CHAPTER 5
MODIFIED ARTIFICIAL NEURAL NETWORKS FOR IMAGE SEGMENTATION

5.1 INTRODUCTION

Image segmentation is the methodology in which the specified input image is partitioned into different clusters based on some similarity measures. In brain image analysis, segmentation is usually performed after the classification process. The abnormal portion from the classified input brain image is extracted using this segmentation process which is useful to perform the volumetric analysis. Though several techniques have been widely used for this segmentation process, ANN has gained a significant position in the medical image segmentation scenario. Even though ANN is claimed to be more accurate and quick, it is really hard to find both these performance measures in the same network. While some ANN is accurate, several other networks are computationally fast. This problem is solved in this research by proposing a modified neural network namely Modified BPN (MBPN) which is comparatively better than the conventional BPN in terms of segmentation efficiency and convergence rate. In this chapter, three ANN such as the Learning Vector Quantization (LVQ), Back Propagation Neural Network (BPN) and the Modified BPN are demonstrated in the context of tumor segmentation. The focus of this chapter is on the modified approach which is an innovative contribution for tumor segmentation.

5.2 PROPOSED METHODOLOGY OF ANN BASED SEGMENTATION

The framework of the ANN based image segmentation technique is shown in Figure 5.1. Both the simulated MR images and the real-time images are used in this work. The input MR images are pre-processed using the skull removal technique. In this approach, the unwanted extra-cranial tissues are removed to enhance the segmentation efficiency. The total number of textural features used in this work is 8. The detailed account on the image database, pre-
processing techniques and the feature extraction methodologies are discussed in sections 3.2, 3.3 and 3.4.

The rest of this chapter is organized as follows: Section 5.3 deals with the LVQ based image segmentation techniques, Section 5.4 covers the BPN based methodologies and Section 5.5 details the modified BPN approach for segmentation. Further, Section 5.6 illustrates the experimental results and Section 5.7 concludes this chapter by highlighting the significant findings of this work.

5.3 LVQ NEURAL NETWORK

LVQ is a pattern classification method in which each output unit represents a particular class or category. During training, the output units are positioned (by adjusting their weights through supervised training) to approximate the decision surfaces of the theoretical Bayes classifier. It is assumed that a set of training patterns with known classifications is provided along with the randomly initialized weight values. After training, input vector is classified by the LVQ network by assigning it to the same class as the output unit that has its weight vector closest to the input vector.
5.3.1 Architecture of LVQ

The prior necessity of the LVQ based implementation is to determine the number of neurons in the different layers of the LVQ. Basically, LVQ consists of an input layer, competitive layer and the output layer. The input layer is used to distribute the inputs to the competitive layer which is further associated with the output layer (linear layer). Hence, the number of input layer neurons is equal to the number of input features. The number of input features used in this work is 8. The number of output layer neurons is equal to the number of pre-defined classes. The input image is segmented into four classes among which the tumor cluster is the region of interest of this work. Hence, the number of output layer neurons is 4. The sub-classes are denoted by the competitive layer neurons which must be a multiple of the output layer neurons. It is selected randomly and the number of neurons used in this work is 12 (three neurons for each class). The architecture of LVQ is shown in Figure 5.2.

![Figure 5.2 Architecture of LVQ](image_url)

In the above figure, the first three neurons of the competitive layer are connected to the first neuron of the output layer and the next three neurons are connected to the second neuron of the output layer and so on. These connections are not
associated with any weights but to show the association between the sub-classes and the main classes. The weight matrix is represented by $\overline{W}$ and the target vectors are represented by ‘$T$’. The pre-defined classes are denoted by this target vectors. The detailed operation of the LVQ network is discussed in the next section.

5.3.2 Training algorithm of LVQ

The summarized mathematical concepts of LVQ are displayed in the following algorithm.

Step 1: The weight vectors ($\overline{W}$) and the learning rate ($\alpha$) are initialized.

Step 2: The steps 3-4 are repeated for each training vector.

Step 3: The Euclidean distance between the inputs and weights are estimated and the hidden layer neuron ‘$j$’ for which the distance value is minimum is determined.

Step 4: The weights $W_j$ are adjusted as follows:

If $T = C_j$ (class or the category represented by the jth unit), then

$$W'_j = W_j + \alpha \{X - W_j\}$$  \hspace{1cm} (5.1)

If $T \neq C_j$, then

$$W'_j = W_j - \alpha \{X - W_j\}$$  \hspace{1cm} (5.2)

Step 5: Test for stopping condition which is usually the specified number of iterations.

