**Automated MRI Image Segmentation For tumour Recognition**

**A PROJECT REPORT**

***Submitted by***

Y.RAVI KANTH 314126510113

T.KRISHNA VENI 314126510124

B.MOUNICA 314126510138

T.MANISH 314126510099

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**



Under esteemed guidance of

**Mr.P.Naga Srinivasu** (Associative Professor)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Affiliated to Andhra University)

**SANGIVALASA, VISAKHAPATNAM – 531162**

2017-2018

**Automated MRI Image Segmentation For tumour Recognition**

**A PROJECT REPORT**

***Submitted by***

Y.RAVI KANTH 314126510113

T.KRISHNA VENI 314126510124

B.MOUNICA 314126510138

T.MANISH 314126510099

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**



Under esteemed guidance of

**Mr.P.Naga Srinivasu** (Associative Professor)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Affiliated to Andhra University)

**SANGIVALASA, VISAKHAPATNAM – 531162**

2017-2018

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM-531162**

**BONAFIDE CERTIFICATE**

Certified that this project report “**Automated MRI image Segmentation for tumour** **recognition”** is the bonafide work “Y.RAVIKANTH(314126510113), T.KRISHNAVENI(314126510134), B.MOUNICA(314126510138), T.MANISH (314126510099)” who carried out the project work under my supervision.

**DR.R.SIVARANJANI MR.P.NAGASRINIVASU**

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

Professor Associate Professor

Department of Computer Science Department of Computer Science

& Engineering & Engineering

ANITS ANITS

**DECLARATION**

This is to certify that the project work entitled “**AUTOMATED MRI IMAGE SEGMENATION FOR TUMOUR RECOGNITION**” is a bonafide work carried out by **Y.RAVIKANTH, T.KRISHNAVENI, B.MOUNICA, T.MANISH**  as a part of **B.TECH** final year 2nd semester of **computer science Engineering** of Andhra University, Visakhapatnam during the year 2017-18.

We, **Y.RAVIKANTH, T.KRISHNAVENI, B.MOUNICA, T.MANISH,** of final semester B.Tech., in the department of Computer Science Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled  **AUTOMATED MRI IMAGE SEGMENTATION FOR TUMOUR RECOGNITION** is carried out by us and submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering** , under Anil Neerukonda Institute of Technology & Sciences during the academic year 2017-18 and has not been submitted to any other university for the award of any kind of degree.

Y.RAVI KANTH 314126510113

T.KRISHNA VENI 314126510124

B.MOUNICA 314126510138

T.MANISH 314126510099

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement always boosted the morale. We take a great pleasure in presenting a project, which is the result of a studied blend of both research and knowledge.

We first take the privilege to thank the Head of our Department, **Dr.R.SIVARANJANI**, for permitting us in laying the first stone of success and providing the lab facilities, we would also like to thank the other staff in our department and lab assistants who directly or indirectly helped us in successful completion of the project.

We feel great to thank **Mr. P. NAGA SRINIVASU**, who are our project guides and who shared their valuable knowledge with us and made us understand the real essence of the topic and created interest in us to work day and night for the project; we also thank our Project coordinators **Mr K .Suresh &Mrs .T .Kranthi**, for their support and encouragement.

We also like to thank **Mr.P.Sainath Vital** for providing us the lab resources; we also thank our friends and college staff who extended their part of support in the successful completion of the project.

Y.RAVI KANTH 314126510113

T.KRISHNA VENI 314126510124

B.MOUNICA 314126510138

T.MANISH 314126510099

**ABSTRACT**

We propose adaptive brain tumour detection technique, Image processing is used for detection of tumour, only MRI images are not able to identify the region .We are using Automated segmentation through multi-Objective by pre-processing of image, By which de-noising is performed over the original image by applying Median filter. Also we are using Genetic algorithm for optimization. It is expected that the experimental results of the proposed system will give better result in comparison to other existing systems.

**Keywords:** Magnetic Resonance Imaging(MRI), Brain Tumour, segmentation.

**CONTENTS**

**TITLE Page No.**

Abstract i

Keywords ii

**1. Introduction**

1.1 Problem statement 1

1.2 Motivation 2

1.3 Contribution 2

1.4 Research Methodology 3

**2. Literature Survey**

2.1 MR images 4

2.2 Image Segementation 6 2.3 Techniques in Image Segmentation 8

2.4 Genetic Algorithm 10

2.5WORKING OF GA 15

WITH AUTOMATED SEGMENTATION

**3. System Software Requirements**

3.1 Software Requirements 16

3.2 Hardware Requirements 16 3.2.1 Hardware Interface 16 3.2.2 Software Interface 16

**TITLE Page No.**

**4. Existing techniques**

4.1 K-means clustering 17

4.2 Fuzzy c-means 19

4.3 KNN queries 20

4.4 Seeded growing 21

4.5 Failures of existing system 22

**5. Proposed System**

5.1 Architecture 23

5.2 Processing Steps 24

5.3 Effectiveness of proposed system 27

**6. Methodologies**

6.1 Noise Removal 31

6.2 Image Resizing 32

6.3 Genetic Algorithm 34

6.4 Skull Stripping 37

**7. Implementation**

7.1 Sample Code for Segmentation 41 7.2 Sample Code for Analysis 43

**8. Results**  50

**9. Conclusion** 55

**References**  56

**1.INTRODUCTION**

**1.1 PROBLEM STATEMENT**

In the space of image processing, the processing of medical images for medical diagnostics is the prime area of research for many decade and image processing plays key role in the health care. Brain Tumour is hysterical expansion of cancer cells and varied types of brain tumour with different characteristics and treatments . A brain tumour is formed because of abnormal cells created within the brain and brain tumour is primarily classified into two types such as benign tumours and malignant or cancerous tumours. Cancerous tumours further divided into two types' primary tumours that start within the brain and secondary tumours, brain metastasis which is spread from somewhere else in the body . In the field of medical, brain tumour grows without any control of typical forces, with the advancement of medical imaging; imaging modalities gain significant part in the brain tumour assessment and huge impact on patient concern.Image segmentation is one of the rudimentary technique in drawing out the contour of the image by which recognition of the objects in the image is made less complicated. In common, the evaluation of segments involving user assistance, approaches involving supervised evaluation and subjective evaluation of the segments. Apart from the challenges of supervised based approach, it is obligatory to automate the segmentation that minimized the burden of the radiologist and more over the unsupervised based approach tends to generate accurate results as it work with a dynamic range of an image.

**1.2 MOTIVATION**

* Genetic algorithm is basically a generate-and-test beam search algorithm. Motivational factors for implementing genetic algorithm are:

• Genetic algorithms are a robust adaptive optimization technique based on a biological paradigm

* They perform efficient search on poorly-defined spaces by maintaining an ordered pool of strings that represent regions in the search space.
* Evolution is known to be successful.
* GAs can search hypotheses containing complex interacting parts.

It is easily parallelizable.

**1.3CONTRIBUTION**

In our work, the Genetic algorithm is used for segmenting the brain which gives optimal results when compared to other segmentation techniques. The purpose of this project is to extract the exact location, size and shape of the tumour from an MRI scanned image of the brain. We have experimented on various brain MRI images which had different sizes and locations of the tumour and verified the results.

**1.4 RESEARCH METHODOLOGY:**

The cause of Brain tumors is unknown. Uncommon risk factors include inherited neurofibromatosis, exposure to vinyl chloride, Epstein–Barr virus, and ionizing radiation and astrocytoma such as glioblastomas. In children, the most common type is a malignant medullo blastoma. Diagnosis is usually by medical examination along with computed tomography or magnetic resonance imaging. This is then often confirmed by a biopsy. Based on the findings, the tumors are divided into different grades of severity.We detect the accurate part of tumour so that it would be helpful for curing it.

