

Mri Brain Image Classification Using Probabilistic Neural Network And Tumor Detection Using Clustering Technique

¹M.V SubbaRao
Assistant Professor

²B.Revanth
Assistant Professor

³D.UdayKumar
Assistant Professor

⁴B.CH.S.N.L.S Saibaba
Assistant Professor

⁵K.RupendraSingh
Assistant Professor

Abstract:-The paper proposes an automatic support system for stage classification using Probabilistic neural network and to detect Brain Tumor through clustering methods for medical application. The detection of the Brain Tumor is a challenging problem, due to the structure of the Tumor cells. This paper presents a segmentation method, K-Means clustering algorithm, for segmenting Magnetic Resonance images to detect the Brain Tumor in its early stages [5]. The artificial neural network will be used to classify the stage of Brain Tumor that is benign, malignant or normal [7]. The manual analysis of the sputum samples is time consuming, inaccurate and requires intensive trained person to avoid diagnostic errors [2]. The segmentation results will be used as a base for a Computer Aided Diagnosis (CAD) system for early detection of Brain Tumor which will improve the chances of survival for the patient. The experimental result shows that the Clustering based segmentation results are more accurate and reliable than Thresholding and clustering methods in all cases [6]. Probabilistic Neural Network with image and data processing techniques was employed to implement an automated Brain Tumor classification. Decision making was performed in two stages: feature extraction using GLCM and the classification using Probabilistic Neural Network (PNN)[1]. The performance of the PNN classifier was evaluated in terms of training performance and classification accuracies. Probabilistic Neural Network gives fast and accurate classification than other neural networks and it is a promising tool for classification of the Tumors [3].

Keywords: Probabilistic Neural Network, Clustering, classification, Segmentation.

I. Introduction

Automated classification and detection of tumors in different medical images is motivated by the necessity of high accuracy when dealing with a human life. Also, the computer assistance is demanded in medical institutions due to the fact that it could improve the results of humans in such a domain where the false negative cases must be at a very low rate. It has been proven that double reading of medical images could lead to better tumor detection. But the cost implied in double reading is very high, that's why good software to assist humans in medical institutions is of great interest nowadays. Conventional methods of monitoring and diagnosing the diseases rely on detecting the presence of particular features by a human observer. Due to large number of patients in intensive care units and the need for continuous observation of such conditions, several techniques for automated diagnostic systems have been developed in recent years to attempt to solve this problem. Such techniques work by transforming the mostly qualitative diagnostic criteria into a more objective quantitative feature classification problem [1].

II. Methodology

In this paper the automated classification of brain magnetic resonance images by using some prior knowledge like pixel intensity and some anatomical features is proposed [1]. Currently there are no methods widely accepted therefore automatic and reliable methods for tumor detection are of great need and interest. The application of PNN in the classification of data for MR images problems are not fully utilized yet [5]. These included the clustering and classification techniques especially for MR images problems with huge scale of data and

consuming times and energy if done manually. Thus, fully understanding the recognition, classification or clustering techniques is essential to the developments of Neural Network systems particularly in medicine problems. Segmentation of brain tissues in gray matter, white matter and tumor on medical images is not only of high interest in serial treatment monitoring of “disease burden” in oncologic imaging, but also gaining popularity with the advance of image guided surgical approaches[6]. Outlining the brain tumor contour is a major step in planning spatially localized radiotherapy (e.g., Cyber knife, iMRT) which is usually done manually on contrast enhanced T1-weighted magnetic resonance Images (MRI) in current clinical practice. On T1 MR Images acquired after administration of a contrast agent (gadolinium), blood vessels and parts of the tumor, where the contrast can pass the blood–brain barrier are observed as hyper intense areas. There are various attempts for brain tumor segmentation in the literature which use a single modality, combine multi modalities and use priors obtained from population atlases.