Theoretically, the competitive layer neurons are allowed to compete with each other to determine the winner neuron based on the distance measure. The output of the winner neuron is set to be ‘1’ and the rest as ‘0’. Next, the output layer linearly associates these sub-classes to the main class by performing a comparison between the target vectors and the output of the competitive layer neurons. The target vector is also represented in binary form for the corresponding input vector. The weight vectors are updated using equation (5.1) if a match occurs (or) the weights are adjusted using equation (5.2) if a mismatch occurs. This process is repeated for the specified number of iterations to obtain the stabilized weights. These weight vectors are further used to test the LVQ network to segment the
unknown input data. During the testing process, the unknown input will be assigned to the first class if any of the first three neurons win the competition. Otherwise, the unknown input data is assigned to the class for which the associated neuron wins the competition.

**5.3.3 Implementation of LVQ**

The software used for the implementation of the LVQ neural network is MATLAB. The steps used are:

1) The input vectors and the target vectors are specified.
2) The minimum and the maximum values of the input ranges are given.
3) The number of layers and the number of neurons in layers are specified.
4) The LVQ network is created.
5) The training parameters such as the number of iterations are specified.
6) The neural network is trained using the conventional algorithm.
7) After training, the network is tested to determine the segmentation efficiency.

The value of the learning rate (α) is 0.7. The maximum number of iterations used for the LVQ network is approximately 1000. The binary representation of the target vector is used in this work. Finally, the segmentation efficiency of the proposed network is determined only using the testing inputs.

**5.4 BACK PROPAGATION NEURAL NETWORK**

BPN is the primarily used supervised neural network for pattern recognition. It is framed by generalizing the Widrow-Hoff learning rule to multiple layer network and non linear differentiable transfer function. Input vectors and the corresponding target vectors are used to train the network until it segment input vectors in an appropriate way as defined in this work.

**5.4.1 Architecture of BPN**

BPN is a multilayer network with an input layer, hidden layer and the output layer. The number of neurons in the input layer is equal to the number of input features and the number of output layer neurons is equal to the number of output
classes. The number of output classes is four and hence the number of output layer neurons is 4. The number of hidden layer neurons is fixed to be 12 to ensure uniformity among the three neural networks reported in this chapter. The architecture of BPN is shown in Figure 5.3.

Two set of weight matrices are used in this work with the connections between the input layer, hidden layer and output layer in a systematic fashion. The target vectors are supplied to the output layer in the opposite direction.

**5.4.2 Training algorithm of BPN**

The training algorithm of BPN is carried out in three stages: the feed forward of the input training pattern, the back propagation of the associated error and the adjustment of the weights. The training algorithm is summarized as follows:

**Step 1:** The weights $u_{ij}$ and $v_{jk}$ are randomly initialized.

**Step 2:** The steps 3-9 are repeated for each training pair.

**Stage 1: Feed forward**

Step 3: Each input unit ($x_i$) receives the input signal and broadcasts the signals to the hidden layer neurons ($z_j$) where $i=1, 2,\ldots,5$ and $j=1, 2,\ldots,12$.

**Step 4:** Each hidden unit sums its weighted input signals.
\[ z_{inj} = \sum x_i \cdot u_{ij} \]  
(5.3)

It is then applied to the activation function to compute its output signal.

\[ z_j = f(z_{inj}) \]  
(5.4)

The activation function used in this work is sigmoid function. This output signal is fed to the output layer neurons \( y_k \) where \( k = 1, 2...4 \).

Step 5: Each output unit sums its weighted input signals.

\[ y_{inj} = \sum z_j \cdot v_{jk} \]  
(5.5)

It is then applied to the activation function (sigmoidal function) to compute its output signal

\[ y_k = f(y_{inj}) \]  
(5.6)

**Stage 2: Back propagation of error**

Step 6: Each output unit receives a target pattern \( (T_k) \) and compute its error information term.

\[ \delta_k = (T_k - y_k) \cdot f'(y_{inj}) \]  
(5.7)

The weight correction term is then calculated using the formula

\[ \Delta v_{jk} = \alpha \cdot \delta_k \cdot z_j \]  
(5.8)

where \( \alpha \) is the learning rate.

Step 7: Each hidden unit compute its error information term

\[ \delta_j = \sum \delta_k v_{jk} \cdot f'(z_{inj}) \]  
(5.9)

The weight correction term is the calculated.

\[ \Delta u_{ij} = \alpha \cdot \delta_j \cdot x_i \]  
(5.10)

**Stage 3: Update weights**

Step 8: Each output unit updates its weights by

\[ v_{jk}(\text{new}) = v_{jk}(\text{old}) + \Delta v_{jk} \]  
(5.11)

Each hidden unit updates its weights by

\[ u_{ij}(\text{new}) = u_{ij}(\text{old}) + \Delta u_{ij} \]  
(5.12)
Step 9: The training stops when the weight correction term in equation (5.11) and equation (5.12) are equal to zero (or) a predefined minimum value which indirectly indicates $(T_k - y_k)$ equal to zero (or) a predefined minimum value.

