Color Image Segmentation through Self-Organizing Map

A dissertation submitted in partial fulfillment of the requirements for the M.Tech (Computer Science) degree of the Indian Statistical Institute, Kolkata.

By

Anupam Biswas

under the supervision of

Dr. Sambhunath Biswas **Dr. Sanghamitra Bandophadyay**

Machine Intelligence Unit

INDIAN STATISTICAL INSTITUTE
203, B.T. Road, Kolkata - 700035
21 July, 2004

Indian Statistical Institute,
203, Barrackpore Trunk Road,
Kolkata - 700 035

Certificate Of Approval

This is to certify that this thesis titled "*Color Image Segmentation through Self-Organizing Map*" submitted by Anupam Biswas towards partial fulfillment of requirements for the degree of **M. Tech.** in *Computer Science* at *Indian Statistical Institute, Kolkata* embodies the work done under our supervision.

Sambhunath Biswas 21/7/04

Dr. Sambhunath Biswas,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata - 700 108.

(External Expert)

Dr. Sanghamitra Bandyopadhyay,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata - 700 108.

Acknowledgement

I take pleasure in thanking Dr. Sambhunath Biswas and Dr. Sanghamitra Bandyophadyay for their friendly guidance throughout the dissertation period. Their pleasant and encouraging words have always kept my spirits up.

Finally I take the opportunity to thank my classmates for their encouragement to finish this work.

Anupam Biswas

Abstract

Capability of Kohonen's Self Organizing Map (SOM) in clustering is well established. We have re-examined the capability of SOM in clustering for color image segmentation. Two different methods have been proposed. One method uses non-partitioned input data, and the other method uses partitioned input dataset. Their effectiveness is tested on color and MRI brain images with and without noise. Segmentation results have been viewed through some measures.

Contents

Chapter 1

Introduction

Image segmentation is a fundamental problem in image analysis. The goal of image segmentation is to partition an image into a set of regions which are uniform and homogeneous in some well defined mathematical sense. Ideally, these regions will have pixels strongly correlated with each other. The segmented regions in an image may be viewed as objects and/or background. Different techniques of object/background segmentation for graylevel images are already reported [1. 2, 3, 4]. Entropy based thresholding techniques for object/background segmentation have been found effective and are popular [5. 6]. Some researchers have also viewed the segmentation problem as an unsupervised classification technique [7, 8, 9]. However, these segmentation techniques are mainly developed considering graylevel images.

This dissertation attempts to segment color images, specially color portrait images and magnetic resonance images (MRI) for brain, using Kohonen's Self Organizing Map (SOM) [10]. The justification for the use of SOM in this work is its ability to serve as a clustering tool for high-dimensional data [11, 12]. It is well known that SOM constructs a topology in which the high-dimensional space is mapped onto a two-dimensional regular lattice. As a result, it provides a projection of high-dimensional data on a two-dimensional plane, helping one to visualize. The projection also preserves the topological properties of the input data distribution.

In this dissertation, we have proposed two different methods that generate hierarchical color maps or clusters in the feature space from non-partitioned (input image) or partitioned (Quadtree of input image) input datasets. The color maps are then used to segment an image. The effectiveness of the proposed methods have been demonstrated on a set of noise-free and noissy color images. Quality of segmentation has also been tested through some objective measures as discussed in [13].

To organize the work, we have described the training procedure is described in Chapter 2, while in Chapter 3 the underlying color model has been discussed. The proposed methods are described in Chapter 4. In Chapter 5 we have depicted different image segmentation measures and it is followed by results in Chapter 6.

Chapter 2

Different aspects of SOM

Self Organising Map (SOM) is an unsupervised competitive artificial neural network (ANN) which transforms high dimensional data into its projection on a desired plane. Kohonen's self-organizing map, as shown in Fig. 2.1, consists of a single layer of neurons arranged in one, two or multi-dimensional arrays. Each neuron has as many input connections as the dimensionality of the feature space. The training procedure for each input (data vector) finds a neuron with weights closest to the input and declare that neuron as the winning neuron. The weights of all neurons in the neighbourhood of the winning neuron are subsequently updated by a learning parameter. The neighbourhood radius and the learning parameter are both reduced as the iteration advances. After each iteration, the RMS error is calculated. As the underlying map becomes more and more self-organized, RMS error decreases more and more with each iteration. The training terminates when the RMS error reduces to an acceptable error value. After completion of the training, the formation of the self-organized map (SOM) is complete. This SOM, thus obtained, is responsible for clustering the input dataset.

2.1 Input data

Let N be the number of data points, each having f features. The algorithm separates the data points in some clusters when they are entered as input to the SOM network. Note that, the input to the SOM network are always normalized. This enables a feature to be compared with one another. Below we briefly discuss the normalization.

Normalization : Let the datapoints are $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}$. $\mathbf{x_i}$, where $i \in \{1, 2, \cdots, N\}$, has features $F_1, F_2, \cdots, F_f$. Hence, $\mathbf{x_i} = [x_{i1}, x_{i2}, \cdots, x_{if}]^T$
Consider a particular feature F_j, where $j \in \{1, 2, \cdots, f\}$.
Let

$$a_j = max\{x_{1j}, x_{2j}, \cdots, x_{Nj}\} \tag{2.1}$$

$$b_j = min\{x_{1j}, x_{2j}, \cdots, x_{Nj}\} \tag{2.2}$$

Hence, the i-th datapoint with j-th feature is normalized as.

$$x'_{ij} = \frac{x_{ij} - b_j}{a_j - b_j} \tag{2.3}$$

x_{ij} therefore can be mapped to x'_{ij} using 2.3 where $i \in \{1, 2, \cdots, N\}$ and $j \in \{1, 2, \cdots, f\}$. Effectively equation 2.3 produces the normalized dataset $\mathbf{x'_1}, \mathbf{x'_2}, \cdots, \mathbf{x'_N}$.