III. Existing System

There are some approaches for image segmentation,

- Thresholding and
- Manual analysis

The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. The key of this method is to select the threshold value (or values when multiple-levels are selected). Several popular methods are used in industry including the maximum entropy method, Otsu's method (maximum variance), and k-means clustering. Recently, methods have been developed for thresholding computed tomography (CT) images. The key idea is that, unlike Otsu's method, the thresholds are derived from the radiographs instead of the (reconstructed) image.

A. Drawbacks

- Difficult to get accurate results
- Not applicable for multiple images for Tumor detection in a short time
- Medical Resonance images contain a noise caused by operator performance which can lead to serious inaccuracies classification [5].

IV. Proposed System

MRI Brain Image Classification and Tumor Detection Is Proposed Based On,

- Probabilistic Neural Network for classification.
- Clustering Algorithm (K means) for effective Segmentation.

A. Segmentation

Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels)[6]. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s) when applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like Marching cubes[6].

Advantages

- It can segment the Brain regions from the image accurately.
- It is useful to classify the Brain Tumor images for accurate detection.

- Brain Tumor will be detected in an early stages

B. Clustering

Clustering can be considered the most important *unsupervised learning* problem, so, it deals with finding a *structure* in a collection of unlabeled data. A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters [10]

Clustering algorithms may be classified as listed below

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering
- Probabilistic Clustering

In the first case data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. On the contrary the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value. A hierarchical clustering algorithm is based on the union between the two nearest clusters [10]. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted

C. K-Means Clustering

Cluster analysis, an important technology in data mining, is an effective method of analyzing and discovering useful information from numerous data. Cluster algorithm groups the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters [10]. Dissimilarities are assessed based on the attribute values describing the objects. Often, distance measures are used. As a branch of statistics and an example of unsupervised learning, clustering provides us an exact and subtle analysis tool from the mathematic view K-means algorithm belongs to a popular partition method in cluster analysis. The most widely used clustering error criterion is squared-error criterion, it can be

$$\text{defined as } J_c = \sum_{j=1}^n \sum_{k=1}^m \|x_k^{(j)} - m_j\|^2$$

where J_c is the sum of square-error for all objects in the database, x_k is the point in space representing a given object, and m_j is the mean of cluster C_j . Adopting the squared-error criterion, K-means works well when the clusters are compact clouds that are rather well separated from one another and are not suitable for discovering clusters with no convex shapes or clusters of very different size. For attempting to minimize the square-error criterion, it will divide the objects in one cluster into two or more clusters. Aiming at the dependency to initial conditions and the limitation of K-means algorithm that applies the square-error criterion to measure the quality of clustering, this paper presents a new improved K-means algorithm that is based on effective techniques of multi-sampling and once-clustering to search the optimal initial values of cluster centers. Our experimental results demonstrate the new algorithm can obtain better stability and excel the original K-means in clustering results

D. Pseudo Code For K-Means

In this section, we briefly describe the original K-means algorithm.

Original K-means(s,k), $s=\{x_1, x_2, \dots, x_n\}$.

Input: the number of clusters K and a dataset containing n objects (x_i).

Output: a set of k clusters c_j that minimize the squared-error criterion

Begin

$m=1$;

Initialize k prototypes $Z_j, j \in [1, K]$;

repeat

for $i=1$ to n do

Begin

for $j=1$ to k do

compute $D(X_i, Z_j) = |X_i - Z_j|$;

if $D(X_i, Z_j) \min \{ D(X_i, Z_j) \}$

then $X_i \in C_j$;

end;

if $m=1$ then $J_c(m) = \sum_{j=1}^k \sum_{x_i \in c_j} |X_i - Z_j|^2$;