5.4.3 Implementation of BPN

The experiments on BPN are performed with MATLAB without the usage of the in-built training and testing functions. The learning rate used in BPN is 0.7. The tolerable error difference is 0.01. The number of iterations used for BPN is approximately 2500. This methodology of training is computationally tedious which accounts for the slow convergence. Since the hidden layer is not directly connected with the target vector, the mathematical operations for weight adjustment in the hidden layer are significantly high which also adds to the complexity. Since the training algorithm is iterative in nature, the probability of the algorithm being trapped in local minimum is also high which reduces the segmentation efficiency to high extent. These drawbacks are tackled in the proposed Modified BPN which is explained in the next section.

5.5 MODIFIED BPN

The high speed back propagation neural network is framed by performing two modifications in the architecture and the training algorithm of the conventional BPN. The performance of the existing Back propagation neural network in terms of accuracy and convergence time period is enhanced by these two modifications. In terms of architecture, the target vectors are supplied to the hidden layer in addition to the output layer neurons in the MBPN. A dimension mismatch problem may arise between the input vector and the target vector but it can be solved by representing the same target vector in a different manner. Since four classes are used in this work, the target vector given to the output layer is represented by 4 bits whereas the same target vector supplied to the hidden layer is represented by 12 bits. Thus the target vector is supplied to both the hidden layer and the output layer without any complications. Since the target is also
supplied to the hidden layer, the term ‘middle layer’ is used instead of ‘hidden layer’.

In terms of training algorithm, there is no necessity for the weight adjustment criterion in the MBPN. Since the target vector is given to both the hidden and the output layers, the multilayer MBPN can be considered as two single layer networks with independent weight calculation procedure for both the layers. The error is calculated for each layer and the weights are adjusted accordingly which avoids the complex mathematical calculations involved in training the weights between the input layer and the hidden layer in the conventional BPN. Since this network is devoid of training, the convergence time period is very less when compared with the conventional BPN.

5.5.1 Architecture of MBPN

The architecture of the proposed modified approach of MBPN is shown in Figure 5.4.

In the above figure, dashed lines are used to represent the target vector which is supplied to the neurons in the hidden layer and the output layer. Hence, this multilayer network can be divided into two supervised single layer networks with the same target vector: one with the input layer and the hidden layer with \( x_i \) as
inputs and the other with the hidden layer and the output layer with $z_j$ as inputs. The number of hidden layer neurons is fixed to be 12 in order to ensure uniformity among all three neural networks reported in this chapter.

5.5.2 Training algorithm of MBPN

The main objective of the training methodology of the conventional BPN is to calculate the weight matrices by minimizing the error value. In this modified algorithm, the weight matrices are calculated without any training methodology. The steps of the conventional training algorithm is followed but in the reverse direction. The detailed steps of the MBPN algorithm are given below.

Step 1: The stabilized weight values are obtained when the error value (target-output) is equal to zero (or) a predefined minimum value. The error value used for convergence in this work is 0.01. The following procedure uses this concept for weight matrices calculation.

Step 2: Four training pairs are formed by averaging the feature values of the training pixels of each class. For each training pair, do steps 3-8.

Calculation of weight matrix between input layer and hidden layer

Step 3: Since $(T_j - z_j) = 0.01$ for convergence, the output of the hidden layer neurons is set equal to the target values

$$z_j = T_j - 0.01$$

where $t_j$ is the target supplied to the hidden layer.

Step 4: Once the output value is calculated, equation (5.4) is used to calculate the sum of the weighted input signals ($z_{\text{in}}$). Since the sigmoid activation function is used, the following equation yields the value for $z_{\text{in}}$.

$$z_{\text{in}} = \ln \left[ \frac{z_j}{1 - z_j} \right]$$

(5.14)

Step 5: Based on the values of $z_{\text{in}}$, the weight matrix $u_{ij}$ is calculated using Equation (5.3).
Calculation of weight matrix between hidden layer and output layer

Step 6: Using the condition in step 3, the output value is set equal to the target value

\[ y_k = T_k - 0.01 \]  \hspace{1cm} (5.15)

Step 7: The same procedure in step 4 is used to calculate the value for \( y_{in_k} \) using the following formula

\[ y_{in_k} = \ln \left( \frac{y_k}{1 - y_k} \right) \]  \hspace{1cm} (5.16)

Step 8: Since the values of \( z_j \) and \( y_{in_k} \) are known, the weight matrix \( v_{jk} \) is calculated using Eqn. (5.5).

These weight calculations are devoid of training which reduces the convergence time period. The weight values are calculated based on the convergence condition and hence these represent the stabilized weight matrix. Since error value of zero is not always possible in practical applications, error values such as 0.01, 0.001 can be used depending on the nature of the application. The dimension mismatch problem in supplying the target vector to the hidden layer is solved by different representation of the target vector. The testing of the unknown pixels is performed by the usual procedure of the conventional BPN with the weight matrices obtained from the training procedure. The pre-defined classes are represented by each output layer neuron and the pixel is allotted to the class for which the output value is maximum.