**2. LITERATURE SURVEY**

**2.1MR IMAGES:**

Magnetic resonance imaging is a medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body in both health and disease. MRI scanners use strong magnetic fields, electric field gradients, and radio waves to generate images of the organs in the body. MRI does not involve X-rays and the use of ionizing radiation, which distinguishes it from CT or CAT scans. Magnetic resonance imaging is a medical application of nuclear magnetic resonance (NMR). NMR can also be used for imaging in other NMR applications such as NMR spectroscopy.

While the hazards of X-rays are now well-controlled in most medical contexts, MRI may still be seen as a better choice than CT. MRI is widely used in hospitals and clinics for medical diagnosis, staging of disease and follow-up without exposing the body to radiation. However, MRI may often yield different diagnostic information compared with CT. There may be risks and discomfort associated with MRI scans. Compared with CT scans, MRI scans typically take longer and are louder, and they usually need the subject to enter a narrow, confining tube. In addition, people with some medical implants or other non-removable metal inside the body may be unable to undergo an MRI examination safely.

MRI was originally called 'NMRI' (nuclear magnetic resonance imaging) and is a form of NMR, though the use of 'nuclear' in the acronym was dropped to avoid negative associations with the word. Certain atomic nuclei are able to absorb and emit radio frequency energy when placed in an external magnetic field. In clinical and research MRI, hydrogen atoms are most often used to generate a detectable radio-frequency signal that is received by antennas in close proximity to the anatomy being examined. Hydrogen atoms exist naturally in people and other biological organisms in abundance, particularly in water and fat. For this reason, most MRI scans essentially map the location of water and fat in the body. Pulses of radio waves excite the nuclear spin energy transition, and magnetic field gradients localize the signal in space. By varying the parameters of the pulse sequence, different contrasts may be generated between tissues based on the relaxation properties of the hydrogen atoms therein.

Since its development in the 1970s and 1980s, MRI has proven to be a highly versatile imaging technique. While MRI is most prominently used in diagnostic medicine and biomedical research, it also may be used to form images of non-living objects. MRI scans are capable of producing a variety of chemical and physical data, in addition to detailed spatial images. The sustained increase in demand for MRI within health systems has led to concerns about cost effectiveness and overdiagnosis.

2.2 **IMAGE SEGMENTATION:**

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). 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 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 are 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.

**2.3TECHNIQUES IN IMAGE SEGMENTATION:**

MRI Segmentation Mechanisms

Semi-Automated Techniques

Automatic Techniques

* K means
* Fuzzy C means
* Seeded Region Growing
* KNN Mechanism
* Genetic Algorithm Based
* Water flow like lgorithm
* Gravitaional search

**2.4GENETIC ALGORITHM**:

* GA’s simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.
* GAs are based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations:
* Individuals in a population compete for resources and mates.
* Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
* Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
* Thus each successive generation will become more suited to their environment.

**CROSS OVER IN GENETIC ALGORITHM**:

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Crossover is a process of taking more than one parent solution and producing a child solution from them. There are methods for selection of the chromosomes. Those are also given below.

**Single-point**

A single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children:

**Two-point**

Two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms:

**Uniform and half uniform**

The uniform crossover uses a fixed mixing ratio between two parents. Unlike single- and two-point crossover, the uniform crossover enables the parent chromosomes to contribute the gene level rather than the segment level.

If the mixing ratio is 0.5, the offspring has approximately half of the genes from first parent and the other half from second parent, although cross over points can be randomly chosen as seen below:

The uniform crossover evaluates each bit in the parent strings for exchange with a probability of 0.5. Empirical evidence suggest that it is a more exploratory approach to crossover than the traditional exploitative approach that maintains longer schemata. This results in a more complete search of the design space with maintaining the exchange of good information. Unfortunately, no satisfactory theory exists to explain the discrepancies between the uniform crossover and the traditional approaches.

In the uniform crossover scheme (UX) individual bits in the string are compared between two parents. The bits are swapped with a fixed probability, typically 0.5.

In the half uniform crossover scheme (HUX), exactly half of the nonmatching bits are swapped. Thus first the Hamming distance (the number of differing bits) is calculated. This number is divided by two. The resulting number is how many of the bits that do not match between the two parents will be swapped.

**Three parent**

This section needs expansion. You can help by adding to it. (June 2013)

In this technique, the child is derived from three randomly chosen parents. Each bit of the first parent is compared with the same bit of the second parent. When these bits are the same it is used in the offspring, otherwise the bit from the third parent is used in the offspring. For example, the following three parents:

p1 110100010

p2 011001001

p3 110110101

will produce the following offspring:

op1p2p3 110100001

For ordered chromosomes

Depending on how the chromosome represents the solution, a direct swap may not be possible. One such case is when the chromosome is an ordered list, such as an ordered list of the cities to be travelled for the traveling salesman problem. There are many crossover methods for ordered chromosomes. The already mentioned N-point crossover can be applied for ordered chromosomes also, but this always needs a corresponding repair process, actually, some ordered crossover methods are derived from the idea. However, sometimes a crossover of chromosomes produces recombinations which violate the constraint of ordering and thus need to be repaired. Several examples for crossover operators (also mutation operator) preserving a given order are given in partially matched crossover (PMX): In this method, two crossover points are selected at random and PMX proceeds by position wise exchanges. The two crossover points give matching selection. It affects cross by position-by-position exchange operations. In this method parents are mapped to each other, hence we can also call it partially mapped crossover.

cycle crossover (CX): Beginning at any gene {\displaystyle i} i in parent 1, the {\displaystyle i} i-th gene in parent 2 becomes replaced by it. The same is repeated for the displaced gene until the gene which is equal to the first inserted gene becomes replaced (cycle).

order crossover operator (OX1): A portion of one parent is mapped to a portion of the other parent. From the replaced portion on, the rest is filled up by the remaining genes, where already present genes are omitted and the order is preserved.

order-based crossover operator (OX2)

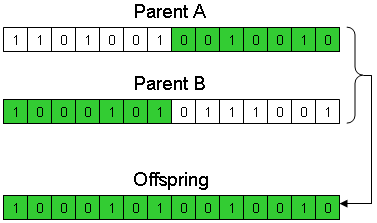
position-based crossover operator (POS)

voting recombination crossover operator (VR)

alternating-position crossover operator (AP)

sequential constructive crossover operator (SCX)

Other possible methods include the edge recombination operator.



**MUTATION IN GENETIC ALGORITHM:**

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. This mutation procedure, based on the biological point mutation, is called single point mutation. Other types are inversion and floating point mutation. When the gene encoding is restrictive as in permutation problems, mutations are swaps, inversions, and scrambles.

The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

For different genome types, different mutation types are suitable:

Bit string mutation

The mutation of bit strings ensue through bit flips at random positions.

Example:

1 0 1 0 0 1 0

↓

1 0 1 0 1 1 0

The probability of a mutation of a bit is {\displaystyle {\frac {1}{l}}} {\frac {1}{l}}, where {\displaystyle l} l is the length of the binary vector. Thus, a mutation rate of {\displaystyle 1} 1 per mutation and individual selected for mutation is reached.

**Flip Bit**

This mutation operator takes the chosen genome and inverts the bits (i.e. if the genome bit is 1, it is changed to 0 and vice versa).

**Boundary**

This mutation operator replaces the genome with either lower or upper bound randomly. This can be used for integer and float genes.

**Non-Uniform**

The probability that amount of mutation will go to 0 with the next generation is increased by using non-uniform mutation operator. It keeps the population from stagnating in the early stages of the evolution. It tunes solution in later stages of evolution. This mutation operator can only be used for integer and float genes.

**Uniform**

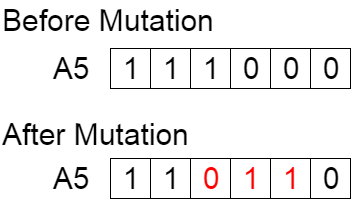
This operator replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene. This mutation operator can only be used for integer and float genes.

**Gaussian**

This operator adds a unit Gaussian distributed random value to the chosen gene. If it falls outside of the user-specified lower or upper bounds for that gene, the new gene value is clipped. This mutation operator can only be used for integer and float genes.