$m=m+1$

For $j=1$ to k do

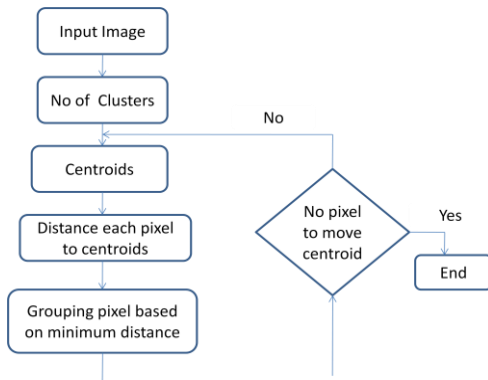
$Z_j = 1/n_j \sum_{i=1}^{N_j} x_i^{(f)} = \sum_{i=1}^{N_j} |X_i - Z_j|^{(f)}$;

Until $J_c(m) - J_c(m-1) < \epsilon$

End

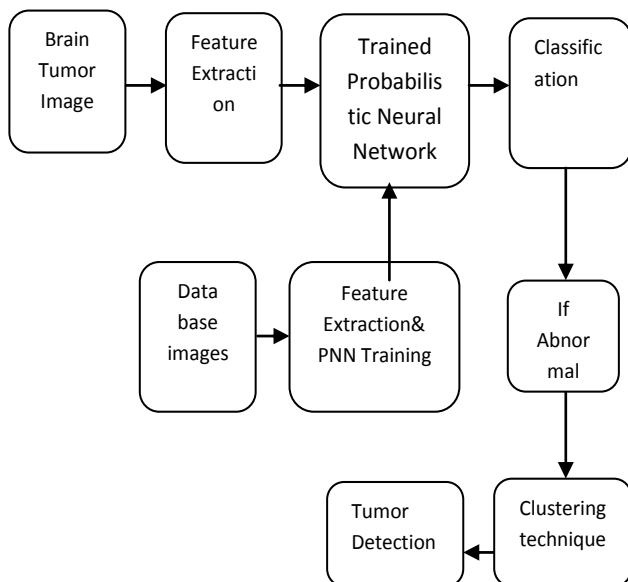
The computational complexity of original K-means algorithm is $O(ndk)$, where n is the total number of objects, k is the number of clusters, and d is the dimensions of datasets.

E. Algorithm Flow diagram



V. System Architecture

As the complexity of systems increases, the specification of the system decomposition is critical. Moreover, subsystem decomposition is constantly revised whenever new issues are addressed. Subsystems are merged into alone subsystem, a complex subsystem is split into parts, and some subsystems are added to take care of new functionality.



A. Modules description

- GLCM Feature Extraction.
- PNN Training and Classification.
- Clustering Method for Tumor Detection.

Texture Analysis

Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface, such as; clouds, leaves, bricks, fabric, etc. It also describes the relationship of the surface to the surrounding environment. In short, it is a feature that describes the distinctive physical composition of a surface.

Texture properties include:

- Coarseness
- Contrast
- Directionality
- Line-likeness
- Regularity
- Roughness

Texture is one of the most important defining features of an image. It is characterized by the spatial distribution of gray levels in a neighborhood [8]. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two-dimensional dependence texture analysis matrix is taken into consideration. This two-dimensional matrix is obtained by decoding the image file; jpeg, bmp, etc.

B. Methods of Representation

There are three principal approaches used to describe texture; statistical, structural and spectral.

- Statistical techniques characterize textures using the statistical properties of the grey levels of the points/pixels comprising a surface image. Typically, these properties are computed using: the grey level co-occurrence matrix of the surface, or the wavelet transformation of the surface.
- Structural techniques characterize textures as being composed of simple primitive structures called "Texel's" (or texture elements). These are arranged regularly on a surface according to some surface arrangement rules.
- Spectral techniques are based on properties of the Fourier spectrum and describe global periodicity of the grey levels of a surface by identifying high-energy peaks in the Fourier spectrum.