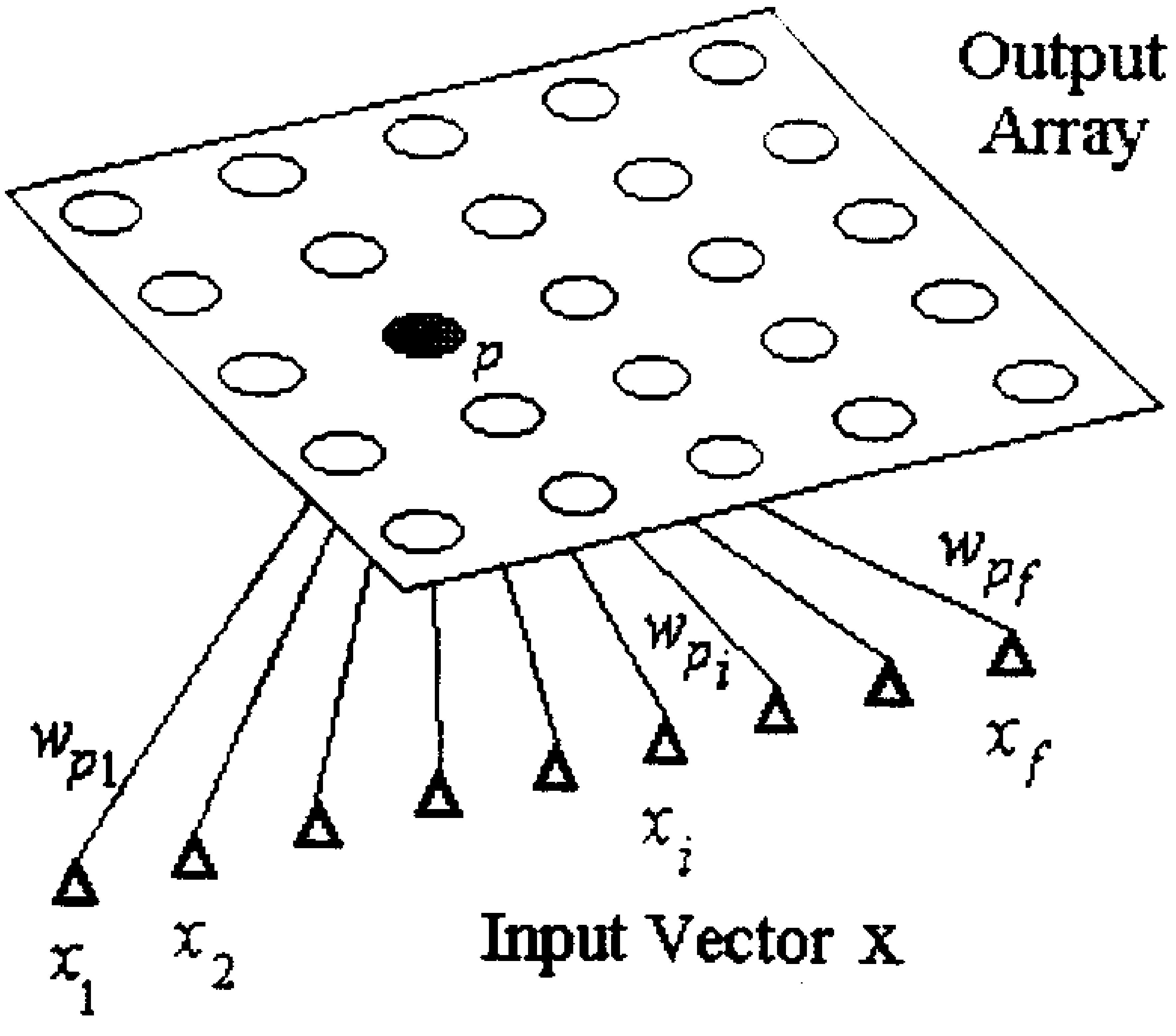

Figure 2.1: Topology of a Self Organising Map with a 5x5 output array,Note that the input vector is connected to all output nodes, not just node p as shown

2.2 Initialization

Each neuron has weight one for each of its attributes. This weight, in general, is initialized through a small random number (between 0 and 1). The initial unordered vectors become ordered after several iterations. The training time, however can be reduced to some extent if one incorporates some domain knowledge to initialize the neural weights without assigning random numbers.

2.3 Learning rule

The SOM is based on competitive learning. In the training period the output neurons with weights $\mathbf{w_p} = [w_{p1}, w_{p2}, \cdots, w_{pf}]^T$ compete with each other to find the best match with the input datapoint $\mathbf{x_i} = [x_{i1}, x_{i2}, \cdots, x_{if}]^T$ where $p \in \{1, 2, \cdots, M\}$ and $i \in \{1, 2, \cdots, N\}$. A widely used measure for the match of $\mathbf{x_i}$ with $\mathbf{w_p}$ is the Euclidean distance between the two, given by:

$$\|\mathbf{x_i} - \mathbf{w_p}\| = \{(x_{i1} - w_{p1})^2 + (x_{i2} - w_{p2})^2 + \cdots + (x_{if} - w_{pf})^2\}^{1/2} \qquad (2.4)$$

The neuron C with weight vector $\mathbf{w_c}$ closest to $\mathbf{x_i}$ is declared the winner, which effectively implies for a winning neuron

$$e(\mathbf{x_i}) = \|\mathbf{x_i} - \mathbf{w_c}\| = \min_{\mathbf{p}}\{\|\mathbf{x_i} - \mathbf{w_p}\|\} \qquad (2.5)$$

$e(\mathbf{x_i})$ can be viewed as the error of the winner from its corresponding input datapoint.

The weight of the winner and also of all the neurons within its neighbourhood N_c are updated by the following learning rule:

$$\mathbf{w_p}(k+1) = \begin{cases} \mathbf{w_p}(k) + \alpha(k)[\mathbf{x_i}(k) - \mathbf{w_p}(k)] & \text{if } p \in N_c(k) \\ \mathbf{w_p} & \text{if } p \notin N_c(k) \end{cases} \tag{2.6}$$

where $\alpha(k)$ is the learning rate at the k^{th} iteration same for both the winning neuron and its neighbours, $(0 < \alpha(k) < 1)$ and $N_c(k)$ is the corresponding neighbourhood. The choice of the neighbourhood, $N_c(k)$, can be found from eqn.(2.8) as described below. The learning rate is a monotonically decreasing function with time. Different learning rates for winner and non-winners can also be chosen. This means, if $\alpha(k)$ is the learning rate for the winning neuron and $\beta(k)$ is that of non-winning neurons then the desired condition is,

$$0 < \beta(k) < \alpha(k) < 1 \tag{2.7}$$

2.4 Neighbourhood region

To select the neighbourhood at the k^{th} iteration let us assume $d_c(k)$ as the neighbourhood radius for the winner node C. $d_c(k)$ is large enough to cover all the neural nodes when the training just begins. In general, $d_c(k)$ is a function that decreases as k increases. The neighbourhood region $N_c(k)$ for the winner node C is then defined as.

$$N_c(k) = \{C\} \cup \{C_{nw} : \|\mathbf{w_c} - \mathbf{w_{c_{nw}}}\| \leq \mathbf{d_c(k)}\} \tag{2.8}$$

2.5 Stopping condition

Learning is a stochastic process and so it is expected that an iterative algorithm that refines more and more a cluster, will find its best possible cluster when the algorithm converges. At each iteration, data points are entered at the SOM input and weights of neurons are updated according to the learning rule given in eqn.(2.6). After each iteration the RMS error is computed. It is the square root of the mean squared error is given by

$$e_{rms}(k) = \sqrt{\frac{e^2(\mathbf{x_1}) + e^2(\mathbf{x_2}) + \cdots + e^2(\mathbf{x_N})}{N}} \tag{2.9}$$

This RMS error, $e_{rms}(k)$ decreases gradually with k. To terminate the iterative algorithm one can choose a small positive number ϵ. The algorithm is stopped when the difference in RMS error $\Delta e_{rms}(k, k+1) \leq \epsilon$ between any two successive iterations.

Chapter 3

Color model

There are several standard color spaces that are widely used in image processing, such as RGB, CMY, HSI, YIQ and $L^*u^*v^*$ [14]. The color space $L^*u^*v^*$ can be calculated from the RGB space through appropriate transformation [15].

In our work we choose $L^*u^*v^*$ color space for the clustering purpose. In $L^*u^*v^*$ color space, u^* and v^* represent color chromaticity and L^* the intensity. $L^*u^*v^*$ color space is used in color clustering because of the simplicity of this model, e.g., the difference between two colors in this model can be simply measured by their Euclidean distance [12].

Initially, the original color image data are converted into CIE XYZ tristimulus values. Subsequently, they are converted into L^*, u^*, v^* values.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.618 & 0.177 & 0.205 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.056 & 0.944 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

Transformation for L^*, u^*, v^* are as follows:

$$L^* = \begin{cases} 903.3\left(\frac{Y}{Y_0}\right) & \text{if } \frac{Y}{Y_0} < 0.008856 \\ 116\left(\frac{Y}{Y_0}\right)^{1/3} - 16 & \text{otherwise} \end{cases} \tag{3.2}$$

$$u^* = 13L^*(u' - u'_0) \tag{3.3}$$

$$v^* = 13L^*(v' - v'_0) \tag{3.4}$$

where

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{3.5}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{3.6}$$

$$u'_0 = \frac{4X_0}{X_0 + 15Y_0 + 3Z_0} \tag{3.7}$$

$$v'_0 = \frac{9Y_0}{X_0 + 15Y_0 + 3Z_0} \tag{3.8}$$

and X_0=96.422, Y_0=100.00 , Z_0=85.521.