**Shrink**

This operator adds a random number taken from a Gaussian distribution with mean equal to the original value of each decision variable characterizing the entry parent vector.]



**2.5WORKING OF GA WITH AUTOMATED SEGMENTATION :**

This is how automated segmentation works with GA

Proceed through iteration tll stropping criteria is attained

Input image

Resulatant segmented image

Image pre-processing for noise removal/contrast enhancement

Apply cross over and mutation

Assigns the pixels to the segment on the distance measure

Estimating the segments centroid

Randomly Choosing the initial number of segments



**3. SYSTEM REQUIREMENT SPECIFICATION**

**3.1 SOFTWARES USED IN THIS PROJECT:**

**Language**: GNU Octave 4.2.1

**OPERATING SYSTEM:** Windows 10

**3.2 HARDWARES USED IN THIS PROJECT:**

**PROCESSOR**: intel Multi Core processor

**RAM**: 2 GB or above

**HARDDISK**:500 GB or above

**3.2.1 HARDWARE INTERFACE:**

**MONITOR:** the outputs are displayed on the monitor screen.

**3.2.2 SOFTWARE INTERFACE:**

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

**4. EXISTING TECHNIQUES**

Detection of brain tumour is done in many ways. The existing systems are :

**4.1Using k-means clustering:**

K-Means is a least-squares partitioning method that divide a collection of objects into K groups. The algorithm iterates over two steps:

Compute the mean of each cluster.

Compute the distance of each point from each cluster by computing its distance from the corresponding cluster mean. Assign each point to the cluster it is nearest to.

Iterate over the above two steps till the sum of squared within group errors cannot be lowered any more.

The initial assignment of points to clusters can be done randomly. In the course of the iterations, the algorithm tries to minimize the sum, over all groups, of the squared within group errors, which are the distances of the points to the respective group means. Convergence is reached when the objective function (i.e., the residual sum-of-squares) cannot be lowered any more. The groups obtained are such that they are geometrically as compact as possible around their respective means. Using the set of feature images, a feature vector is constructed corresponding to each pixel ($[e\_1(a,b)$, $e\_2(a,b)$, ... ,$e\_d(a,b)]$), where d is the number of feature images used for the segmentation process. The K-Means can then be used to segment the image into three clusters - corresponding to two scripts and background respectively. For each additional script, one more cluster is added. Here, each feature is assigned a different weight, which is calculated based on the feature importance as described in the previous Section. The distance between two vectors is computed using Equation 19. Once the image has been segmented using the K-Means algorithm, the clustering can be improved by assuming that neighboring pixels have a high probability of falling into the same cluster. Thus, even if a pixel has been wrongly clustered, it can be corrected by looking at the neighboring pixels.

** **

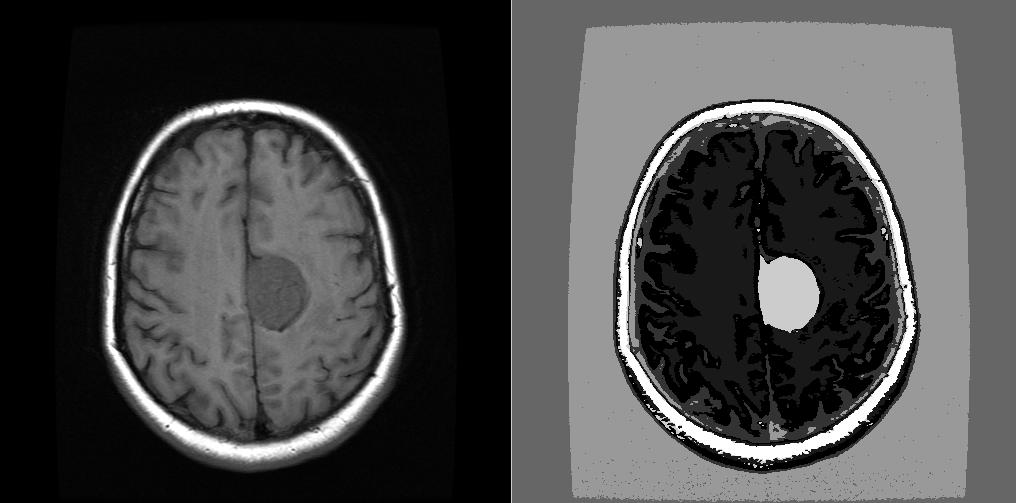
**Before segmentation After segmentation**

**Problem with K-means:**

* K means is sensitive to initial k value that may lead to over segmentation or under segmentation.

**4.2Using fuzzy c –means:**

The Fuzzy C-Means (FCM) clustering algorithm was first introduced by Dunn [ and later was extended by Bezdek . The algorithm is an iterative clustering method that produces an optimal c partition by minimizing the weighted within group sum of squared error objective function JF CM : JF CM = Xn k=1 Xc i=1 (uik) q d 2 (xk, vi) where X = {x1, x2, · · · , xn} ⊆ Rp is the data set in the p-dimensional vector space, n is the number of data items, c is the number of clusters with 2 ≤ c < n, uik is the degree of membership of xk in the i th cluster, q is a weighting exponent on each fuzzy membership, vi is the prototype of the centre of cluster i, d 2 (xk, vi) is a distance 20 Y. Yang, S. Huang measure between object xk and cluster centre vi . A solution of the object function JF CM can be obtained via an iterative process, which is carried out as follows: 1. Set values for c, q and ǫ. 2. Initialize the fuzzy partition matrix U = [uik]. 3. Set the loop counter b = 0. 4. Calculate the c cluster centers n v (b) i o with U (b) : v (b) i = Pn k=1 u (b) ik q xk Pn k=1 u (b) ik q (2) 5. Calculate the membership U (b+1). For k = 1 to n, calculate the following: Ik = {i |1 ≤ i ≤ c, dik = kxk − vik = 0}, 6I; for the k th column of the matrix, compute new membership values: (a) if Ik = φ, then u (b+1) ik = 1 Pc j=1 dik djk 2 (q−1) , (3) (b) else u (b+1) ik = 0 for all i 6∈ I and P i∈Ik u (b+1) ik = 1; next k. 6. If U (b) − U (b+1) < ǫ, stop; otherwise, set b = b + 1 and go to step 4. 2.2 Neighborhood EM Algorithm In order to incorporate the spatial dependence into the objects, a modified version of the conventional expectation maximization (EM) algorithm has been proposed in . In this approach, the maximum likelihood criterion is penalized by a term that quantifies the degree of spatial contiguity of the pixels supporting the respective components of the probability density function (pdf) model. The spatial structure of a given data set is defined by using matrix W = (wjk): wjk = ( 1 if xj and xk are neighbors and j 6= k, 0 otherwise. The following term is then used for regularizing the maximum likelihood criterion G (c) = 1 2 Xn j=1 Xn k=1 Xc i=1 cij cikwjk, Image Segmentation by FCM Clustering Algorithm 21 where c is the number of classes and cij is the probability that xj belongs to class i. This term characterizes the homogeneity level of the partition. The more the classes contain adjacent elements, the greater this term is. The new criterion of the NEM algorithm is obtained by optimizing the weighted sum of two terms U (c, φ) = D (c, φ) + βG (c) , (6) where D (c, φ) is the log-likelihood function of EM algorithm, β > 0 is a fixed coefficient. Details about NEM can be found in [15]. This algorithm is maximized to get the optimum results just as the same structure as the PersonNameProductIDEM algorithm. SuccessfulEM algorithm. Successful results have been reported for image segmentation using this algorithm.

****

* How ever fuzzy c means that work on the basis of membership also suffer with initial number of clusters and consumes more time to converge towards best possible number of segments,how ever this doesn’t exhibit a better performance with more dimensions(features).

**4.3Using knn Queries( k-Nearest Neighbors):**

One is contour-based and the other one is region–based Methods. First category use discontinuity in an image to detect edges or contours in the image, and then use them to partition the image. Methods of the second category try to divide pixels in an image into different groups corresponding to coherent properties such as color etc., that is, it mainly use decision criteria to segment an image into different regions according to the similarity of the pixels. By regarding the image segmentation as the problem of partitioning pixels into different clusters according to their color similarity and spatial relation, we propose our color image segmentation method. It is a region-based method. According to the method, pixels in each segmented region should be connective in spatial and similar in color.