For optimum classification purposes, what concern us are the statistical techniques of characterization.[1]This is because it is these techniques that result in computing texture properties. The most popular statistical representations of texture are:

- Co-occurrence Matrix
- Tamura Texture
- Wavelet Transform

C. Co-occurrence Matrix

Originally proposed by R.M. Haralick, the co-occurrence matrix representation of texture features explores the grey level spatial dependence of texture [2]. A mathematical definition of the co-occurrence matrix is as follows [4]:

- Given a position operator $P(i,j)$,
- let A be an $n \times n$ matrix
- Whose element $A[i][j]$ is the number of times that points with grey level (intensity) $g[i]$ occur, in the position specified by P , relative to points with grey level $g[j]$.
- Let C be the $n \times n$ matrix that is produced by dividing A with the total number of point pairs that satisfy P . $C[i][j]$ is a measure of the joint probability that a pair of points satisfying P will have values $g[i], g[j]$.
- C is called a co-occurrence matrix defined by P .

Examples for the operator P are: “ i above j ”, or “ i one position to the right and two below j ”, etc.

This can also be illustrated as follows... Let t be a translation, then a co-occurrence matrix C_t of a region is defined for every grey-level (a, b) by [1]:

$$C_t(a,b) = \text{card}\{(s,s+t) \in R^2 | A[s]=a, A[s+t]=b\}$$

Here, $C_t(a, b)$ is the number of site-couples, denoted by $(s, s + t)$ that are separated by a translation vector t , with a being the grey-level of s , and b being the grey-level of $s + t$.

For example; with an 8 grey-level image representation and a vector t that considers only one neighbor, we would find [1]

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	0	0	0
2	0	1	0	2	0	0	0	0
3	0	0	1	1	0	0	0	0
4	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Figure: Classical Co-occurrence matrix

At first the co-occurrence matrix is constructed, based on the orientation and distance between image pixels. Then meaningful statistics are extracted from the matrix as the texture representation. Haralick proposed the following texture features:

1. Energy
2. Contrast
3. Correlation
4. Homogeneity
5. Entropy

Hence, for each Haralick texture feature, we obtain a co-occurrence matrix. These co-occurrence matrices represent the spatial distribution and the dependence of the grey levels within a local area. Each (i,j) th entry in the matrices, represents the probability of going from one pixel with a grey level of ' i ' to another with a grey level of ' j ' under a predefined distance and angle. From these matrices, sets of statistical measures are computed, called feature vectors.

Energy: It is a gray-scale image texture measure of homogeneity changing, reflecting the distribution of image gray-scale uniformity of weight and texture.

$$E = \sum_x \sum_y P(x,y)^2$$

Contrast: Contrast is the main diagonal near the moment of inertia, which measure the value of the matrix is distributed and images of local changes in number, reflecting the image clarity and texture of shadow depth.

$$I = \sum_x \sum_y (x-y)^2 p(x,y)$$

Entropy: It measures image texture randomness, when the space co-occurrence matrix for all values is equal, it achieved the minimum value.

$$S = - \sum_x \sum_y P(x,y) \log p(x,y)$$

Correlation Coefficient: Measures the joint probability occurrence of the specified pixel pairs.

Correlation: $\text{sum}(\text{sum}((x - \mu_x)(y - \mu_y)p(x, y)) / (\sigma_x \sigma_y))$

Homogeneity: Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

Homogeneity = $\text{sum}(\text{sum}(p(x, y) / (1 + [x - y])))$

D. Discrete Wavelet Transform (DWT)

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time).

Types of DWT

There are two types of DWT. They are

- One dimensional DWT(1D DWT)
- Two Dimensional DWT(2D DWT)

One Dimensional Dwt (1 -D)

The DWT of a signal x is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse g resulting in a convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k].$$

The signal is also decomposed simultaneously using a high-pass filter h . The outputs giving the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters are related to each other and they are known as a quadrature mirror filter.

Two Dimensional Dwt (2 -D)

However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter outputs are then subsample by 2 (Mallat's and the common notation is the opposite, g- high pass and h- low pass):

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n-k].$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n-k].$$

This decomposition has halved the time resolution since only half of each filter output characterizes the signal. However, each output has half the frequency band of the input so the frequency resolution has been doubled.

