support vector machines," *IEEE Transactions on Neural Networks*, V14, N3, pp. 696-702, 03.

[10] Paz, E.C., "Pruning neural networks with distribution estimation algorithms," *GECCO 2003*, pp. 790-800, 03.

[11] Opitz, D.W., Shavlik, J.W., (Cohen W., Hirsh, H. eds.), "Using Genetic Search to Refine Knowledge-Based Neural Networks," *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann, San Fransisco, CA, 94.

[12] Hagiwara, M., "Removal of hidden units and weights for backpropagation networks," *Proceedings of the International Joint Conference on Neural Networks*, V1, pp. 351-354, 93.

[13] Sietsma, J., Dow, R.J.F., "Creating artificial neural networks that generalize," *Neural Networks*, V4, pp. 67-79, 91.

[14] Whitley, D., Bogart, C., "The evolution of connectivity: Pruning neural networks using genetic algorithms," *International Joint Conference on Neural Networks*, V1, pp. 134-137, 90.

[15] White, D., Ligomenides, P., (Mira, J., Cabestany, J., Prieto, A. eds.) "GANNet: A genetic algorithm for optimizing topology and weights in neural network design," *International Workshop on Artificial Neural Networks*, Springer-Verlag, Berlin, Germany, pp. 332-327, 93.

[16] Cottrell, M., Girard, B., Girard, Y., Mangeas, M., Muller, C., "SSM: A statistical stepwise method for weight elimination," *Proceedings of International Conference on Artificial Neural Networks*, V1, pp. 681-684, 94.

[17] Xue, Q., Hu, Y., Tompkins, W.J., "Analysis of the hidden units of backpropagation model," *Proceedings of International Joint Conference on Neural Networks*, V1, pp. 739-742, 90.

[18] Fletcher, L., Katkovnik, V. Steffens, F.E., Engelbrecht, A.P., "Optimizing the number of hidden nodes of a feedforward artificial neural network," *Proceedings of International Joint Conference on Neural Networks*, Anchorage, AK, pp. 1608-1612, 98.

[19] Levin, A.U., Leen, T.K., Moody, J.E., (Cowan, J.D., Tesauro, G, Alspector, J. eds.) "Fast pruning using principal components," *Advances in Neural Information Processing Systems*, Morgan-Kaufmann, San Mateo, CA, V6, pp. 35-42, 94.

[20] Perantonis, S.J., Virvilis, V., "Feature selection using supervised principal components analysis," *Neural Processing Letters*, V10, N3, pp. 243-253, 99.

[21] Steppe, J.M., Bauer, K.W., Rogers, S.K., "Integrated feature and architecture selection," *IEEE Transactions on Neural Networks*, V7, pp. 1007-1014, 96.

[22] Belue L.M., Bauer, K.W., "Determining input features for multilayer perceptrons," *Neurocomput.*, V7, p. 111-121, 95.

[23] Prechelt, L., "Connection pruning with static and adaptive pruning schedules," *Neurocomput.*, V16, N1, pp. 49-61, 97.

[24] Karnin, E.D., "A simple procedure for pruning backpropagation trained neural networks," *IEEE Transactions on Neural Networks*, V1, pp. 239-242, 90.

[25] Zurada, J.M., Malinowski, A., Cloete, I., "Sensitivity analysis for minimization of input data dimension for feedforward neural network," *Proceedings of IEEE International Symposium on Circuits Systems*, London, U.K., 94.

[26] Zurada, J.M., Malinowski, A., Usui, S., "Perturbation method for deleting redundant inputs of perceptron networks," *Neurocomputing*, V14, pp. 177-193, 97.

[27] Engelbrecht, A.P., Cloete, I., "A sensitivity analysis algorithm for pruning feedforward neural networks," *Proceedings of IEEE International Conference on Neural Networks*, V2, pp. 1274-1277, 96.

[28] Engelbrecht, A.P., "A New Pruning Heuristic Based on Variance Analysis of Sensitivity Information," *IEEE Transactions on Neural Networks*, V12, N6, pp. 1386-1399, 01.

[29] Ponnapalli, P.V.S., Ho, K.C., Thomson, M., "A Formal Selection and Pruning Algorithm for Feedforward Artificial Neural Network Optimization," *IEEE Transactions on Neural Networks*, V10, N4, pp. 964-968, 99.

[30] Le Cun, Y. Denker, J.S., Solla, S.A., (Touretzky, D.S. ed.), "Optimal brain damage," *Advances in Neural Information Processing systems*, San Mateo, CA: Morgan Kaufmann, V2, pp. 598-605, 90.

[31] Hassibi, B., Stork, D.G., (Lee Giles, C., Hanson, S.J., Cowan, J.D. eds.), "Second-order derivatives for network pruning: Optimal brain surgeon," *Advances in Neural Information Processing Systems*, V5, pp. 164-171, 93.

[32] Cibas, T. Fogelman Souli, F. Gallinari, P. Raudys, S., "Variable selection with neural networks," *Neurocomput.*, V12, pp. 223-248, 96.

[33] Burrascano, P., "A pruning technique maximizing generalization," *Proceedings of International Joint Conference on Neural Networks*, V1, pp. 347-350, 93.

[34] Zadeh, L.A., "Fuzzy Sets," *Inform. and Control*, V8, pp. 338-353, 65.

[35] Ross, T.J., Ross, T., it Fuzzy Logic With Engineering Applications, McGraw Hill College Div., 1995.

[36] Bhattacharyya, S., Dasgupta, K., "Color object extraction from a noisy background using parallel multi layer self organizing neural networks," *Proceedings of CSI-YITPA(E) 2003*, pp. 32-36, 03.