**Problems with KNN Queries:**

* KNN is a distance measure based segmentation of image that might lead to a inaccurate assignment of pixels to the centroids

**4.4 Seeded Growing technique:**

 Conventional image segmentation techniques using region growing requires initial seeds selection, which increases computational cost & execution time. To overcome this problem, a single seeded region growing technique for image segmentation is proposed, which starts from the center pixel of the image as the initial seed. It grows region according to the grow formula and selects the next seed from connected pixel of the region. We use intensity based similarity index for the grow formula and Otsu's Adaptive thresholding is used to calculate the stopping criteria for the grow formula. We apply the proposed method to the Berkley segmentation database images and discuss results based on Liu's F-factor that shows efficient segmentation.

**Problems with Sedded Growing Technique:**

* Seeded growing mechanism is sensitive for initial data points in case of choosing a wrong centers may lead to inappropriate segmentation.

**4.5DISADVANTAGES OF EXISTING SYSTEM:**

* K means is sensitive to initial k value that may lead to over segmentation or under segmentation.
* How ever fuzzy c means that work on the basis of membership also suffer with initial number of clusters and consumes more time to converge towards best possible number of segments,how ever this doesn’t exhibit a better performance with more dimensions(features).
* Seeded growing mechanism is sensitive for initial data points in case of choosing a wrong centers may lead to inappropriate segmentation.
* KNN is a distance measure based segmentation of image that might lead to a inaccurate assignment of pixels to the centroids

**5. PROPOSED SYSTEM**

In this we propose a fully automatic brain tissue classification from magnetic resonance images (MRI) The accurate segmentation of MR images into different tissue classes, especially gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF) is a tough task so we use automated segmentation to divide the image into segments and use genetic algorithm to optimize the segmented image and to get the grey matter(GM) position ie the tumour region. Image segmentation can be pursued by many different ways. One of them is called multi-thresholding. Since we want to segment image to more than two segments (more than one threshold) we need to determine at least two thresholds. If we want to segment gray level image with 256 levels to three segments we are likely supposed to examine 256\*255 = 65280 different threshold combinations. This brute force approach would end up with optimal solution, however computing time would be definitely high. Genetic algorithm searches space containing all possible solutions and obtain the best solution among all examined in much less time than brute force algorithm.

**Implementation:**

Genetic algorithm was implemented in GNU Octave.

This language was selected due to its fast prototyping.

**Initialization:**

This part of code is found at \*ga\_segmentation.m\*.

There are few settings which can affect the result of algorithm.

n\_population:

size of population; contains different solutions

n\_iterations:

number of iterations; algorithm terminates after all iterations are done

n\_thresholds:

number of desired thresholds; n\_thresholds = (number of segments - 1)

The next three settings are related to each other.

They are ratios of selection (n\_selection), crossover (n\_crossover) and mutation (n\_mutation).

The sum of all these parameters has to be equal to 1.

This condition guarantees the size of population does not change between iterations.

Single chromosome is implemented as vector of binary numbers.

The length of vector is L, n, where L denotes ‘log(number of gray levels)’ and n is the number of desired thresholds.

Population is represented a matrix, where each row is single chromosome and number of rows corresponds to size of population.

An initial population was randomly generated.

In this part we also load an image for segmentation.

Algorithm works only with grayscale images, so we always convert each image at the beginning before we start to work with it.

**Evaluation of fitness:**

According to paper, which was mentioned above, evaluation of fitness should be performed with ratio between between inter-object variance and intra-object variance.

We have decided for slightly easier evaluation based on sum of intra-object variance.

The solution with the lowest sum is the most accurate.

**Selection:**

Current solution selects to new generation only the best solutions.

The number of forwarded solution depends on ratio n\_selection.

**Crossover:**

Chromosomes, which are supposed to crossover, are chosen from two vectors containing randomly permutated indexes (numbers of row) of chromosomes.

The number chromosomes, which crossover, depends on ratio n\_crossover.

One-point crossover is employed in current solution.

The point of crossover is randomly generated with uniform distribution.

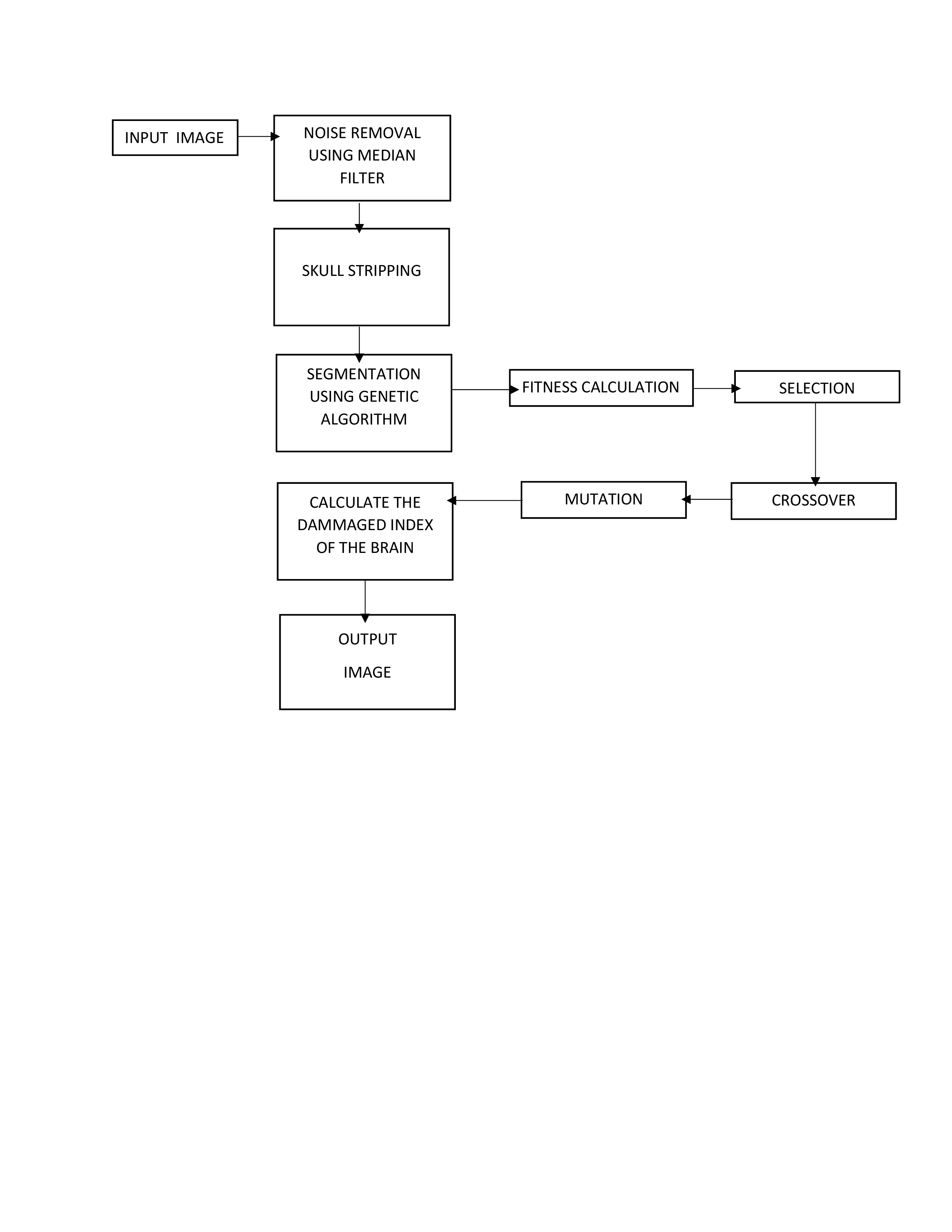
**Mutation:**

Indexes (number of rows) of chromosomes are randomly permutated and the very first of them are more likely to be chosen.