5.5.3 Implementation of MBPN

The parameters used in MBPN are same as that of BPN. But, there are several other issues which are discussed in this section.

1) An error value of ‘0’ cannot be used if the target representation is in terms of binary values. In such a case, a non-binary target representation can be used. But in any case (conventional neural networks), achieving a zero training error is practically non-feasible.

2) The dimensions for the weight calculation equation for the output layer is
\[ [4\times4] = [4\times12] [12\times4]; A=BC \]

The dimensions for the weight calculation equation for the input layer is
\[ [4\times12] = [4\times8] [8\times12]; B=DE \]

3) Different set of training samples have yielded different values (difference is <0.1) but an ‘averaging’ methodology with random samples solve this problem. Since the difference is small, the results remain unaffected.

4) Also, the convergence rate may vary for different processors. In any case, MBPN will be superior to any other conventional neural networks since it is devoid of iterations.

5.6 EXPERIMENTAL RESULTS AND DISCUSSIONS

The experiments are carried out on the Pentium processor with speed 1.6 GHz and 1 GB RAM. The software used for the implementation is MATLAB (version 7.0), developed by Math works Laboratory. The proposed algorithm is initially experimented with simulated images and further the experiments are extended with the real-time images. Experiments on LVQ and BPN are also conducted to show the superior nature of the proposed approach. Since the proposed approach is employed for segmentation, the training data points are the pixels and not the entire image. The simulated image data set used for this segmentation is shown in Table 5.1.

<table>
<thead>
<tr>
<th>Input image</th>
<th>Class</th>
<th>Training data (pixels/image)</th>
<th>Testing data (pixels/image)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe tumor stage</td>
<td>White matter</td>
<td>40</td>
<td>65536 (Full image)</td>
</tr>
<tr>
<td></td>
<td>Grey matter</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tumor</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Moderate tumor stage</td>
<td>White matter</td>
<td>40</td>
<td>65536 (Full image)</td>
</tr>
<tr>
<td></td>
<td>Grey matter</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tumor</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Mild tumor stage</td>
<td>White matter</td>
<td>40</td>
<td>65536 (Full image)</td>
</tr>
<tr>
<td></td>
<td>Grey matter</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tumor</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>
In this work, 140 training points are chosen from each of the simulated images according to their location in the phantom images to generate the correspondence between the pixels and the corresponding class. But, the entire image is used for the testing purpose. The simulated images include abnormal brain images of different stages.

The results are analyzed in terms of segmentation efficiency and correspondence ratio. A brief analysis on the convergence rate for the three techniques is also reported in this work. Segmentation efficiency is a measure of the correctly classified pixels and correspondence ratio is the measure of non-tumor pixels classified as tumor pixels. Segmentation Efficiency (SE) is defined as the ratio of the true positives to the number of ground truth tumor pixels. These calculations are based on True Positive (TP) pixels, False Positive (FP) pixels and the Ground Truth (GT) pixels

\[
SE = \frac{TP}{GT} 
\]  

(5.17)

The value of Correspondence Ratio (CR) is 1 when the system identifies no pixel as tumor when none existed and 0 when the system identifies a pixel as tumor when none existed.

\[
CR = \frac{TP - (0.5 \times FP)}{GT} 
\]  

(5.18)

Much emphasis is given to the quantitative analysis of the region of interest (abnormal tumor portion). An analysis for other regions such as WM, GM and CSF is also reported to show the superior nature of the proposed approach.

5.6.1 Result analysis of LVQ

The severe, moderate and mild stage input images are used in the implementation of LVQ.

5.6.1.1 Segmentation efficiency results of LVQ

The segmentation efficiency analysis is further categorized into qualitative analysis and quantitative analysis. In the qualitative analysis, the image is initially clustered into four categories among which the region of interest is the tumor region. Three different stages of inputs such as severe, moderate and mild stage
images are used in this work. The segmented images for the severe stage image are shown in Figure 5.5.

![Segmented Images](image1.png)

Figure 5.5 Severe image results of LVQ: (a) Input image (b) Gray matter phantom (c) White matter phantom (d) CSF phantom (e) Tumor phantom (f) Gray matter segment (g) White matter segment (h) CSF segment (i) Tumor segment.

The segmented images for the moderate stage image are shown in Figure 5.6.

![Segmented Images](image2.png)

Figure 5.6 Moderate image results of LVQ: (a) Input image (b) Gray matter phantom (c) White matter phantom (d) CSF phantom (e) Tumor phantom (f) Gray matter segment (g) White matter segment (h) CSF segment (i) Tumor segment.

The segmented images for the mild stage image are shown in Figure 5.7.
Figure 5.7 Mild image results of LVQ: (a) Input image (b) Gray matter phantom (c) White matter phantom (d) CSF phantom (e) Tumor phantom (f) Gray matter segment (g) White matter segment (h) CSF segment (i) Tumor segment.