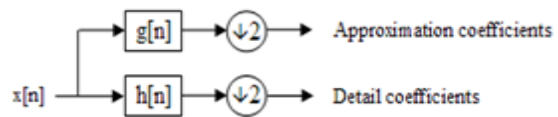


Figure: Block diagram of filter analysis

E. 2-D Transform Hierarchy

The generic form for a two-dimensional (2-D) wavelet transform is shown in Figure.

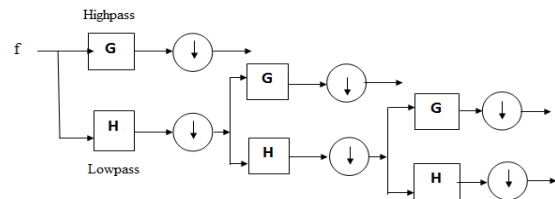


Figure: 2D Wavelet Decomposition

The 1-D wavelet transform can be extended to a two-dimensional (2-D) wavelet transform using separable wavelet filters. With separable filters the 2-D transform can be computed by applying a 1-D transform to all the rows of the input, and then repeating on all of the columns.

LL1	HL1
LH1	HH1

Figure: Sub band Labeling Scheme for a one level, 2-D Wavelet Transform

The original image of a one-level (K=1), 2-D wavelet transform, with corresponding notation is shown in the above figure. The example is repeated for a three-level (K =3) wavelet expansion in the below figure. In all of the discussion K represents the highest level of the decomposition of the wavelet transform.

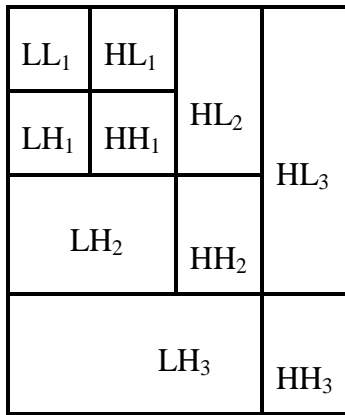


Figure: Sub-band labeling Scheme for a Three Level, 2-D Wavelet Transform

The 2-D sub-band decomposition is just an extension of 1-D sub-band decomposition. The entire process is carried out by executing 1-D sub-band decomposition twice, first in one direction (horizontal), then in the orthogonal (vertical) direction. For example, the low-pass sub-bands (Li) resulting from the horizontal direction is further decomposed in the vertical direction, leading to LLi and LHi sub-bands. Similarly, the high pass sub-band (Hi) is further decomposed into HLi and HHi. After one level of transform, the image can be further decomposed by applying the 2-D sub-band decomposition to the existing LLi sub-band. This iterative process results in multiple “transform levels”. In Fig. 2.14 the first level of transform results in LH1, HL1, and HH1, in addition to LL1, which is further decomposed into LH2, HL2, HH2, LL2 at the second level, and the information of LL2 is used for the third level transform. The sub-band LLi is a low-resolution sub-band and high-pass sub-bands LHi, HLi, HHi are horizontal, vertical, and diagonal sub-band respectively since they represent the horizontal, vertical, and diagonal residual information of the original image.

F. Probabilistic Neural Networks (PNN):

Probabilistic (PNN) and General Regression Neural Networks (GRNN) have similar architectures, but there is a fundamental difference: Probabilistic networks perform classification where the target variable is categorical, whereas general regression neural networks perform regression where the target variable is continuous. If you select a PNN/GRNN network, DTREG will

automatically select the correct type of network based on the type of target variable [3].