The number of chromosomes, which happens to be chosen for mutation, depends on ratio n\_mutation.

Current solution allows to mutate only one gene of chromosome.

**5.1 ARCHITECTURE:**

****

**5.1.1 A Clear View Of Proposed Architecture**

**Pixels are classified and rearranged based on laplacian threshold**

**Image preprocessing through harmonic mean**

**Original Image rendered from MRI scanner**

**Terminating**

**touchstone**

**Attained**

**NO**

**Improvised genetic Algorithm is applied over initial classes**

**Pixels are arranged in descending order of their fitness value**

**YES**

**NO**

**Terminating**

**touchstone**

**Attained**

**Resultant image is fed as input for SGO**

**Final Segmented image with estimated damaged area is populated**

**YES**

**5.2 Processing Steps:**

1. We take a MR image as an input and apply noise removal filters so as to remove noise from the image .
2. After removing noise we strip of the skull part from the brain MRI
3. Now we apply segmentation based on genetic algorithm .Later we calculate fitness function.
4. As fitness function ranks best to worse segmented part we select best off spring for mutation and cross over.
5. After the segmentation we calculate the damaged index of the segmented part.
6. And we get our output of the damaged part of the brain.

**5.3 Why Proposed System Is Better Than Existing System:**

**PSNR vs MSE Comparison Chart:**

peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most easily defined via the mean squared error (*MSE*). Given a noise-free *m*×*n* monochrome image *I* and its noisy approximation *K*, *MSE* is defined as:

{\displaystyle {\mathit {MSE}}={\frac {1}{m\,n}}\sum \_{i=0}^{m-1}\sum \_{j=0}^{n-1}[I(i,j)-K(i,j)]^{2}}

The PSNR (in dB) is defined as:

{\displaystyle {\begin{aligned}{\mathit {PSNR}}&=10\cdot \log \_{10}\left({\frac {{\mathit {MAX}}\_{I}^{2}}{\mathit {MSE}}}\right)\\&=20\cdot \log \_{10}\left({\frac {{\mathit {MAX}}\_{I}}{\sqrt {\mathit {MSE}}}}\right)\\&=20\cdot \log \_{10}\left({{\mathit {MAX}}\_{I}}\right)-10\cdot \log \_{10}\left({\mathit {MSE}}\right)\end{aligned}}}Here, *MAXI* is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear PCM with *B* bits per sample, *MAXI* is 2B−1.

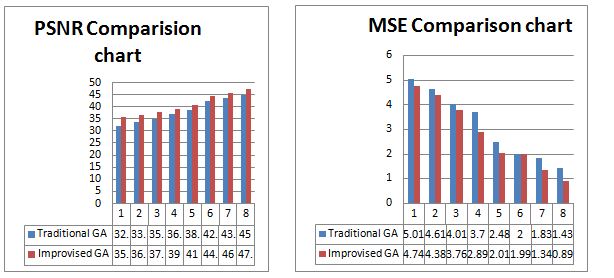
### Application in color images:

For color images with three RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. Alternately, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space, e.g., YCbCr or HSL.

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors or deviations—that is, the difference between the estimator and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance of the estimator. Like the variance, MSE has the same units of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields the root-mean-square error or root-mean-square deviation (RMSE or RMSD), which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard deviation.

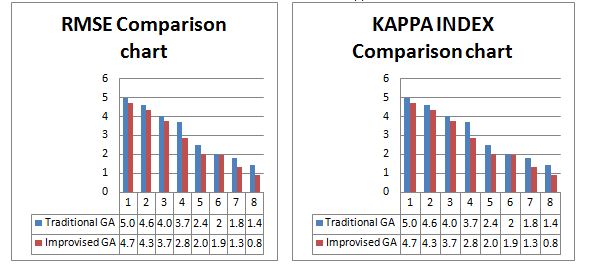
****

**RMSE vs KAPPA INDEX COMPARISON:**

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) (or sometimes root-mean-squared error) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed. The RMSD represents the sample standard deviation of the differences between predicted values and observed values. These individual differences are called residuals when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSD is a measure of accuracy, to compare forecasting errors of different models for a particular data and not between datasets, as it is scale-dependent.

RMSD is the square root of the average of squared errors. The effect of each error on RMSD is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSD. Consequently, RMSD is sensitive to outliers.

Cohen's kappa coefficient (κ) is a statistic which measures inter-rater agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation, as κ takes into account the possibility of the agreement occurring by chance. There is controversy surrounding Cohen’s Kappa due to the difficulty in interpreting indices of agreement. Some researchers have suggested that it is conceptually simpler to evaluate disagreement between items

****

**6.METHODOLOGIES**

In this project, we implemented two important modules to generate the MR image for getting better information or result.

1. Noise removal

2. Image resizing

3. Genetic algorithm

**6.1Noise Removal:**

Noise reduction is the process of removing noise from a signal.

All recording devices, both analog and digital, have traits that make them susceptible to noise. Noise can be random or white noise with no coherence, or coherent noise introduced by the device's mechanism or processing algorithms.

In electronic recording devices, a major form of noise is hiss caused by random electrons that, heavily influenced by heat, stray from their designated path. These stray electrons influence the voltage of the output signal and thus create detectable noise.

In the case of photographic film and magnetic tape, noise (both visible and audible) is introduced due to the grain structure of the medium. In photographic film, the size of the grains in the film determines the film's sensitivity, more sensitive film having larger sized grains. In magnetic tape, the larger the grains of the magnetic particles (usually ferric oxide or magnetite), the more prone the medium is to noise.

To compensate for this, larger areas of film or magnetic tape may be used to lower the noise to an acceptable level.

Many noise reduction algorithms tend to damage more or less signals. The local signal-and-noise orthogonalization algorithm[1] can be used to avoid the damages to signals.

To know the affected part we apply Automated segmentation. Later we take the

segmented values in a array of elements and apply Genetic algorithm to optimize the image to get the grey part(GM) of the image that is the affected area of tumour

**Median Filter.:**

The median filter is a nonlinear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise (but see discussion below), also having applications in signal processing.

Kaverage = filter2(fspecial('average',3),J)/255;

Kmedian = medfilt2(J);

imshowpair(Kaverage,Kmedian,'montage')

## 6.2Image Resizing:

Resizing the input image to a certain size can result in better application of noise removal and segmentation techniques which are crucial elements of this project.

We achieve this by using the ‘imresize’ function available in matlab

I=imread('brainimg1.jpg');

B=imresize(I,[512 512]);

This results in an 512\*512 image irrespective of the size of image at the time of input.

**Nearest-neighbor interpolation:**

One of the simpler ways of increasing image size is nearest-neighbor interpolation, replacing every pixel with multiple pixels of the same color: The resulting image is larger than the original, and preserves all the original detail, but has (generally undesirable) jaggedness. Diagonal lines, for example, show a "stairway" shape.

**Bilinear and bicubic algorithms:**

Bilinear interpolation works by interpolating pixel color values, introducing a continuous transition into the output even where the original material has discrete transitions. Although this is desirable for continuous-tone images, this algorithm reduces contrast (sharp edges) in a way that may be undesirable for line art. Bicubic interpolation yields substantially better results, with only a small increase in computational complexity.

**Sinc and Lanczos resampling:**

Sinc resampling in theory provides the best possible reconstruction for a perfectly bandlimited signal. In practice, the assumptions behind sinc resampling are not completely met by real-world digital images. Lanczos resampling, an approximation to the sinc method, yields better results. Bicubic interpolation can be regarded as a computationally efficient approximation to Lanczos resampling.

**Box sampling:**

One weakness of bilinear, bicubic and related algorithms is that they sample a specific number of pixels. When down scaling below a certain threshold, such as more than twice for all bi-sampling algorithms, the algorithms will sample non-adjacent pixels, which results in both losing data, and causes rough results.

The trivial solution to this issue is box sampling, which is to consider the target pixel a box on the original image, and sample all pixels inside the box. This ensures that all input pixels contribute to the output. The major weakness of this algorithm is that it is hard to optimize.

