

**ANALYSIS OF IMAGE FUSION WITH**

**SEGMENTATION ON MEDICAL IMAGES**

**A PROJECT REPORT**

***Submitted by***

**SIVARANJANI.G (211411205099)**

**SNEGHA.R (211411205100)**

**SRINITHA SOUNDAR (211411205101)**

**SUBITSHA.R (211411205104)**

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

***Of***

**BACHELOR OF TECHNOLOGY**

IN

INFORMATION TECHNOLOGY

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600 123**

**ANNA UNIVERSITY: CHENNAI 600 025**

APRIL 2015

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“ANALYSIS OF IMAGE FUSION WITH SEGMENTATION ON MEDICAL IMAGES”** is the bonafide work of “**SIVARANJANI.G (211411205099), SNEGHA.R (211411205100), SRINITHA SOUNDAR (211411205101), SUBITSHA.R (2114111205104)**” who carried out the project work under my supervision.

**SIGNATURE**  **SIGNATURE**

Dr.M. Helda Mercy,M.E., Ph.D., Mr.M.Dillibabu M.E.,

**HEAD OF THE DEPARTMENT**  **ASSISTANT PROFESSOR**

Department of Information Technology, Department of Information Technology,

Panimalar Engineering College, Panimalar Engineering College,

Bangalore Trunk Road, Poonamalle, Bangalore Trunk Road, Poonamalle,

Chennai –600 123 . Chennai –600 123.

Certified that the above mentioned were examined in Anna University Project viva-voce held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We pay our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to our **Honorable Founder and Chairman, Dr.JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere endeavor in educating us in his premier Institution.

We would like to express our deep gratitude to our **Most Respected** **Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., M.Phil., Ph.D.,** for his kind words and enthusiastic motivation which inspired us a lot in completing this project and we also express our sincere thanks to **Our Dynamic Directors Mrs. C. VIJAYA RAJESHWARI, Mr. C. SAKTHI KUMAR, M.E.,** and **Mrs. SARANYA SREE SAKTHIKUMAR, B.E.,** for providing us with the necessary facilities for the completion of this project.

We also express gratitude to Our **Principal Dr. K.MANI, M.E., Ph.D.,** who has been a source of constant encouragement and support.

We would also wish to convey our sincere thanks and heart-filled gratitude to **Dr. M. HELDA MERCY, M.E., Ph.D., HEAD OF THE DEPARTMENT,** Information Technology, for her support without whose co-operation this venture would not have been a success.

We would also like to thank our internal guide **Mr. M.DILLI BABU M.E.,** for his valuable guidance.

Finally, we would like to thank our respected Professors, dear Parents and friends for their constant moral support, encouragement and the environment to work efficiently.

**DECLARATION**

We hereby declare that the project report entitled **“ANALYSIS OF IMAGE FUSION WITH SEGMENTATION ON MEDICAL IMAGES** “which is being submitted in partial fulfillment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology’ in **PANIMALAR ENGINEERING COLLEGE, AFFILIATED TO ANNA UNIVERSITY- CHENNAI** is the result of the project carried out by us under the guidance and supervision of **Mr.M.DILLI BABU M.E.,ASSISTANT PROFESSOR IN THE DEPARTMENT OF INFORMATION TECHNOLOGY**. We further declared that any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

Date: Signature of Batch Members

Place**: Chennai**

(SIVARANJANI.G)

(SNEGHA.R)

(SRINITHA SOUNDAR)

(SUBITSHA.R)

It is certified that this project has been prepared and submitted under my guidance.

Date: (**Mr.M.DILLI BABU)**

Place: **Chennai**  (Assistant professor/IT)

**ABSTRACT**

Brain tumor segmentation is an important procedure for early tumor diagnosis and radiotherapy planning. Enhancing tumor segmentation methods is still challenging because brain tumor images exhibit complex characteristics, such as high variations in tumor appearance and ambiguous tumor boundaries. To address this problem, we propose novel automatic tumor segmentation along with fusion for brain tumor images. Fusion of two or more images produces a new one which contains more accurate information on the scene than any of the individual source images. This technique improves the quality of data. Image fusion is one of the important re-processing steps in digital image reconstruction. Medical imaging field demands images with high resolution and higher information contents for necessary disease diagnosis and visualization.