G. Architecture of a PNN

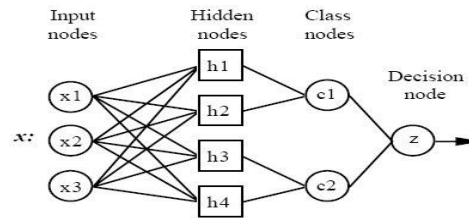


Figure: Architecture of a PNN

All PNN networks have four layers:

1. **Input layer** — There is one neuron in the input layer for each predictor variable. In the case of categorical variables, $N-1$ neurons are used where N is the number of categories. The input neuron (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the interquartile range. The input neurons then feed the values to each of the neurons in the hidden layer [3].
2. **Hidden layer** — This layer has one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. When presented with the x vector of input values from the input layer, a hidden neuron computes the Euclidean distance of the test case from the neuron's center point and then applies the RBF kernel function using the sigma value(s). The resulting value is passed to the neurons in the pattern layer [3].
3. **Pattern layer / Summation layer** — The next layer in the network is different for PNN networks and for GRNN networks. For PNN networks there is one pattern neuron for each category of the target variable. The actual target category of each training case is stored with each hidden neuron; the weighted value coming out of a hidden neuron is fed only to the pattern neuron that corresponds to the hidden neuron's category. The pattern neurons add the values for the class they represent (hence, it is a weighted vote for that category) [3].

For GRNN networks, there are only two neurons in the pattern layer. One neuron is the denominator summation unit the other

is the numerator summation unit. The denominator summation unit adds up the weight values coming from each of the hidden neurons. The numerator summation unit adds up the weight values multiplied by the actual target value for each hidden neuron.

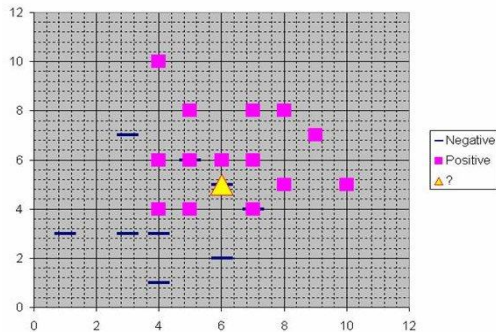
4. **Decision layer** — The decision layer is different for PNN and GRNN networks.

For PNN networks, the decision layer compares the weighted votes for each target category accumulated in the pattern layer and uses the largest vote to predict the target category.

For GRNN networks, the decision layer divides the value accumulated in the numerator summation unit by the value in the denominator summation unit and uses the result as the predicted target value[4].

H. How PNN network work

Although the implementation is very different, probabilistic neural networks are conceptually similar to *K-Nearest Neighbor* (k-NN) models. The basic idea is that a predicted target value of an item is likely to be about the same as other items that have close values of the predictor variables [3]. Consider this figure:



Assume that each case in the training set has two predictor variables, x and y . The cases are plotted using their x,y coordinates as shown in the figure. Also assume that the target variable has two categories, *positive* which is denoted by a square and *negative* which is denoted by a dash. Now, suppose we are trying to predict the value of a new case represented by the triangle with predictor values $x=6$, $y=5.1$. Should we predict the target as positive or negative?

Notice that the triangle is position almost exactly on top of a dash representing a negative

value. But that dash is in a fairly unusual position compared to the other dashes which are clustered below the squares and left of center. So it could be that the underlying negative value is an odd case.

The nearest neighbor classification performed for this example depends on how many neighboring points are considered. If 1-NN is used and only the closest point is considered, then clearly the new point should be classified as negative since it is on top of a known negative point. On the other hand, if 9-NN classification is used and the closest 9 points are considered, then the effect of the surrounding 8 positive points may overbalance the close negative point.

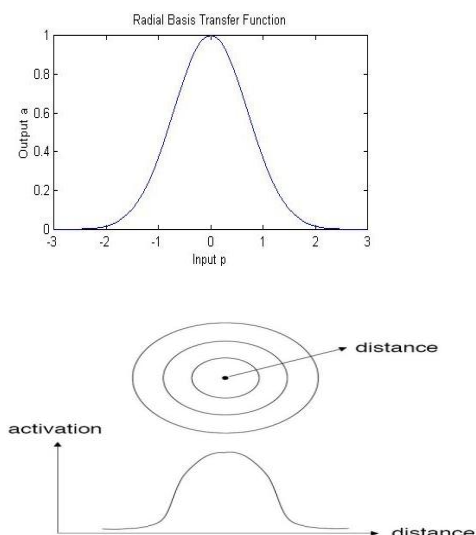
A probabilistic neural network builds on this foundation and generalizes it to consider all of the other points. The distance is computed from the point being evaluated to each of the other points, and a *radial basis function* (RBF) (also called a *kernel function*) is applied to the distance to compute the weight (influence) for each point. The radial basis function is so named because the radius distance is the argument to the function [9].