**Mipmap:**

Another solution to the downscale problem of bi-sampling scaling are mipmaps. A mipmap is a prescaled set of downscale copies. When downscaling the nearest larger mipmap is used as the origin, to ensure no scaling below the useful threshold of bilinear scaling is used. This algorithm is fast, and easy to optimize. It is standard in many frameworks such as OpenGL. The cost is using more image memory, exactly one third more in the standard implementation.

**Fourier-transform methods:**

Simple interpolation based on Fourier transform pads the frequency domain with zero components (a smooth window-based approach would reduce the ringing). Besides the good conservation (or recovery) of details, notable is the ringing and the circular bleeding of content from the left border to right border (and way around).

**Edge-directed interpolation:**

Edge-directed interpolation algorithms aim to preserve edges in the image after scaling, unlike other algorithms, which can introduce staircase artifacts.

Examples of algorithms for this task include New Edge-Directed Interpolation (NEDI), Edge-Guided Image Interpolation (EGGI), Iterative Curvature-Based Interpolation (ICBI), and Directional Cubic Convolution Interpolation (DCCI). A 2013 analysis found that DCCI had the best scores in PSNR and SSIM on a series of test images.

**Hqx:**

For magnifying computer graphics with low resolution and/or few colors (usually from 2 to 256 colors), better results can be achieved by hqx or other pixel-art scaling algorithms. These produce sharp edges and maintain high level of detail.

**Vectorization:**

Vector extraction, or vectorization, offer another approach. Vectorization first creates a resolution-independent vector representation of the graphic to be scaled. Then the resolution-independent version is rendered as a raster image at the desired resolution. This technique is used by Adobe Illustrator, Live Trace, and Inkscape. Scalable Vector Graphics are well suited to simple geometric images, while photographs do not fare well with vectorization due to their complexity.

**Deep convolutional neural networks:**

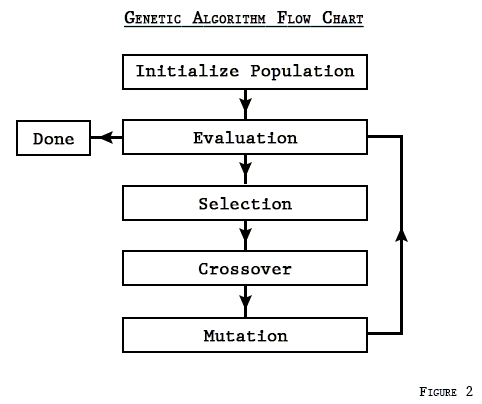
This method uses machine learning for more detailed images such as photographs and complex artwork. Programs that use this method include Waifu2x and Reshade.

**6.3 Genetic Algorithm:**

A genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

Optimization problems

In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.



The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

**A typical genetic algorithm requires:**

a genetic representation of the solution domain,

a fitness function to evaluate the solution domain.

A standard representation of each candidate solution is as an array of bits.

Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

**Initialization**

The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Often, the initial population is generated randomly, allowing the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

**Selection**

During each successive generation, a portion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise.

In some problems, it is hard or even impossible to define the fitness expression; in these cases, a simulation may be used to determine the fitness function value of a phenotype (e.g. computational fluid dynamics is used to determine the air resistance of a vehicle whose shape is encoded as the phenotype), or even interactive genetic algorithms are used.

**Genetic operators**

Main articles: Crossover (genetic algorithm) and Mutation (genetic algorithm)

The next step is to generate a second generation population of solutions from those selected through a combination of genetic operators: crossover (also called recombination), and mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests that more than two "parents" generate higher quality chromosomes.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions. These less fit solutions ensure genetic diversity within the genetic pool of the parents and therefore ensure the genetic diversity of the subsequent generation of children.

Opinion is divided over the importance of crossover versus mutation. There are many references in Fogel (2006) that support the importance of mutation-based search.

Although crossover and mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithms.

It is worth tuning parameters such as the mutation probability, crossover probability and population size to find reasonable settings for the problem class being worked on. A very small mutation rate may lead to genetic drift (which is non-ergodic in nature). A recombination rate that is too high may lead to premature convergence of the genetic algorithm. A mutation rate that is too high may lead to loss of good solutions, unless elitist selection is employed.

**Heuristics**

In addition to the main operators above, other heuristics may be employed to make the calculation faster or more robust. The speciation heuristic penalizes crossover between candidate solutions that are too similar; this encourages population diversity and helps prevent premature convergence to a less optimal solution.

**Termination**

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

A solution is found that satisfies minimum criteria

Fixed number of generations reached

Allocated budget (computation time/money) reached

The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results

Manual inspection

Combinations of the above

**6.4 Skull Stripping:**

Skull stripping, is designed to eliminate non-brain tissues from MR brain images for many clinical applications and analyses, its accuracy and speed are considered as the key factors in the brain image segmentation and analysis. However, the accurate and automated skull stripping methods help to improve the speed and accuracy of prognostic and diagnostic procedures in medical applications.

Skull stripping methods which are available in the literature are broadly classified into five categories: mathematical morphology-based methods, intensity-based methods, deformable surface-based methods, atlas-based methods, and hybrid methods. We have implemented morphology-based method for skull stripping.

### Morphology-Based Methods:

Generally, these methods use the morphological erosion and dilation operations to separate the skull from the brain region. These methods require a combination of thresholding and edge detection methods to find the initial ROI (region of interest). The main drawbacks of these methods are that they often depend on many parameters such as size and shape of the structural element for morphological operation. These parameters are fixed by empirical experimentation; the value on these parameters directly influences the final output of these methods.

The following are the steps:

• Crop the image

• Convert image into binary image

• Seal off the bottom of the head - make the last row white.

• Erode away 15 layers of pixels.

• Mask the gray image

**IMPLEMENTATION**

**7.1Sample Code for image Segmentation**

pkg load image

% Default variables

n\_population = 50;

n\_iterations = 70;

n\_bins = 256;

n\_thresholds = 5;

% Ratios of all GA operations

p\_selection = 0.1;

p\_crossover = 0.7;

p\_mutation = 0.2;

assert(sum([p\_selection, p\_crossover, p\_mutation]) == 1, 'Total sum of proportions have to be 1!');

% Read image

image = imread('s.jpg');

% Resize image

image = imresize(image, [512 512]);

%removing noise

image=medfilt2(image);

% Convert image to gray levels

if (size(image, 3) == 3)

image\_gray = rgb2gray(image);

else

image\_gray = image;

endif

% Initialization

population = initialization(n\_population, n\_bins, n\_thresholds);

for i = 1:n\_iterations

new\_population = [];

% Evaluation of fitness

ranking = fitness(image, population, n\_thresholds);

%% Reproduction

% Selection

% TODO create more strategies (like roulette wheel)

new\_population = first\_best(ranking, population, p\_selection, new\_population);

% Crossover

new\_population = crossover(population, p\_crossover, new\_population);

% Mutation

new\_population = mutation(population, p\_mutation, new\_population);

population = new\_population;

endfor

% Accepting the solution

accept\_solution(image\_gray, population, n\_thresholds);

function accept\_solution(image, population, n\_thresholds)

segmentation = zeros(size(image));

segmentation\_value = 1/n\_thresholds;

genome\_size = size(population, 2)/n\_thresholds;

% Retrieve the best solution

[b, b\_i] = sort(fitness(image, population, n\_thresholds));

best\_genome = population(b\_i(1),:);

threshold = sort(threshold\_bin2dec(best\_genome, n\_thresholds));

value = 0;

end\_i = size(threshold, 2) + 1;

for i = 1:end\_i

if (i == 1)

% The first threshold

left = 0;

right = threshold(i);

elseif (i == end\_i)

% The last threshold

left = threshold(i-1);

right = max(image(:));

else

% Regular threshold

left = threshold(i-1);

right = threshold(i);

endif

% <0; x) <x; y) <y; max(image))

left\_mask = image >= left;

right\_mask = image < right;

mask = left\_mask .\* right\_mask;

segmentation += value\*mask;