Chapter 4

Image segmentation : Proposed Methods

In this dissertation, we propose two different methods for image segmentation using SOM. Method 1 uses all the pixels of the image to input to the SOM, whereas for the Method 2, the image is divided in four quads or sub-images and each of these sub-images is fed to the SOM. The methods are discussed below in detail.

4.1 Method 1 (Non-partitioned Input Data)

Method 1 considers the non-partitioned single image as the input to the SOM. In other words, the entire image is first normalized and fed to the input of the SOM.In this method all the datapoints corresponding to all pixels of the input image are fed to the SOM network.

4.1.1 Choice of ANN

In the Kohonen's self-organizing feature map the neurons or the nodes are arranged in grid-fashion as shown in Fig. 2.1. For a grid of size $m \times n$, the number of nodes, $M = m * n$. Hence, in the worst case, the maximum number of clusters is M.

It should be noted that the final segmentation of the color image, as expected, would not have large color variation. This fact dictates M to be small. In both the methods the grid size at the topmost level of the network is chosen 4×4, i.e. $M = 16$.

Due to the large intensity variation at the input, it is found that a hierarchically reduced SOM is effective in clustering. Hence, instead of deciding a single grid of size 4×4 at its final stage we reduce the grid size in hierarchy to come to the final stage, as

- $16 \times 16 \rightarrow 8 \times 8 \rightarrow 4 \times 4$ (3 stage network)
- $16 \times 16 \rightarrow 4 \times 4$ (2 stage network)
- $8 \times 8 \rightarrow 4 \times 4$ (2 stage network)
- 4×4 (single stage network)

For each stage, training is applied on the input image and it produces an output image which has maximum number of colors/classes as the number of nodes of the SOM grid. The output image is applied as input to the next stage.

4.1.2 Choice of features

In the present problem of image segmentation the input data set is the intensity values of the pixels. For a color image each data point has three intensity values which are L, u^* and v^* of the pixel. For the MRI brain images, they are intensity values of a pixel along three bands, namely pd (proton density), t1 (longitudinal relaxation time) and t2(transverse relaxation time). Thus, we have three different features.

Three more features we have chosen to guide the SOM. These are the cluster- centres generated by the K-means clustering algorithm. K-means clusters control the SOM when their centres are embedded in the SOM updating rules. Each centre provides three features and hence we have six features altogether. K-means algorithm, therefore, can be viewed as a preprocessing tool in our segmentation work. To guide or control the SOM by the K-means clusters, we examine the datapoint $\mathbf{x}_i$; if $\mathbf{x}_i$ lies in cluster, say C_j, then cluster-center of C_j is taken as the feature for the datapoint $\mathbf{x}_i$.

4.1.3 Initialization

Let us assume,
M = number of neurons of SOM and
K = number of clusters of K-means algorithm
The neuron-weights are initialized by the first M cluster-centers if $M < K$. Otherwise, the first K neurons are initialized by the K cluster-centers, and the rest are initialized randomly.

4.1.4 Learning factor

Let us assume SOM has a topological grid of dimension $m \times n$. In the present work the learning rate at k^{th} iteration is chosen as

$$\alpha(k) = \alpha_0 \times (1 - l(k) \cdot \alpha_L - l^2(k) \cdot \beta_L) \tag{4.1}$$

where,
$l(k) = level(k)/MAXLEVEL$
$level(k) = k/(Iteration\_at\_each\_Level)$ [In our problem iteration at each level = 8]
$MAXLEVEL = 0.75{*}L$
$L = max(m, n)$
α_0 is the initial learning rate
α_L and β_L are two constants
In the present work we have considered
$\alpha_0 = 0.01, \alpha_L = 0.6, \beta_L = 0.001$ (For 1st stage)
$\alpha_0 = 0.001, \alpha_L = 0.2, \beta_L = 0.0001$ (For next stages)

4.1.5 Neighbourhood radius of winner

The winner neuron C is changed with k as

$$d_c(k) = Max\_Rad - level(k) \tag{4.2}$$

where $Max\_Rad = (L + 1)/2$

4.1.6 Stopping condition

In the present work, we have chosen $\epsilon = 10^{-5}$ to terminate the training as described in sec. 2.5

4.1.7 Output Image

The output image of a stage is the input to the next stage. The SOM divides the image dataset in clusters. In the output image the intensity values of all pixels in a cluster are replaced by their average intensity value.

4.1.8 Merging

For better segmentation merging small segment is found to be effective. This raises two issues: which regions are to be merged and where are to be merged. In order to do this the following steps are performed on the image obtained from SOM network. It should be noted that this image containes only a few colors.

1. The isolated pixels are eliminated with the help of a gliding window. If an isolated pixel is found in the gliding window, its associated class will be substituted by the most frequent class occurred in the gliding window. The gliding window moves across the whole image from left to right, and from top to bottom. This process is repeated until no isolated pixels are left in the image.

2. The small segments of 2 and 3 pixels which are totally sorrounded by another large segment are merged in that large segment.

3. If two neighboring segments are very close in the color space, they are merged. To measure the closeness, the Euclidean distance of the colors of those segments should be less than a predefined positive value (10 in our case) in color space. This process is repeated for the entire image.

4. Finally, the segments having less than 4 pixels and not totally sorrounded by another segment are merged with its largest neighbour.

The output image after segment-merging is the final segmented image of the given input image.

4.2 Method 2 : Partitioned Input Data

For method 2, the input image is divided into its four quads or sub-images. All the sub-images are applied to SOM network and their corresponding output sub-images are obtained. These four output sub-images are merged to form a single image, and then this image is again fed to the SOM network. The output of this SOM network is the result which is then applied for segment-merging. Thus in this method a two pass-SOM is applied. In the first pass all the sub-images are trained, and in the second pass the whole image is trained.

The method is discussed in detail below:

4.2.1 Choice of ANN

In method 2 the final stage of SOM network is also 4 × 4. So, there may be atmost 16 different colors in the output image from SOM network. Since there are four sub-images, so the merged image of the four output sub-images contains atmost 64 different colors. Hence. the first stage of the second pass should contain same or less than 64 nodes.

In the first pass the original image is fed to the SOM network. So the sequence of stages are exactly the same as in the method 1.

In the first pass, four different networks are considered:

- $16 \times 16 \rightarrow 8 \times 8 \rightarrow 4 \times 4$ (3 stage network)
- $16 \times 16 \rightarrow 4 \times 4$ (2 stage network)
- $8 \times 8 \rightarrow 4 \times 4$ (2 stage network)
- 4×4 (single stage network)

whereas, in the second pass two networks are considered:

- $8 \times 8 \rightarrow 4 \times 4$ (2 stage network)
- 4×4 (single stage network)

The algorithm, therefore provides two different cases. The first case considers 2 stage network in the second pass, while the second case considers the single stage network in the second pass.

4.2.2 Choice of other parameters

The choice of features. initialization, learning factor, neighbourhood radius of winner neuron and stopping condition, all are exactly the same as those in Method 1.

4.2.3 Segment Merging

Merging is also the same as in method 1.

Chapter 5

Image comparison measures

In order to evaluate the quality of segmentation produced by the proposed algorithms and to compare them we are using following measures,

5.1 Number of segments

After segment-merging we get finally the segmented image. This is our desired final output. The number of segments present in this image is an important measure. It should not be very large. We find number of segments of the segmented image in the following way,

- From the top-left to bottom-right we scan the image. Initially all the pixels are labeled 0.
- If two neighbouring pixels have the same color, they are given same label, which is the segment number. The segment is increased using 8-N neighbourhood. All the pixels in this segment is labeled the same segment-number.
- Segment-number is increased after finding all pixels present in a segment.
- From the next unlabeled pixel the same procedure is repeated until there is no unlabeled pixel.
- Highest segment-number plus one is the total number of segments.