$$\text{Weight} = \text{RBF}(\text{distance})$$

The further some other point is from the new point, the less influence it has.

Radial Basis Function

Different types of radial basis functions could be used, but the most common is the Gaussian function:



CONCLUSION

This study was undertaken to develop an PNN to classify stage of the brain tumor images and detect the Tumor using clustering technique .grey level index values were assigned to the pixels of the indexed image and used as PNN inputs. There were 15 images, for training, and 8 images for testing. Probabilistic Neural Network with image and data processing techniques was employed to implement an automated Brain Tumor classification [7]. Decision making was performed in two stages: feature extraction using GLCM and the classification using Probabilistic Neural Network (PNN). This paper presents a segmentation method, K-Means clustering algorithm, for segmenting Magnetic Resonance images to detect the Brain Tumor in its early stages. Although the study was limited by the available computational resources and training data, the results indicate the potential of ANNs for fast image recognition and classification. Fast image recognition and classification can be useful in the control of real-world, site-specific herbicide application.

- The paper has been appreciated by all the users in the organization.
- It is easy to use, since it uses the GUI provided in the user dialog.
- User friendly screens are provided.
- It also provides the user with variable options in customizing the packet capture.
- It has been thoroughly tested and implemented.

The presented samples demonstrate that the initial aim of the library was achieved - it is flexible, reusable, and it is easy to use it for different tasks. Although, there is still much work to do, because of a great range of different neural network architectures and their learning algorithms, but still - the library can be used for many different problems, and can be extended to solve even more[7]. I hope the library will become useful not only in my further research work, but other different researchers will find it interesting and useful.

REFERENCES

- [1]. N. Kwak, and C. H. Choi, "Input Feature Selection for Classification Problems", IEEE Transactions on Neural Networks, 13(1), 143–159, 2002.
- [2].E. D. Ubeyli and I. Guler, "Feature Extraction from Doppler Ultrasound Signals for Automated Diagnostic Systems", Computers in Biology and Medicine, 35(9), 735–764, 2005.
- [3].D.F. Specht, "Probabilistic Neural Networks for Classification, mapping, or associative memory", Proceedings of IEEE International Conference on Neural Networks, Vol.1, IEEE Press, New York, pp. 525-532, June 1988.
- [4].D.F. Specht, "Probabilistic Neural Networks" Neural Networks, vol. 3, No.1, pp. 109-18, 1990.
- [5].Georgiadis. Et ali, "Improving brain tumor characterization on MRI by probabilistic neural networks and non-linear transformation of textural features", Computer Methods and program in biomedicine, vol 89, pp24-32, 2008
- [6] Kaus M., "Automated segmentation of MRI brain tumors", *Journal of Radiology*, vol.218, pp. 585-591, 2001
- [7] Kornel, P., Bela, M., Rainer, S., Zalan, D., Zsolt, T. and Janos, F., "Application of neural network in medicine", *Diag. Med. Tech.*, vol. 4, issue 3, pp: 538-54, 1998.
- [8] Messen W, Wehrens R, Buydens L, "Supervised Kohonen networks for classification problems", *Chemometrics and Intelligent Laboratory Systems*, vol.83, pp:99-113, 2006.
- [9] Orr M.J.L., Hallam J., Murray A., and Leonard .T, "Assessing rbf networks using delve," *International Journal of Neural Systems*, vol. 10, issue 5, pp. 397-415, 2000.
- [10] Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". *Journal of the Royal Statistical Society, Series*