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGENO**

**ABSTRACT iv LIST OF TABLE ix**

**LIST OF FIGURES ix**

**1 INTRODUCTION**

* 1. SYNOPSIS OF K MEANS ALGORITHM 7
  2. NEED FOR PROPOSED SYSTEM 8
  3. DESCRIPTION OF PROPOSED SYSTEM 9
  4. BENEFITS OF PROPOSED SYSTEM 9

**2 LITREATURE REVIEW**

2.1 SEGMENTATION OF TUMOR IMAGES 11

2.2 ATLAS BASED SEGMENTATION 12

2.3 ROBUST FACE RECOGNITION 13

2.4 SPARSE BRAIN NETWORK RECOVERY 14

2.5 A NON-PARAMETRIC METHOD

FOR CORRECTION IN MRI DATA 15 2.6 FEASABILITY STUDY 18

**3 SYSTEM DESIGN**

3.1 ARCHITECTURAL DESIGN 22

3.2 DATA FLOW DIAGRAM

3.2.1 Level 0 Data Flow Diagram 23

3.2.2 Level 1 Data Flow Diagram 24

3.2.3 Sequence Diagram 26

**4 HARDWARE AND SOFTWARE REQUIREMENTS**

4.1 HARDWARE SPECIFICATIONS 29

4.2 SOFTWARE SPECIFICATIONS 32

4.2.1 OPERATING SYSTEM 32

4.2.2 MATLAB 33

4.2.2.1 General 33

4.2.3 The MATLAB System 34

4.2.3.1 General 34

4.2.3.2 The MATLAB Language 34 4.2.3.3 Working Environment 35

4.2.3.4 Handle Graphics 35

4.2.3.5 Mathematical Function Library 35

4.2.3.6 Application Program Interface 36

5  **IMPLEMENTATION**

5.1 METHODOLOGY 38

5.1.1 Algorithm 40

5.2 MODULE DESIGN 41

5.2.1 Pre-processing 41

5.2.2 Fusion 42

5.2.3 Segmentation 43

5.2.4 Tumor Extraction 44

5.3 CODING 45 5.4 SNAPSHOTS 52

5.4.1 Pre-processing 52

5.4.2 Fusion 52

5.4.3 Segmentation 53

5.4.4 Tumor Extraction 53

5.4.5 Tumor Area Calculation 54

6  **TESTING**

6.1 INTRODUCTION 56

6.2 NEED FOR TESTING 56

6.3 TAXONOMY OF TESTING 58

6.3.1 Correctness Testing 58

6.3.2 Black-box Testing 58

6.3.3 White-box Testing 59

6.3.4 Performance Testing 61

6.3.5 Reliability Testing 61

6.3.6 Security Testing 63

6.3.7 Unit Testing 63

6.3.8 Integration Testing 64

6.3.9 System Testing 64

6.4 TEST REPORT 65

**7 CONCLUSION AND FUTURE ENHANCEMENT**

7.1 CONCLUSION 68

7.2 FUTURE ENHANCEMENT 68 **REFERENCES 69**

**LIST OF TABLES**

**TABLE NO TITLE PAGE NO**

2.5.6 Comparison table16

I Test Report 65

**LIST OF FIGURES**

**FIGURE NO TITLE PAGE NO**

3.1 Architectural Design 22

3.2 Level 0 DFD Diagram 23

3.3 Level 1 DFD Diagram 24

3.4 Sequence Diagram 26

5.1 Flowchart for Pre-processing Process 41

5.2 Flowchart for Fusion Process 42

5.3 Flowchart for Segmentation Process 43

5.4 Flowchart for Tumor Extraction Process 44

5.5 Selected Images 52

5.6 Fused Image 52

5.7 Segmented Clusters 53

5.8 Tumor Extraction 53

5.9 Tumor Area Calculation 54

**CHAPTER 1**

**CHAPTER 1**

**INTRODUCTION**

With the advances in imaging technology, diagnostic imaging has become an indispensable tool in medicine today. X-ray angiography (XRA), magnetic resonance angiography (MRA), magnetic resonance imaging (MRI), computed tomography (CT), and other imaging modalities are heavily used in clinical practice. Such images provide complementary information about the patient. While increased size and volume in medical images required the automation of the diagnosis process, the latest advances in computer technology and reduced costs have made it possible to develop such systems.