5.2 Contrast

Another requirement for a good segmentation is that the contrast at inter-region boundaries must be very high compared to that for the interior points. This criterion immediately suggests that the average contrast. i.e., contrast per pixel, say $\bar{K_b}$, of all inter-region boundary points in all sub-images should be high compared to that,say $\bar{K_\Omega}$ over all points enclosed within the boundaries. Therefore,

$$\bar{K_b} \gg \bar{K_\Omega}$$

The contrast c_{ij}, at the pixel position (i, j) can be computed as

$$c_{ij} = \frac{B - B_{ij}}{B} = \frac{\Delta B}{B} \tag{5.1}$$

where B is the immediate surrouding luminance of the (i,j)th pixel with intensity B_{ij}. For the color image we consider intensity as the average of *red.green* and *blue* values.

Let SB be the set of all boundary points and SI be the set of all interior points ($SB \cup SI = F =$ Total image.$SB \cup SI = null\ set$). Contrast to all boundary points. K_b and that of interior points, K_Ω are, therefore,

$$K_b = \sum_{(i,j)\in SB} c_{ij} \tag{5.2}$$

$$K_\Omega = \sum_{(i,j)\in SI} c_{ij} \tag{5.3}$$

Note that K_Ω is an indicant of homogeneity within-regions - lower the value of K_Ω, higher is the homogeneity. The contrast per pixel, $\bar{K_b}$, of all inter-region boundary points and that over all points enclosed within the boundaries,$\bar{K_\Omega}$ can be obtained dividing K_b by the number of boundary points and K_Ω by the number of interior points.

Chapter 6

Results and Discussions

6.1 Results

The input images consist of color pepper image of size 256 × 256 as shown in Fig. 6.1.(a) and two MRI brain images, each of size 181 × 217. as shown in Fig. 6.2.(d) and Fig. 6.3.(d). The rgb values in 6.2.(d) corresponds to different weights (namely, pd, t1 and t2) as shown in Fig. 6.2.(a)-(c). Similarly, the rgb values in Fig. 6.3.(d) reflect the pd, t1, t2 weights of the corresponding gray weights as shown in Fig. 6.3.(a)-(c).

The algorithms for two different methods in different cases, as briefly furnished below, consider the input images and provide their own results.

- Method 1 : Uses non-partitioned input data and is based on a single pass
- Method 2 : Uses partitioned input data and is based on two passes giving rise two cases
  - Case 1 : Uses 2 stage-network in the second pass
  - Case 2 : Uses single stage-network in the second pass

In the Method 1, we consider four different networks as described in Sec. 4.1.1. The first pass of the Method 2 uses exactly the same four different networks of the Method 1. These four different networks are listed below for convenience:

- Network 1 : $16 \times 16 \rightarrow 8 \times 8 \rightarrow 4 \times 4$
- Network 2 : $16 \times 16 \rightarrow 4 \times 4$
- Network 3 : $8 \times 8 \rightarrow 4 \times 4$
- Network 4 : 4×4

In order to compare the results of the two methods and to evaluate the quality of segmentation produced by each of these we have examined in Tables 6.1-6.3 the objective measures as discussed in Chapter 5. It is clear from the tables that for all the input images Case 1 of Method 2 provides the best result for Network 3, i.e., the network $8 \times 8 \rightarrow 4 \times 4$ in both the first pass, and in the second pass. This is because in this case for all the input images, the number of segmented regions is found to be minimum with high between-region boundary contrast $(\bar{K}_b)$ and low within-region contrast $(\bar{K}_\Omega)$. This case is also found to produce the best results for noissy images.

6.2 Noise addition

To test the algorithm for noissy images, we have considered Gaussian noise with variance $\sigma = 1$ and 2 in three different noissy input images. Fig. 6.1.(n) and Fig. 6.1.(o) are noissy pepper images for $\sigma = 1$ and $\sigma = 2$ respectively. Fig. 6.1.(p) and Fig. 6.1.(q) are their corresponding output images. Similarly, the results for MRI brain image 1 and brain image 2 are shown in Fig. 6.4. Table 6.4 provides with segmentation results in the best case for different images for two different noise. It is seen from the Table 6.4 that the number of segmented regions, as expected, increses with noise. However. the between-region contrast and the within-region contrast remain almost the same. This establishes, therefore, the capability of the algorithm to handle noissy images.

6.3 Output image

In Fig. 6.1 results of *pepper* image are given. Fig. 6.1.(b)-(e) are outputs by Method 1 using Network 1, 2, 3 and 4 respectively. Fig. 6.1.(f)-(i) are outputs for Case 1 of Method 2 using Network 1, 2, 3 and 4 in the first pass, while Fig. 6.1.(j)-(m) are outputs for Case 2 of Method 2 using Network 1, 2, 3 and 4 in the first pass. Fig. 6.1.(n)-(q) are related with noissy pepper images and their outputs as described in Sec. 6.2.

In Fig. 6.2 results of MRI brain image 1 are shown. Fig. 6.2.(a)-(c) are pd, t1 and t2-weighted images respectively. Fig. 6.2.(d) is the color version of Fig. 6.2.(a)-(c) where pd-t1-t2 are taken as r-g-b. In our algorithms Fig. 6.2.(d) is the input brain image 1. 6.2.(e)-(h) are outputs by Method 1 and Fig. 6.2.(i)-(p) are outputs by Method 2.

In the same arrangement like Fig. 6.2, in Fig. 6.3 input and result images of MRI brain image 2 are shown. Fig. 6.4 shows the results when additive gaussian noise are added with the MRI brain images. In Fig. 6.5, some more segmentation results are shown which we have obtained using our best algorithm on some color images.

Table 6.1: *pepper image*

	Method 1			Method 2 : Case 1			Method 2 : Case 2		
	Segments	K_b	K_Ω	Segments	K_b	K_Ω	Segments	K_b	K_Ω
Network 1	938	0.146	0.0071	691	0.15	0.0067	585	0.15	0.006
Network 2	993	0.16	0.007	657	0.15	0.006	640	0.156	0.0065
Network 3	745	0.164	0.0062	520	0.17	0.0059	685	0.155	0.0064
Network 4	761	0.134	0.0052	665	0.15	0.0064	656	0.167	0.0064

Table 6.2: *MRI brain image 1*

	Method 1			Method 2 : Case 1			Method 2 : Case 2		
	Segments	K_b	K_Ω	Segments	K_b	K_Ω	Segments	K_b	K_Ω
Network 1	566	0.161	0.01	600	0.179	0.0107	460	0.183	0.0104
Network 2	657	0.163	0.0091	455	0.181	0.0094	540	0.17	0.0097
Network 3	669	0.16	0.0105	429	0.177	0.0085	433	0.176	0.0086
Network 4	621	0.169	0.0106	449	0.186	0.0095	454	0.186	0.0094

Table 6.3: *MRI brain image 2*

	Method 1			Method 2 : Case 1			Method 2 : Case 2		
	Segments	K_b	K_Ω	Segments	K_b	K_Ω	Segments	K_b	K_Ω
Network 1	769	0.16	0.013	753	0.131	0.0123	581	0.165	0.0091
Network 2	727	0.16	0.0127	659	0.16	0.011	689	0.141	0.0133
Network 3	713	0.142	0.0112	553	0.16	0.0111	625	0.184	0.0124
Network 4	563	0.157	0.012	624	0.146	0.0125	609	0.153	0.0113

Table 6.4: Results for noissy images (Case 1 of Method 2. Network 3)

Input images	$\sigma = 1$			$\sigma = 2$		
	Segments	K_b	K_Ω	Segments	K_b	K_Ω
pepper image	663	0.194	0.0096	797	0.203	0.0109
brain image 1	580	0.205	0.0138	654	0.19	0.0153
brain image 2	500	0.231	0.0174	763	0.210	0.0169

fig.(a)

fig.(b)

fig.(c)

fig.(d)

fig.(e)

fig.(f)

fig.(g)

fig.(h)

fig.(i)

fig.(j)

fig.(k)