% Display segments

%if (i >= 2)

% figure

% mask\_value = value\*mask;

% imshow(mask\_value);

% imwrite(mask\_value, strcat(num2str(i), ".png"));

%endif

value += segmentation\_value;

endfor

imshow(segmentation);

imshow(segmentation,'Colormap',jet(255));

endfunction

**7.2 Sample code forAnalysis:**

**Crossover:**

function new\_population = crossover(population, p\_crossover, new\_population)

population\_size = size(population, 1);

% Random permutation of genomes order

parent\_first = randperm(population\_size);

parent\_second = randperm(population\_size);

% Number of couples used for crossover

n\_crossovers = round(p\_crossover\*population\_size)/2;

for i = 1:n\_crossovers

% Crossovers parents

[desc\_first desc\_second] = crossover\_one(population(parent\_first(i), :), ...

population(parent\_second(i), :));

% Add crossover descendants

new\_population = [new\_population; desc\_first; desc\_second];

endfor

endfunction

function [desc\_first desc\_second] = crossover\_one(parent\_first, parent\_second)

parent\_size = size(parent\_first, 2);

% Randomly generated number between 1 and the length of parent's genome.

point = round(unifrnd(1, parent\_size-1));

% Crossover

desc\_first = [parent\_first(1:point) parent\_second(point+1:parent\_size)];

desc\_second = [parent\_second(1:point) parent\_first(point+1:parent\_size)];

endfunction

**Mutation:**

function new\_population = mutation(population, p\_mutation, new\_population)

population\_size = size(population, 1);

% Random permutation of genomes order

mutation\_order = randperm(population\_size);

for i = 1:round(p\_mutation\*population\_size);

new\_population = [new\_population; mutate\_one(population(mutation\_order(i), :))];

endfor

endfunction

function new\_chromosome = mutate\_one(chromosome)

new\_chromosome = chromosome;

chromosome\_size = size(chromosome, 2);

gene = round(unifrnd(1, chromosome\_size));

% Mutate one gene

if (chromosome(gene) == 1)

new\_chromosome(gene) = 0;

else

new\_chromosome(gene) = 1;

endif

endfunction

**Actual code**

function accept\_solution(image, population, n\_thresholds)

segmentation = zeros(size(image));

segmentation\_value = 1/n\_thresholds;

genome\_size = size(population, 2)/n\_thresholds;

% Retrieve the best solution

[b, b\_i] = sort(fitness(image, population, n\_thresholds));

best\_genome = population(b\_i(1),:);

threshold = sort(threshold\_bin2dec(best\_genome, n\_thresholds));

value = 0;

end\_i = size(threshold, 2) + 1;

for i = 1:end\_i

if (i == 1)

% The first threshold

left = 0;

right = threshold(i);

elseif (i == end\_i)

% The last threshold

left = threshold(i-1);

right = max(image(:));

else

% Regular threshold

left = threshold(i-1);

right = threshold(i);

endif

% <0; x) <x; y) <y; max(image))

left\_mask = image >= left;

right\_mask = image < right;

mask = left\_mask .\* right\_mask;

segmentation += value\*mask;

% Display segments

%if (i >= 2)

% figure

% mask\_value = value\*mask;

% imshow(mask\_value);

% imwrite(mask\_value, strcat(num2str(i), ".png"));

%endif

value += segmentation\_value;

endfor

imshow(segmentation);

endfunction

**Code For Skull Striping:**

grayImage = imread("h.jpg");

% Get the dimensions of the image.

% numberOfColorBands should be = 1.

[rows, columns, numberOfColorBands] = size(grayImage);

if numberOfColorBands > 1

% It's not really gray scale like we expected - it's color.

% Convert it to gray scale by taking only the green channel.

grayImage = grayImage(:, :, 2); % Take green channel.

end

% Display the original gray scale image.

subplot(2, 3, 1);

imshow(grayImage, []);

axis on;

fontSize=20;

title('Original Grayscale Image', 'FontSize', fontSize);

% Enlarge figure to full screen.

set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);

% Give a name to the title bar.

set(gcf, 'Name', 'Demo by ImageAnalyst', 'NumberTitle', 'Off')

% Let's compute and display the histogram.

[pixelCount, grayLevels] = imhist(grayImage);

subplot(2, 3, 2);

bar(grayLevels, pixelCount);

grid on;

title('Histogram of original image', 'FontSize', fontSize);

xlim([0 grayLevels(end)]); % Scale x axis manually.

% Crop image to get rid of light box surrounding the image

grayImage = grayImage(3:end-3, 4:end-4);

% Threshold to create a binary image

binaryImage = grayImage > 20;

% Get rid of small specks of noise

binaryImage = bwareaopen(binaryImage, 10);

% Display the original gray scale image.

subplot(2, 3, 3);

imshow(binaryImage, []);

axis on;

title('Binary Image', 'FontSize', fontSize);

% Seal off the bottom of the head - make the last row white.

binaryImage(end,:) = true;

% Fill the image

binaryImage = imfill(binaryImage, 'holes');

subplot(2, 3, 4);

imshow(binaryImage, []);

axis on;

title('Cleaned Binary Image', 'FontSize', fontSize);

% Erode away 15 layers of pixels.

se = strel('disk', 15, 0);

binaryImage = imerode(binaryImage, se);

subplot(2, 3, 5);

imshow(binaryImage, []);

axis on;

title('Eroded Binary Image', 'FontSize', fontSize);

% Mask the gray image

finalImage = grayImage; % Initialize.

finalImage(~binaryImage) = 0;

subplot(2, 3, 6);

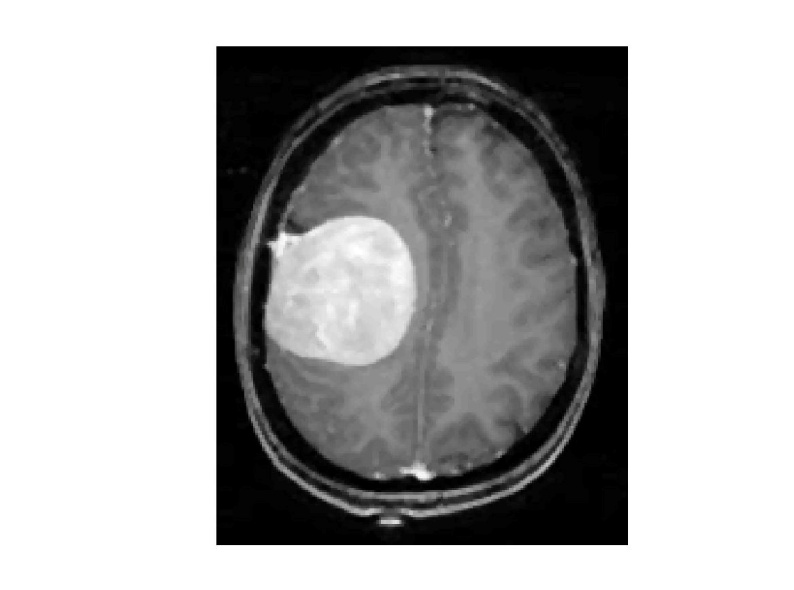
imshow(finalImage, []);

axis on;