The phantom images are the ground truth images available in the website. The segment images are the output of the LVQ network. A visual analysis on these images has shown low quality results for LVQ. The interference of other tissues with the gray matter segment is clearly visible in these images. The number of discontinuities in the CSF segment is also much high. The white matter segment is comparatively good but the LVQ failed to detect any tumor tissues in the mild stage image. This has verified the fact that LVQ holds good only if the region of interest is sufficiently large which may not be possible always. These inferences are further verified using the quantitative analysis shown in Table 5.2.

Table 5.2 Quantitative results of LVQ for the region of interest

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage</td>
<td>297</td>
<td>262</td>
<td>69</td>
<td>88</td>
<td>0.77</td>
</tr>
<tr>
<td>Moderate stage</td>
<td>86</td>
<td>62</td>
<td>30</td>
<td>72</td>
<td>0.55</td>
</tr>
<tr>
<td>Mild stage</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The LVQ network has yielded very low quality results which are evident from Table 5.2. The segmentation efficiency and the correspondence ratio values are considerably reduced. The low efficiency of the LVQ network is due to the random initialization of many parameters such as the initial weights, learning rate, etc. The distance measure used for selecting the prototype of each class also has significant effect on the efficiency. Improper selection of distance measures may lead to low quality results for the LVQ network. The segmentation efficiency is also dependent on the number of iterations. The quantitative analysis of the LVQ for other regions is shown in Table 5.3.

Table 5.3 Quantitative analysis of LVQ for WM, GM and CSF

<table>
<thead>
<tr>
<th>Input</th>
<th>GT</th>
<th>TP</th>
<th>FP</th>
<th>SE</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage image</td>
<td>WM</td>
<td>11764</td>
<td>8743</td>
<td>2689</td>
<td>74.34</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>10936</td>
<td>2766</td>
<td>2712</td>
<td>25.29</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>2883</td>
<td>110</td>
<td>44.37</td>
</tr>
<tr>
<td>Moderate stage image</td>
<td>WM</td>
<td>11764</td>
<td>8777</td>
<td>768</td>
<td>74.43</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>10936</td>
<td>3165</td>
<td>583</td>
<td>28.91</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>3403</td>
<td>365</td>
<td>52.36</td>
</tr>
<tr>
<td>Mild stage image</td>
<td>WM</td>
<td>11764</td>
<td>10926</td>
<td>1758</td>
<td>92.87</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>10936</td>
<td>3501</td>
<td>1875</td>
<td>32.01</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>3019</td>
<td>1052</td>
<td>46.46</td>
</tr>
</tbody>
</table>

The average segmentation efficiency is 79% for the WM, 47% for the CSF and 28% for the GM region. The correspondence ratio of the LVQ for WM, CSF and GM is insufficient for the practical applications. Thus, the inferior nature of the LVQ network is verified through the experimental results shown in Table 5.3.

5.6.1.2 Convergence rate analysis of LVQ

The computational complexity of the LVQ network is relatively lesser than other neural networks. Since the number of weight matrix to be determined is only one, the computational complexity is sufficiently low. This has lead to single weight adjustment equation which accounts for the simplicity of the system. Also, the mathematical calculations involved in LVQ are only the distance measure calculation unlike BPN which involves both NET value and OUT value
calculation. These factors have led to the better convergence time of LVQ which is approximately 820 CPU seconds for achieving the efficiency mentioned above. Though faster than other neural networks, it is still inferior to the due to the inferior segmentation efficiency.

5.6.2 Result analysis of BPN

The severe, moderate and mild stage input images are also used in the implementation of BPN.

5.6.2.1 Segmentation efficiency results of BPN

Initially, the qualitative results are displayed followed by the extensive quantitative analysis. The segmented images for the severe stage, moderate stage and mild stage images are shown in Figures 5.8, 5.9, 5.10 respectively. The phantom images have been already displayed in the analysis of LVQ. Only the segmented outputs using BPN is shown in this section.

![Segmentation results of BPN](image1)

Figure 5.8 Severe image results of BPN: (a) Input image (b) Gray matter segment (c) White matter segment (d) CSF segment (e) Tumor segment.

![Segmentation results of BPN](image2)

Figure 5.9 Moderate image results of BPN: (a) Input image (b) Gray matter segment (c) White matter segment (d) CSF segment (e) Tumor segment.
From the above results, it is evident that the efficiency of the conventional BPN technique is better than the LVQ approach. The difficulty in segmenting the gray tissues is once again proved by these results. The segmented outputs of BPN are comparatively better than the results of LVQ. An improvement in the correspondence ratio is also seen in the BPN network.

But, several discontinuities are found in the segmented results of BPN while the level of discontinuities is much lesser in the phantom images which indirectly indicate that better results can be achieved by the modified BPN. In the mild stage case, the tumor tissue is hardly seen which again verified the low segmentation capability of the conventional BPN network for small size region of interest. Though the results of conventional BPN are not worse, the fact that has been proved is that these results can be further improved by the modified BPN technique. The quantitative results of the conventional BPN network are shown in Table 5.4.