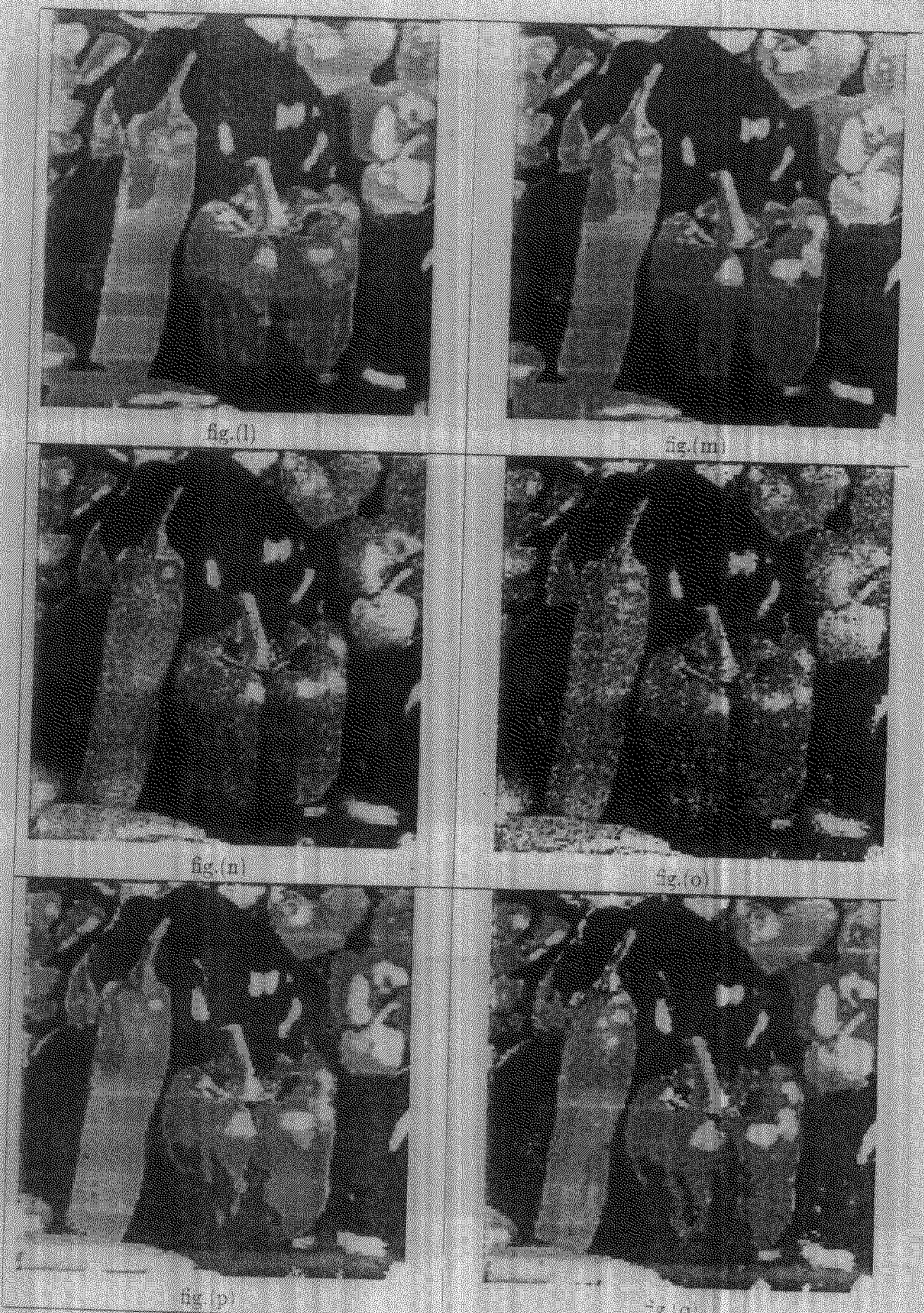

fig.(l)

fig.(m)

fig.(n)

fig.(o)

fig.(p)

fig.(q)

Fig. 6.1: (a) *Pepper* image. (b)-(e) Output of Method 1. (f)-(m) Output of Method 2. (n)-(o) Noisy *pepper* images. (p) Output of (n). (q) Output of (o).

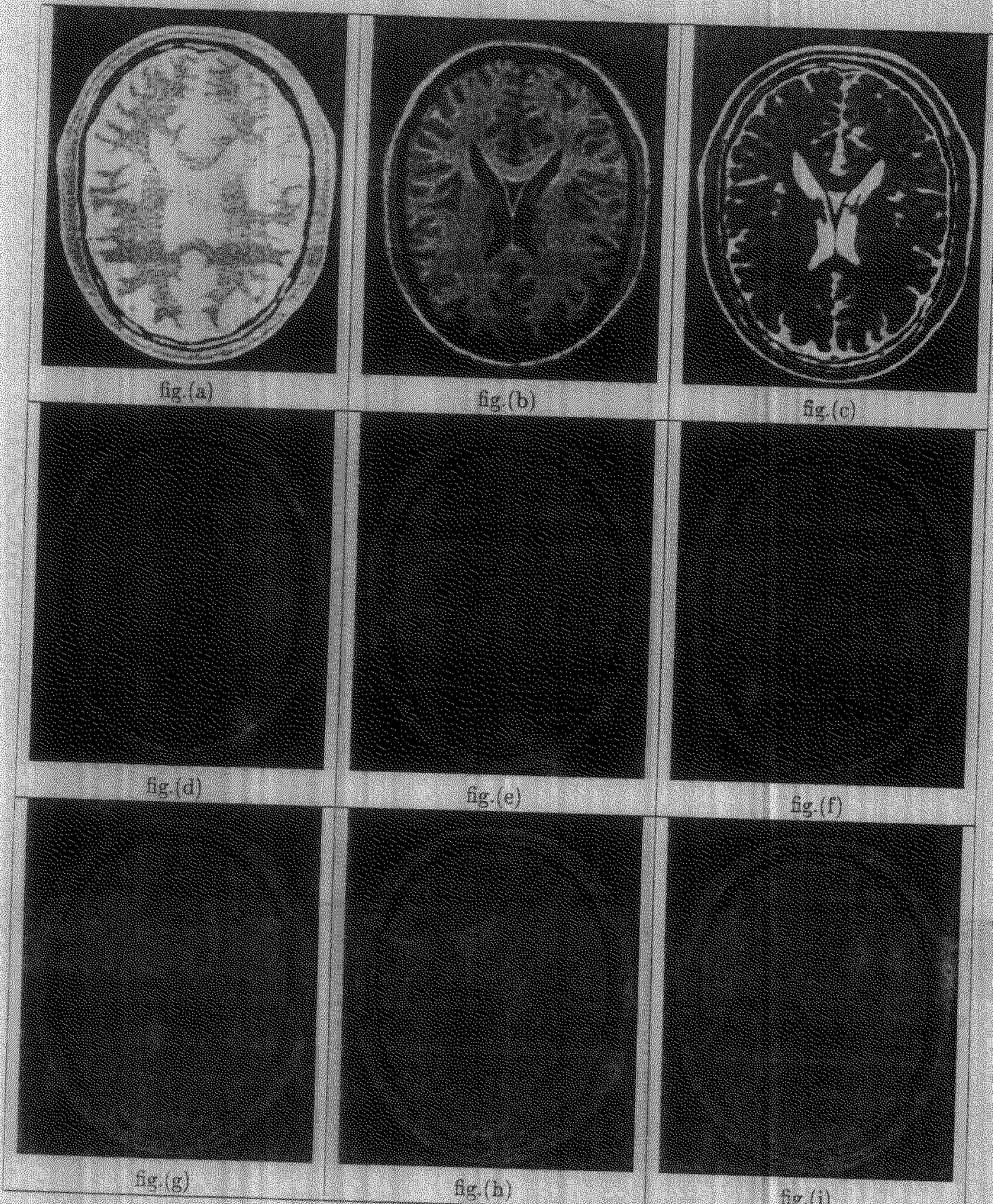

fig.(a) fig.(b) fig.(c)

fig.(d) fig.(e) fig.(f)

fig.(g) fig.(h) fig.(i)