Blood vessel delineation on medical images forms an essential step in solving several practical applications such as diagnosis of the vessels (e.g. stenosis or malformations) and registration of patient images obtained at different times. Segmentation algorithms form the essence of medical image applications such as radiological diagnostic systems, multimodal image registration, creating anatomical atlases, visualization, and computer-aided surgery.

Vessel segmentation algorithms are the key components of automated radiological diagnostic systems. Segmentation methods vary depending on the imaging modality, application domain, method being automatic or semi-automatic, and other specific factors. There is no single segmentation method that can extract vasculature from every medical image modality. While some methods employ pure intensity-based pattern recognition techniques such as threshold followed by connected component analysis, some other methods apply explicit vessel models to extract the vessel contours.

Depending on the image quality and the general image artifacts such as noise, some segmentation methods may require image preprocessing prior to the segmentation algorithm. On the other hand, some methods apply post-processing to overcome the problems arising from over segmentation. Vessel segmentation algorithms and techniques can be divided into six main categories, pattern recognition techniques, model-based approaches, tracking-based approaches, artificial intelligence-based approaches, neural network-based approaches, and miscellaneous tube-like object detection approaches.

Pattern recognition techniques are further divided into seven categories, multi-scale approaches, skeleton-based approaches, region growing approaches, ridge-based approaches, differential geometry-based approaches, matching filters approaches, and mathematical morphology schemes. Clustering analysis plays an important role in scientific research and commercial application. This thesis provides a survey of current vessel segmentation methods using clustering approach and provides both early and recent literature related to vessel segmentation algorithms and techniques.

An algorithm is described for segmenting 3D MR brain image into K different tissue types, which include gray, white matter and CSF, and maybe other abnormal tissues. MR images considered can be either scale- or multi-valued. Each scale-valued image is modeled as a collection of regions with slowly varying intensity plus a white Gaussian noise. Each tissue type is modeled by a Markov random field with the second order neighborhood in a 3D lattice.

The proposed algorithm is an adaptive K-means clustering algorithm for 3-dimensional and multi-valued images. Every iteration consists of two steps: estimate mean intensity at each location for each type, and estimate tissue types by maximizing the a posteriori probability. The algorithm slowly adapts to the local intensity variation of each region, so it is robust to the “shading” effect. It has the potential to routinely process clinical MR images with minimal user involvement. Its performance is tested using patient data.

Numerous techniques developed and reported in the literature for segmenting MR brain images. Most of them are based on clustering voxels with similar features as the same tissue type. Methods that rely exclusively on partitioning the feature space such as the histogram or scatter gram work well only for MR images with excellent soft tissue contrast, high signal-to noise ratio (SNR) and negligible “shading effect”. Unfortunately, most of clinically available MR images do not yet meet such a quality standard.

It has been shown that the incorporation of spatial interactions between neighboring voxels into the segmentation procedure reduces significantly the degradation of segmentation outcome due to poor SNR and feature contrast. But the quality of segmentation is still limited by the failure of effectively compensating for the “shading” effect, which is due to an inhomogeneous RF field across the imaging volume. In contrast, edge based segmentation techniques, which explore local intensity contrast to partition the image into regions through the Marr-Hildreth operator or graph theory, are insensitive to the “shading” effect.

The actual fusion process can take place at different levels of information representation, a generic categorization is to consider the different levels as, sorted in ascending order of abstraction: signal, pixel, feature and symbolic level. This site focuses on the so-called pixel level fusion process, where a composite image has to be built of several input images. To date, the result of pixel level image fusion is considered primarily to be presented to the human observer, especially in image sequence fusion (where the input data consists of image sequences).

A possible [application](http://www.metapix.de/examples.htm) is the fusion of forward looking infrared (FLIR) and low light visible images (LLTV) obtained by an airborne sensor platform that aid a pilot navigate during poor weather conditions or darkness. In pixel-level image fusion, some generic requirements can be imposed on the fusion result. The fusion process should preserve all relevant information of the input imagery in the composite image (pattern conservation). Fusion scheme should not introduce any artifacts or inconsistencies which would distract the human observer or following processing stages.

The fusion process should be shift and rotationally invariant, i.e. the fusion result should not depend on the location or orientation of an object the input imagery. Image sequence fusion arise the additional problem of temporal stability and consistency of the fused image sequence. The human visual system is primarily sensitive to moving light stimuli, so moving artifacts or time depended contrast changes introduced by the fusion process are highly distracting to the human observer. So, in case of image sequence fusion the two additional requirements apply.