Table 5.4 Quantitative results of conventional BPN for region of interest

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage</td>
<td>297</td>
<td>277</td>
<td>90</td>
<td>93</td>
<td>0.78</td>
</tr>
<tr>
<td>Moderate stage</td>
<td>86</td>
<td>69</td>
<td>31</td>
<td>80</td>
<td>0.62</td>
</tr>
<tr>
<td>Mild stage</td>
<td>21</td>
<td>9</td>
<td>9</td>
<td>43</td>
<td>0.22</td>
</tr>
</tbody>
</table>

The gradual improvement in the quality of results of BPN over the LVQ is evident from Table 5.4. The number of detected tumor pixels is high since the
interference of non-tumorous tissues is low. Both these factors have contributed to the better segmentation efficiency and the correspondence ratio. Even though, the efficiency is better than LVQ, the performance measures are not sufficient for practical applications. One of the main reasons behind these inferior results is the iterative nature of the conventional BPN. This iteration-dependent behavior of BPN has lead to the concept of local minimum which ultimately reduces the quality of the results. Another factor is the randomly initialized weights which also can reduce the quality of results. The quantitative analysis for other regions is shown in Table 5.5.

Table 5.5 Quantitative analysis of conventional BPN for WM, GM and CSF

<table>
<thead>
<tr>
<th>Input</th>
<th>GT</th>
<th>TP</th>
<th>FP</th>
<th>SE</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage image</td>
<td>WM</td>
<td>11764</td>
<td>9250</td>
<td>1566</td>
<td>78.65</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>10936</td>
<td>3165</td>
<td>1282</td>
<td>28.94</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>3461</td>
<td>1914</td>
<td>53.27</td>
</tr>
<tr>
<td>Moderate stage image</td>
<td>WM</td>
<td>11764</td>
<td>8849</td>
<td>1095</td>
<td>75.04</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>17936</td>
<td>5447</td>
<td>659</td>
<td>30.36</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>3245</td>
<td>1043</td>
<td>50.00</td>
</tr>
<tr>
<td>Mild stage image</td>
<td>WM</td>
<td>11764</td>
<td>10493</td>
<td>2270</td>
<td>89.19</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>10936</td>
<td>2768</td>
<td>1342</td>
<td>25.30</td>
</tr>
<tr>
<td></td>
<td>CSF</td>
<td>6497</td>
<td>3304</td>
<td>2021</td>
<td>50.85</td>
</tr>
</tbody>
</table>

The average segmentation efficiency is 81% for the WM, 51% for the CSF and 28% for the GM region. The segmentation ability is better than LVQ with WM yielding a higher value and other regions yielding a lower value. Even though the conventional BPN yields sufficiently accurate results, the scope for improvement in terms of efficiency is always available in these networks.

5.6.2.2 Convergence rate analysis of BPN

The conventional BPN is also computationally heavy in terms of the training algorithm. The required number of arithmetic operations is significantly high due to the two weight adjustment equations. Several parameters such as the initial weights and learning rate also have impact on the convergence rate. If the learning rate is too low, then the time period requirement is very high which makes the selection of the learning rate very crucial. The efficiency is also
indirectly dependent on the number of iterations and hence the convergence time period drastically increases for better efficiency. The increase in the complexity of the algorithm has resulted in requirement for 1650 CPU seconds which has made the proposed approach practically non-feasible for real-time medical applications. This time period is estimated for the above mentioned data set and processor specifications. The iterative nature of the conventional BPN has also limited the possibility of hardware implementation of such algorithms. Though the efficiency is sufficient, BPN significantly suffers from the drawback of high computational complexity which is eliminated by the modified BPN discussed in the next section.

5.6.3 Result analysis of MBPN

The segmentation efficiency results are discussed initially followed by the convergence rate analysis.

5.6.3.1 Segmentation efficiency results of MBPN

The qualitative results of Modified BPN for the three stage input images are displayed in Figures 5.11-5.13.

![Figure 5.11 Severe image results of MBPN: (a) Input image (b) Gray matter segment (c) White matter segment (d) CSF segment (e) Tumor segment.](image)

The superior nature of the proposed approach in terms of segmentation efficiency is verified with the careful observation of the above mentioned results. The phantom images are shown in the analysis of LVQ and the segmented outputs are shown in Figures 5.11. The segmented outputs of gray matter, white matter, CSF and the region of interest (tumor region) have been segmented with sufficiently
high accuracy. The segmented images for the moderate stage image are shown in Figure 5.12.

Figure 5.12 Moderate image results of MBPN: (a) Input image (b) Gray matter segment (c) White matter segment (d) CSF segment (e) Tumor segment.