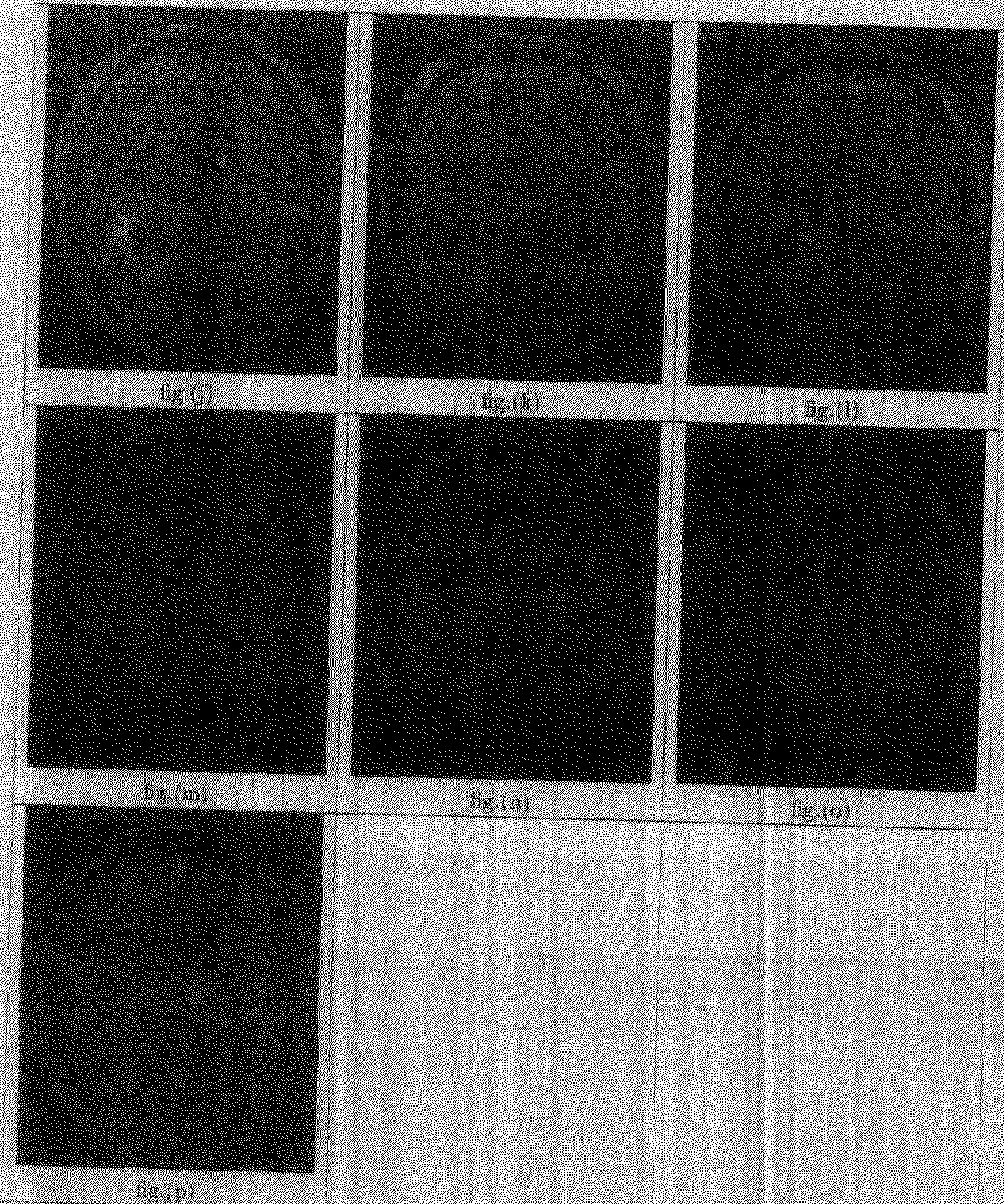

fig.(j)

fig.(k)

fig.(l)

fig.(m)

fig.(n)

fig.(o)

fig.(p)

Fig. 6.2: (a)-(c) MRI image *brain*1 (pd,t1,t2-weighted), (d) Color version of (a)-(c) taking pd-t1-t2 as r-g-b, (e)-(h) Output of Method 1, (i)-(p) Output of Method 2

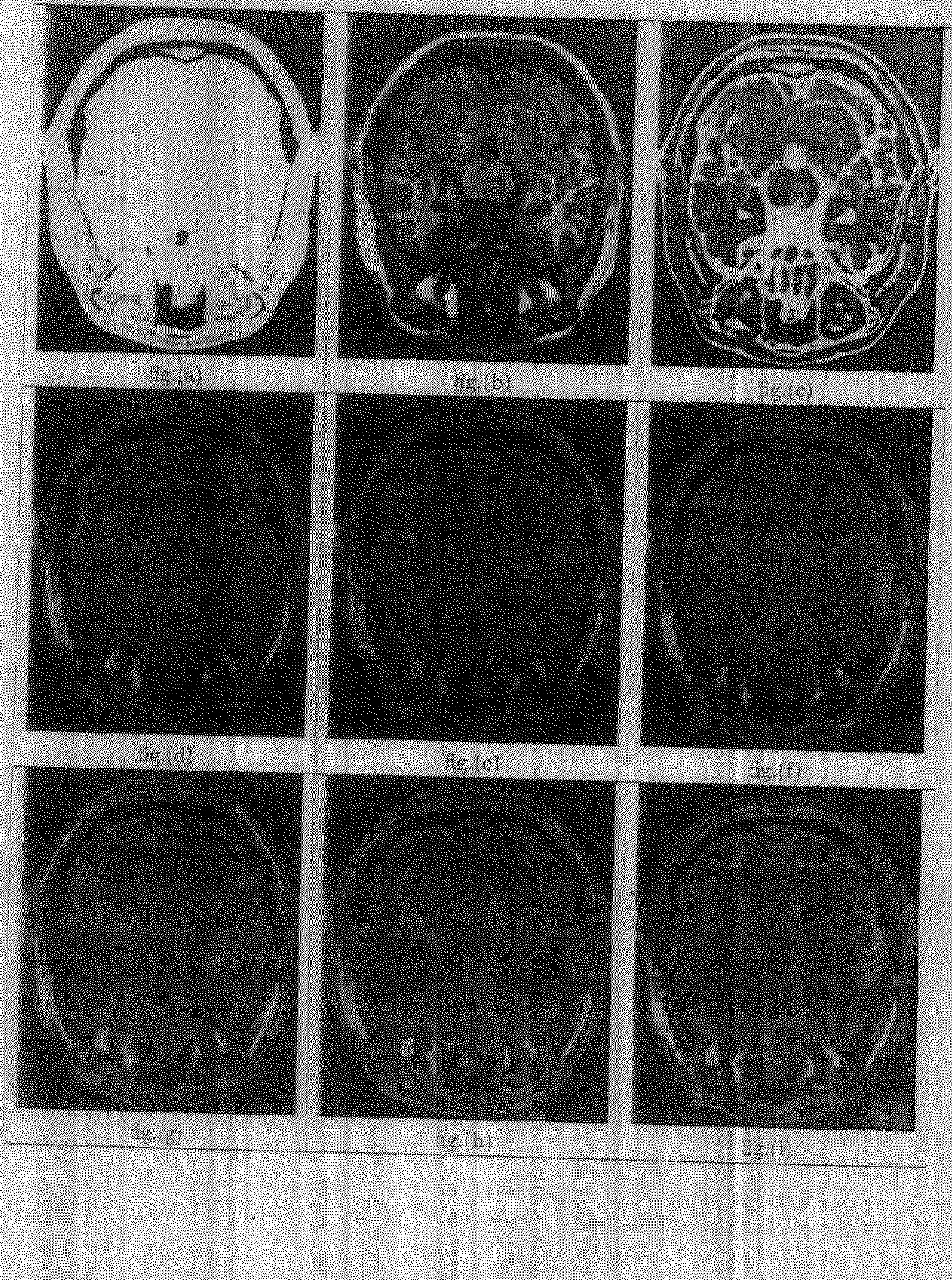

fig.(a) fig.(b) fig.(c)

fig.(d) fig.(e) fig.(f)

fig.(g) fig.(h) fig.(i)