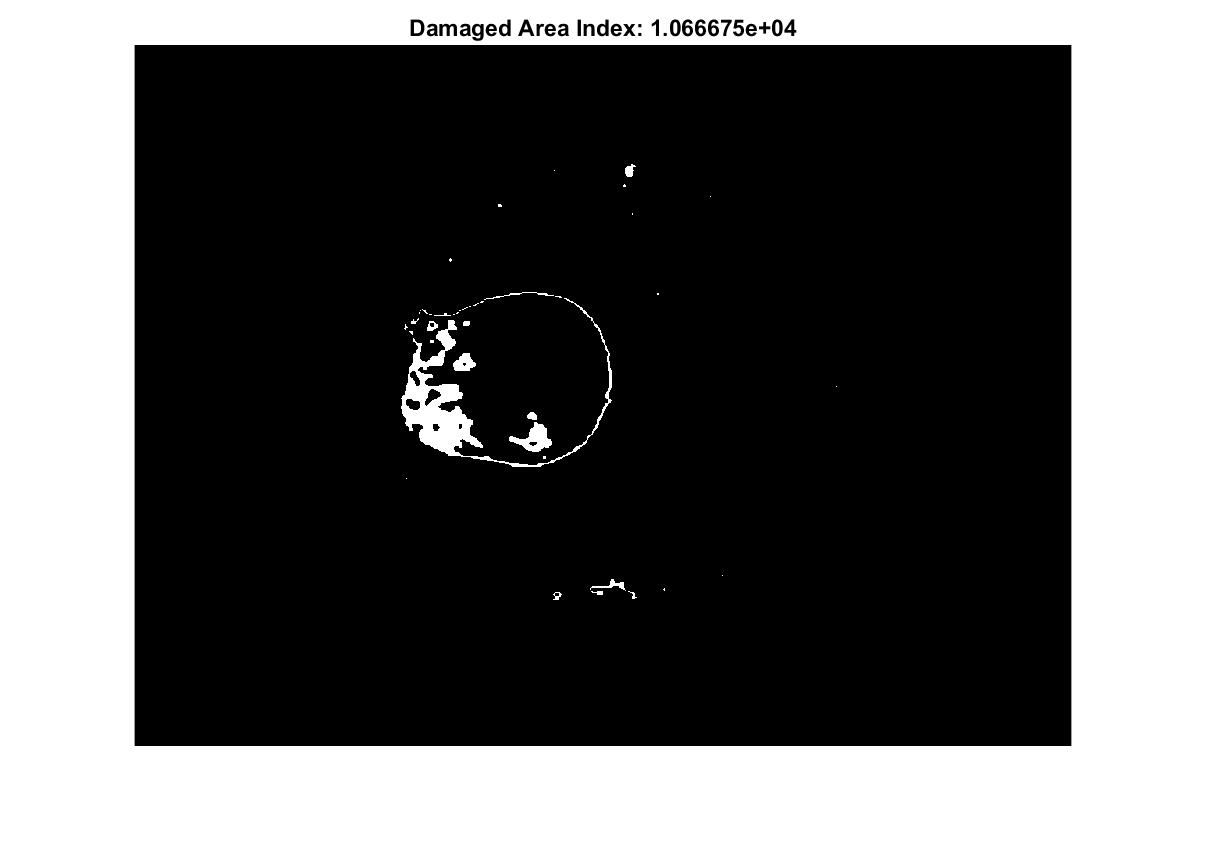
title('Skull stripped Image', 'FontSize', fontSize);

msgbox('Done with demo');

S**ample Input:**

****

**Sample Output:**

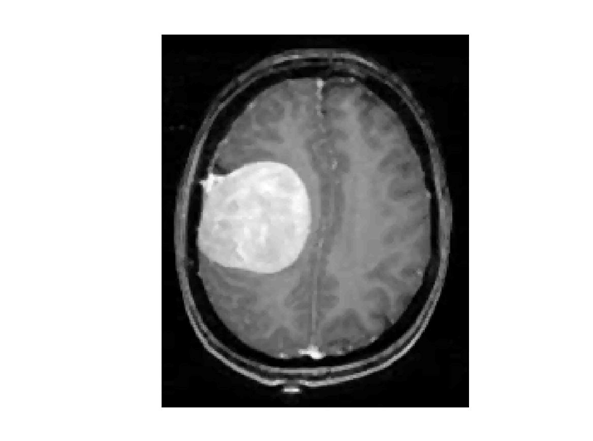
****

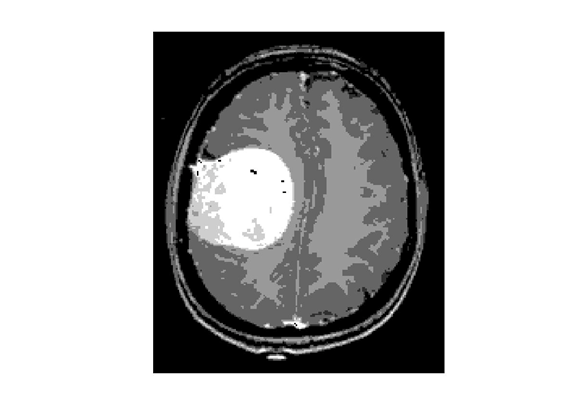
**8.RESULTS**

**8.1Input image:**

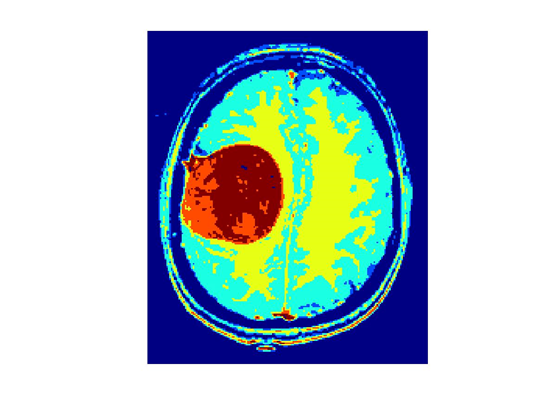


**8.2 Image Filtering Result:**

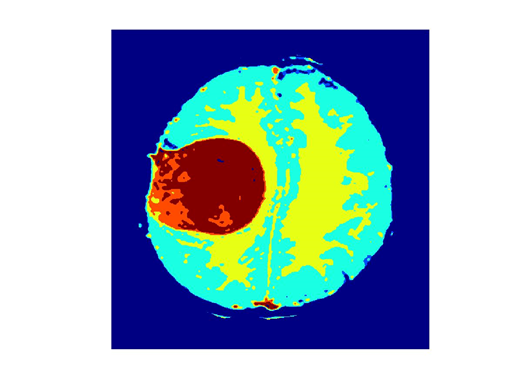
****

**8.3Segmentation Result: **

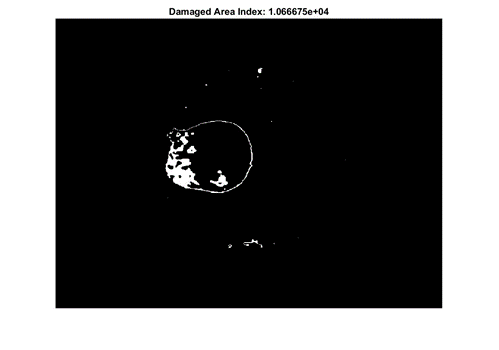
**8.4Colormap Result:**

****

**8.5Skull Stripping Result:**

****

**8.6Damaged Index:**

****

**9.CONCLUSION**

In this study, using MR images of the brain, we segmented brain tissues into normal tissues such as white matter, gray matter, cerebrospinal fluid (background), and tumor-infected tissues. We proposed a Genetic algorithm based automated MRI brain tumour recognition that can identify the tumour region efficiently when compared to semi-automated techniques like k-means, fuzzy c-means, etc. We used preprocessing to improve the signal-to-noise ratio and to eliminate the effect of unwanted noise. We performed skull stripping using morphological operations like erosion, masking, etc., to separate the skull of the brain from remaining brain tissues. We calculated the damaged area index of the brain to obtain the tumour volume. Tumour recognition was successful on more than 10 MRI images of brain. The computation time of the proposed system was very impressive. The algorithm could be refined by two-point crossover or roulette wheel for selecting chromosomes to the next population.

**REFERENCES**

**WEB SITES:**

1. https://en.wikipedia.org/wiki/Genetic\_algorithm
2. <http://www.cai.sk/ojs/index.php/cai/article/viewFile/296/228>
3. <https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)>
4. <https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)>
5. <https://en.wikipedia.org/wiki/K-means_clustering>
6. <https://en.wikipedia.org/wiki/Image_scaling>
7. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>
8. <https://documentation.apple.com/en/finalcutpro/usermanual/index>
9. <https://www.sciencedirect.com/science/article/pii/S1877050915014143>