The results of the moderate stage image are better than the segmented outputs of LVQ and BPN. Among the tissues, white matter and CSF segmented outputs are better than the gray matter segment in both moderate and severe stage images. One of the reasons is that the gray matter is found to be sparsely located in the image whereas the other two tissues are found to be available in bulk in the image. The segmented images for the mild stage image are shown in Figure 5.13.

Figure 5.13 Mild image results of MBPN: (a) Input image (b) Gray matter segment (c) White matter segment (d) CSF segment (e) Tumor segment.

In the above figures, a careful analysis of the input images are required to distinguish between the three stages of tumor which otherwise looks similar. Figure 5.13 (e) illustrate the ability of the proposed network for detecting the tumor even if the volume is low in the input image. In the case of mild stage image, the tumor portion is hardly visible in the phantom image which makes the segmentation process extremely difficult for the artificial neural network. From the above results, it is evident that the proposed approach is successful in even
identifying the very low level tumor portion in the abnormal images. The quantitative results of MBPN for the region of interest are displayed in Table 5.6.

Table 5.6 Quantitative results of MBPN for the tumor region

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage image</td>
<td>297</td>
<td>288</td>
<td>52</td>
<td>96</td>
<td>0.88</td>
</tr>
<tr>
<td>Moderate stage image</td>
<td>86</td>
<td>75</td>
<td>20</td>
<td>87</td>
<td>0.75</td>
</tr>
<tr>
<td>Mild stage image</td>
<td>21</td>
<td>15</td>
<td>9</td>
<td>71</td>
<td>0.55</td>
</tr>
</tbody>
</table>

The severe stage image has shown commanding results than the other two stages since the tumor portion is highly significant in this image. The segmentation efficiency of severe and moderate stage images is sufficiently good. The reason for the inferior segmentation efficiency of mild stage is that the number of ground truth pixels is very less. Even a loss of single tumor pixel has great impact on the final efficiency. Similarly, the CR value is also nominal stating the fact that other tissues hardly interfere with the tumor tissues. The size of the tumor segment and tumor phantom is almost same which is evident from the qualitative analysis indicating the low impact of non-tumorous tissues. But, still the mild stage image yields a lower CR value over the other stages due to the less ground truth pixels. The quantitative analysis of MBPN for WM, GM and CSF is shown in Table 5.7.

Table 5.7 Quantitative analysis of MBPN for WM, GM and CSF

<table>
<thead>
<tr>
<th>INPUT</th>
<th>GT</th>
<th>TP</th>
<th>FP</th>
<th>SE</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severe stage image</td>
<td>WM 11764</td>
<td>10477</td>
<td>3568</td>
<td>89.09</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>GM 10936</td>
<td>3387</td>
<td>1122</td>
<td>30.97</td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>CSF 6497</td>
<td>2913</td>
<td>517</td>
<td>44.83</td>
<td>0.41</td>
</tr>
<tr>
<td>Moderate stage image</td>
<td>WM 11764</td>
<td>10064</td>
<td>3181</td>
<td>85.35</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>GM 10936</td>
<td>3975</td>
<td>1934</td>
<td>36.30</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>CSF 6497</td>
<td>3821</td>
<td>1145</td>
<td>58.83</td>
<td>0.50</td>
</tr>
<tr>
<td>Mild stage image</td>
<td>WM 11764</td>
<td>11051</td>
<td>53</td>
<td>93.93</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td>GM 10936</td>
<td>4277</td>
<td>3079</td>
<td>39.10</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>CSF 6497</td>
<td>3345</td>
<td>1328</td>
<td>51.48</td>
<td>0.41</td>
</tr>
</tbody>
</table>
The average segmentation efficiency is 89% for the WM, 51% for the CSF and 35% for the GM region. The segmentation efficiency is high for WM since the portion is present in bulk in the image whereas the CSF and the GM are sparsely available. A reasonable correspondence ratio is also achieved through this approach. On the whole, the performance measures are much better than the other conventional neural networks. It may be noted that the emphasis of this research work is on the abnormal tumor portion which has been segmented with high accuracy. Thus, the segmentation capability of the proposed approach is proved through these experimental results.

5.6.3.2 Convergence rate analysis of MBPN

A further analysis is also made on these networks in terms of the computational complexity and the convergence rate which is one of the objectives of the work. The number of mathematical calculations required for MBPN is highly reduced since the proposed approach is an iteration-free network. If the number of required calculations are ‘m’ for MBPN, then BPN and LVQ would require almost ‘nt’ calculations where ‘t’ is the number of iterations and ‘n > m’. This approach has also eliminated the necessity for random initial weight initialization which often impacts the segmentation efficiency. This reduction in the computational complexity has reduced the convergence rate requirement of the proposed approach to high extent. The required training time period is 0.965 CPU seconds and has made the proposed approach practically feasible for real-time medical applications. Another significant feature is that this work has tackled the basic drawback of requirement for high training time of the conventional BPN. Thus, the MBPN network has found to be efficient in terms of segmentation efficiency and computational complexity.