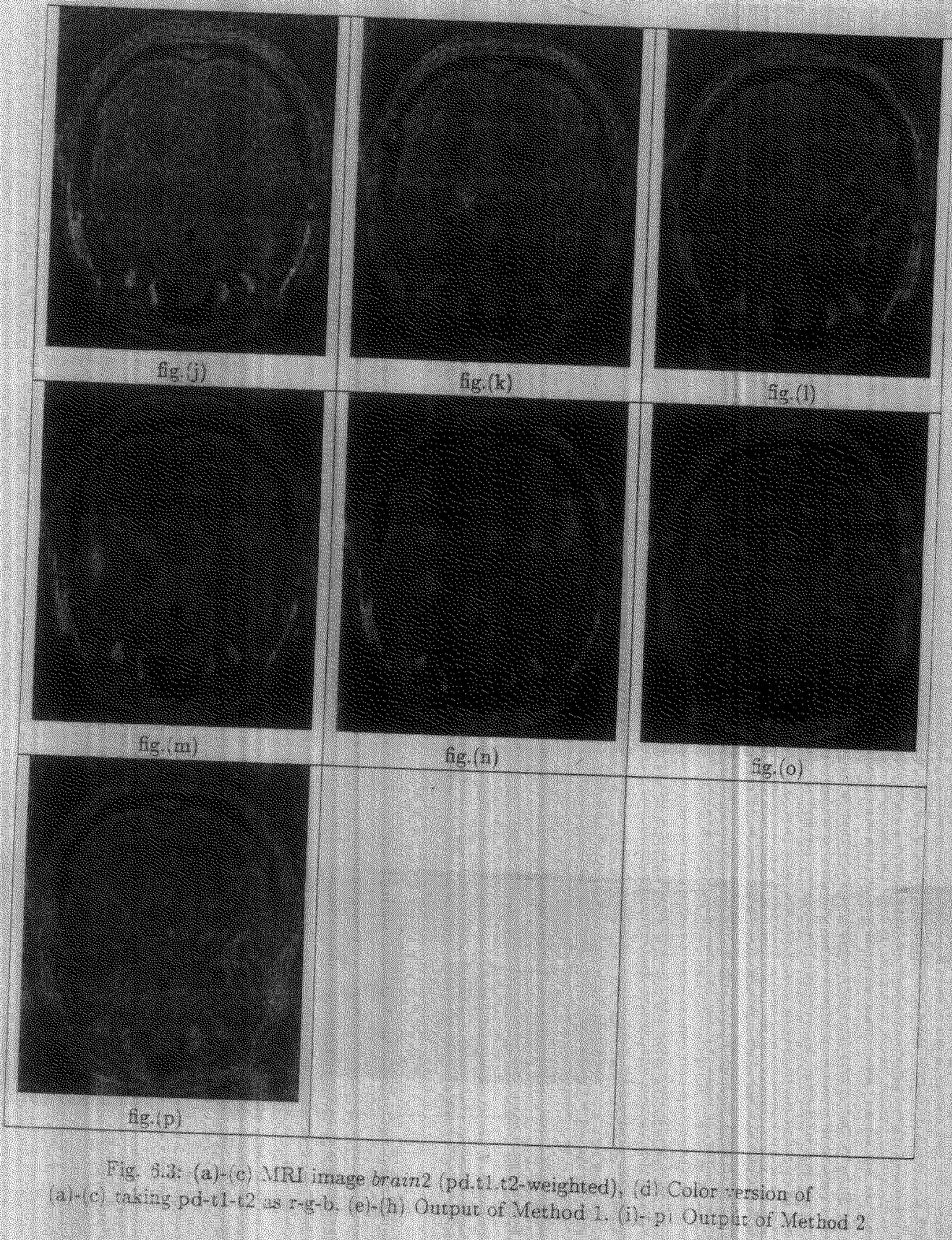

fig.(j) fig.(k) fig.(l)

fig.(m) fig.(n) fig.(o)

fig.(p)

Fig. 6.3: (a)-(c) MRI image *brain*2 (pd.t1.t2-weighted), (d) Color version of (a)-(c) taking pd-t1-t2 as r-g-b. (e)-(h) Output of Method 1. (i)-(p) Output of Method 2

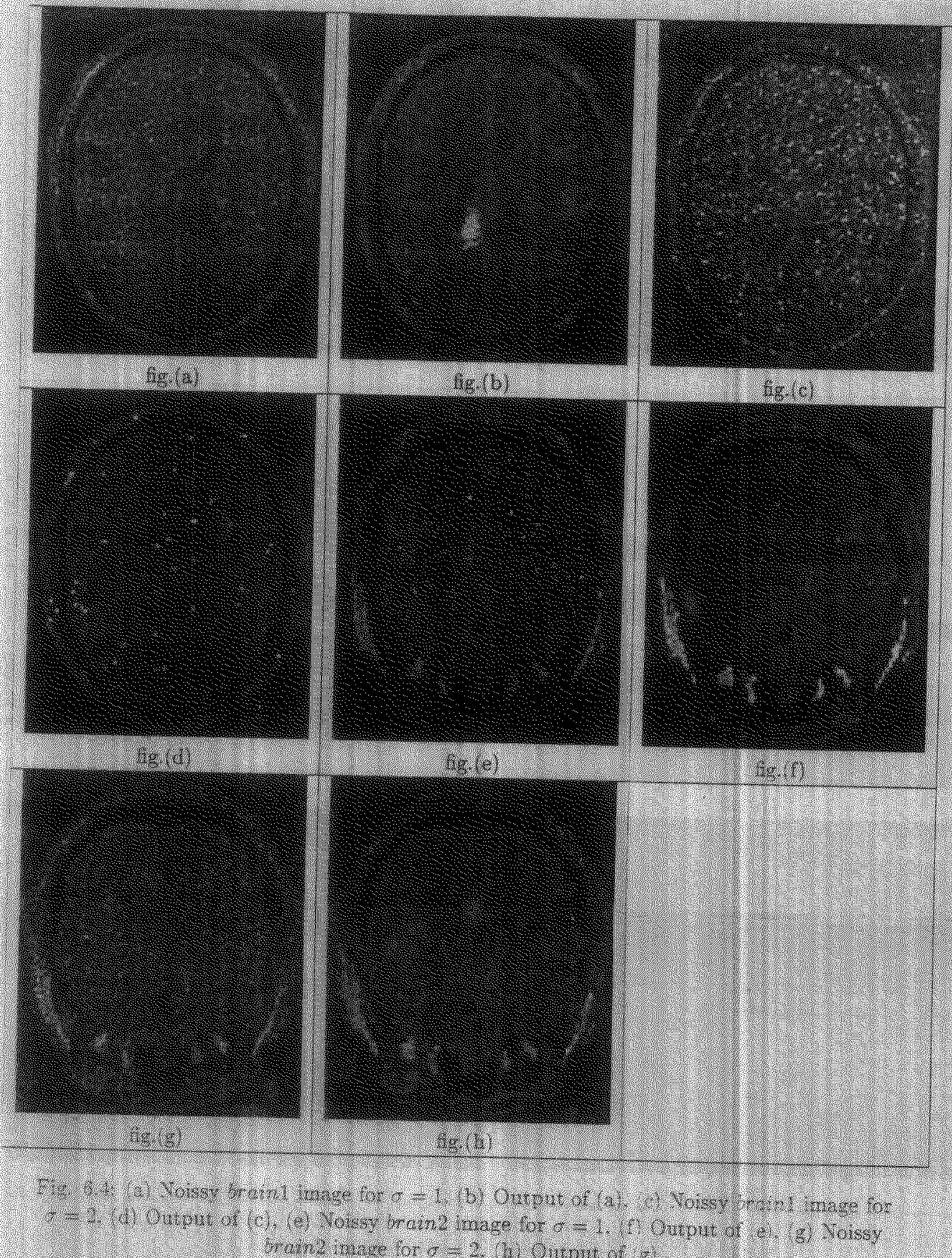

fig.(a)

fig.(b)

fig.(c)

fig.(d)

fig.(e)

fig.(f)

fig.(g)

fig.(h)