From the above results, it is evident that the proposed Modified BPN is better than the other two approaches. To show the robustness of the proposed approach, the three networks are also tested with real-time images. A brief analysis of the experimental results is discussed in the next section.
5.6.4 Experimental results of the AI techniques with real-time data set

The three AI techniques are also implemented with the real-time images to prove the effectiveness of the proposed approach. Since, the phantom images are available only for the tumor region, bi-level classification is performed with 540 real-time images. Hence, the number of output layer neurons is 2. In this work, 50 pixels from tumor portion are given for training and the testing process is done with the whole image. The experiments are conducted on all the images but only sample output images are shown in this section. The qualitative analysis of LVQ technique is shown in Figure 5.14.

![Sample segmented real time images with LVQ](image)

Figure 5.14 Sample segmented real time images with LVQ: (a) input images, (b) tumor segment, (c) Tumor phantom

The quantitative analysis of the LVQ technique is shown in Table 5.8.
Table 5.8 Quantitative analysis of LVQ for real-time dataset

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>1596</td>
<td>1222</td>
<td>127</td>
<td>77</td>
<td>0.73</td>
</tr>
<tr>
<td>Image 2</td>
<td>3836</td>
<td>2945</td>
<td>540</td>
<td>77</td>
<td>0.70</td>
</tr>
<tr>
<td>Image 3</td>
<td>5405</td>
<td>4307</td>
<td>2108</td>
<td>80</td>
<td>0.60</td>
</tr>
</tbody>
</table>

The average segmentation efficiency is 78% and the average correspondence ratio is 0.68. From the above table, it is evident that the accuracy of LVQ is not sufficient for practical applications such as brain image analysis. The qualitative analysis of BPN based segmentation is shown in Figure 5.15.

![Sample segmented real time images with BPN](image)

(a) (b) (c)

Figure 5.15 Sample segmented real time images with BPN: (a) input images, (b) tumor segment, (c) Tumor phantom.

The quantitative analysis of the BPN technique is shown in Table 5.9.
Table 5.9 Quantitative analysis of BPN for real-time dataset

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>1596</td>
<td>1332</td>
<td>47</td>
<td>83</td>
<td>0.82</td>
</tr>
<tr>
<td>Image 2</td>
<td>3836</td>
<td>3251</td>
<td>530</td>
<td>85</td>
<td>0.78</td>
</tr>
<tr>
<td>Image 3</td>
<td>5405</td>
<td>4527</td>
<td>538</td>
<td>84</td>
<td>0.79</td>
</tr>
</tbody>
</table>

The average segmentation efficiency is 84% and the average correspondence ratio is 0.79. The efficiency of the BPN network is better than the LVQ network but an improvement in the performance measures would make BPN more suitable for real-time applications. The qualitative analysis of Modified BPN based segmentation is shown in Figure 5.16.

![Sample segmented real time images with MBPN](image_url)

Figure 5.16 Sample segmented real time images with MBPN: (a) input images, (b) tumor segment, (c) Tumor phantom

The quantitative analysis of the Modified BPN technique is shown in Table 5.10.
Table 5.10 Quantitative analysis of Modified BPN for real-time dataset

<table>
<thead>
<tr>
<th>Input</th>
<th>No. of ground truth pixels</th>
<th>True Positive pixels</th>
<th>False Positive pixels</th>
<th>Segmentation Efficiency (%)</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>1596</td>
<td>1413</td>
<td>38</td>
<td>89</td>
<td>0.87</td>
</tr>
<tr>
<td>Image 2</td>
<td>3836</td>
<td>3572</td>
<td>373</td>
<td>93</td>
<td>0.88</td>
</tr>
<tr>
<td>Image 3</td>
<td>5405</td>
<td>5014</td>
<td>468</td>
<td>92</td>
<td>0.88</td>
</tr>
</tbody>
</table>

The average segmentation efficiency is 91% and the average correspondence ratio is 0.88. The superior nature of the proposed approach over the conventional approaches is verified from Table 5.10. Thus, the robustness of the proposed approach is verified using the real-time dataset. Thus, the proposed approach is highly efficient in terms of accuracy and convergence time.

5.7 CONCLUSION

In this work, three different ANN are implemented for tumor segmentation in MR brain images. Two of them are conventional ANN while the other is the modified ANN. The performance measures of these networks are analyzed to show the superior nature of the proposed approach (modified BPN). The results have clearly revealed that the modified BPN is much superior to BPN and LVQ in terms of segmentation efficiency and convergence time. While the segmentation efficiency of BPN is almost similar to LVQ, the time requirement for LVQ is significantly lesser than the conventional BPN. The drawbacks encountered in conventional neural networks are also tackled by this modified BPN. Thus, this work has suggested suitable alternate for conventional BPN in the context of medical imaging applications.