Fig. 6.4: (a) Noissy *brain*1 image for $\sigma = 1$. (b) Output of (a). (c) Noissy *brain*1 image for $\sigma = 2$. (d) Output of (c), (e) Noissy *brain*2 image for $\sigma = 1$. (f) Output of (e). (g) Noissy *brain*2 image for $\sigma = 2$. (h) Output of (g).

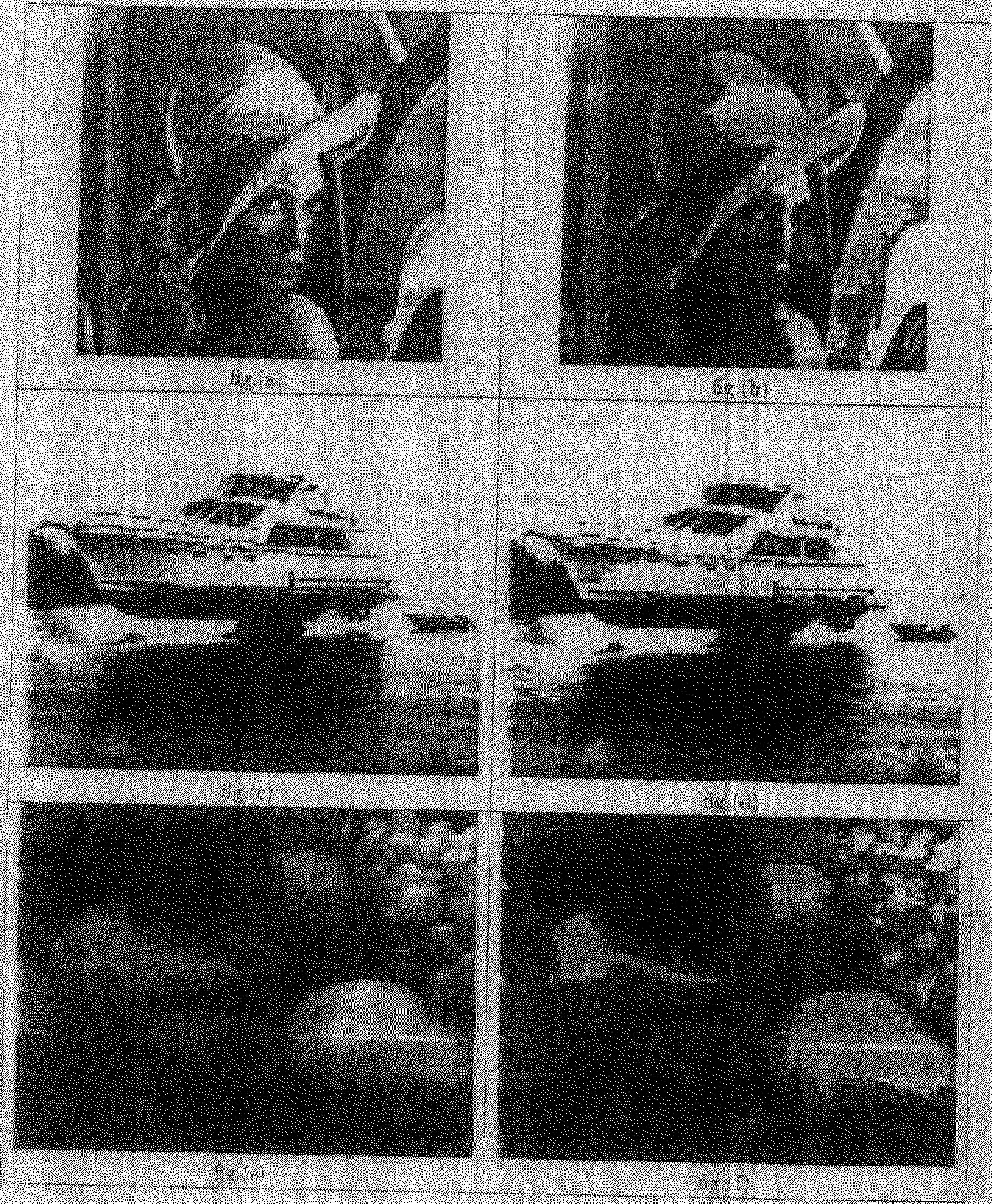

fig.(a)

fig.(b)

fig.(c)

fig.(d)

fig.(e)

fig.(f)

Fig. 3.5: (a) Lenna image. (b) Output of (a). (c) Ship image. (d) Output of (c). (e) Fruit image. (f) Output of (e).

Chapter 7

Conclusions

A multiple-stage SOM based neural network has been used in the work for segmenting color portrait and MRI brain images. Two different methods have been devised to produce the self-organizing map which provides clustering for input images. Segmentation subsequently uses this color map. K-means clustering algorithm for initialization effectively does not provide much gain in time but it helps to generate cluster centers which provides another useful feature for segmentation.

The segmented region shows good contrast of the between-region boundary and good homogeneity through the within-region contrast. It is seen that the partitioned input (Method 2) provides better segmentation than the non-partitioned input (Method 1). This is due to the benefit obtained from both the local and global behaviour of the input. The result supports the same fact when noissy input images are used. Thus, the method with partitioned input seems to be robust to some extent over that with non-partitioned with.

Bibliography

[1] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison and Wesley, 2 edition, 1993.

[2] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[3] N. R. Pal and S. K. Pal. Segmentation using contrast and homogeneity measures. *Pattern Recognition Letters*, 5:293–304, 1987.

[4] B. Chanda, B. B. Chaudhuri, and D. Dutta Majumder. On image enhancement and threshold selection using the gray level co-occurrence matrix. *Pattern Recognition Letters*, 3:243–251, 1985.

[5] N. R. Pal and S. K. Pal. Object-background segmentation using new definitions of entropy. *IEEE Proc.*, 136:284–295, 1989.

[6] N. R. Pal and S. K. Pal. Entropy thresholding. *Signal Process*. 16:97–108, 1989.

[7] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao. Segmentation of color images using a two-stage self-organizing network. *Image and Vision Computing*, 20:279–289, 2002.

[8] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao. Segmentation of color images using a two-stage self-organizing network. *Image and Vision Computing*, 20:279–289, 2002.

[9] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao. Segmentation of color images using a two-stage self-organizing network. *Image and Vision Computing*, 20:279–289, 2002.

[10] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 2nd edition. 1997.

[11] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao. Segmentation of color images using a two-stage self-organizing network. *Image and Vision Computing*, 20:279–289, 2002.

[12] Y. Jiang, K. J. Chen, and Z. H. Zhou. Som based image segmentation. *Artificial Intelligence 2639*, pages 640–643, 2003.

[13] S. Biswas and N. R. Pal. On hierarchical segmentation for image compression. *Pattern Recognition Letters*, 21:131–144, 2000.

[14] Y. I. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*. 13:222–241, 1980.

[15] K. N. Plataniotis and A. N. Venetsanopoulos. *Color Image Processing and Application*. Springer, 2000.