**Temporal stability**: The fused image sequence should be temporal stable, i.e. the gray level changes in the fused sequence must only be caused by the gray level changes in the input sequences, but it must not be introduced by the fusion scheme itself;

**Temporal consistency:** Gray level changes occurring in the input sequences must be present in the fused sequence without any delay or contrast change.

The wavelet transform is used here for fusion process. The wavelet transform results in a non redundant image representation. The discrete 2-dim wavelet transform is computed by the recursive application of low pass and high pass filters in each direction of the input image (i.e. rows and columns) followed by sub sampling. One major drawback of the wavelet transform when applied to image fusion is its well known shift dependency, i.e. a simple shift of the input signal may lead to complete different transform coefficients.

This results in inconsistent fused images when invoked in image sequence fusion. To overcome the shift dependency of the wavelet fusion scheme, the input images must be decomposed into a shift invariant representation. There are several ways to achieve this: The straightforward way is to compute the wavelet transform for all possible circular shifts of the input signal. In this case, not all shifts are necessary and it is possible to develop an efficient computation scheme for the resulting wavelet representation. Another simple approach is to drop the subsampling in the decomposition process and instead modify the filters at each decomposition level, resulting in a highly redundant signal representation.

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each component with a resolution matched to its scale. The basic idea of the wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts).

They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Many new wavelet applications such as image compression, turbulence, human vision, radar, and earthquake prediction are developed in recent years. In wavelet transform the basic functions are wavelets. Wavelets tend to be irregular and symmetric.

The wavelets are called orthogonal when their inner products are zero. The smaller the scaling factor is, the wider the wavelet is. Wide wavelets are comparable to low-frequency sinusoids and narrow wavelets are comparable to high-frequency sinusoids. Wavelet analysis produces a time-scale view of a signal. Scaling a wavelet simply means stretching (or compressing) it. The scale factor is used to express the compression of wavelets and often denoted by the letter a. The smaller the scale factor, the more “compressed” the wavelet. The scale is inversely related to the frequency of the signal in wavelet analysis.

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. If the scales and positions are chosen based on powers of two, the so-called dyadicscales and positions, then calculating wavelet coefficients are efficient and just as accurate. This is obtained from discrete wavelet transform(DWT). For many signals, the low-frequency content is the most important part. It is the identity of the signal. The high-frequency content, on the other hand, imparts details to the signal. In wavelet analysis, the approximations and details are obtained after filtering. The approximations are the high-scale, low frequency components of the signal.

* 1. **Synopsis Of K Means Algorithm:**

To segment the 3D medical image using K Means clustering algorithm. To propose an algorithm that can be better for large datasets and to find initial centroid. To compare the performance, an algorithm is described for segmenting 3D MR brain image into *K* different tissue types, which include gray, white matter and CSF, and maybe other abnormal tissues. MR images considered can be either scale- or multi-valued. Each scale-valued image is modeled **as** a collection of regions with slowly varying intensity plus a white Gaussian noise. The proposed algorithm is an adaptive K-means clustering algorithm for 3-dimensional and multi-valued images.

A pixel-level multi-sensor image fusion algorithm is used to generate a contrast-enhanced fused image while preserving visually salient information. The proposed algorithm is modified. In order to accomplish both image fusion and contrast enhancement simultaneously, we introduce a fusion Strategy by modifying the framework of sub band-decomposed multi-scale retinex (SDMSR), a contrast enhancement algorithm based on a multi-scale scheme.

* 1. **Need For Proposed System**

The improved K-means algorithm is a solution to handle large scale data, which can select initial clustering center purposefully, reduce the sensitivity to isolated point, avoid dissevering big cluster. By using this technique locating the initial seed point is easy and which will give more accurate and high-resolution result. By using various techniques we can study or compare the results and find out which technique gives higher resolution.

Initial centroid algorithm is useful to avoid the formation of empty clusters, as the centroid values are taken with respect to the intensity value of the image. Proposed algorithm is better for large datasets and to find initial centroid. Wavelet algorithm is useful here to remove noise from the image.

* In hospitals for scanning purpose
* In cryptography and image hiding process

**1.3 Description Of Proposed System**

The adaptive K-mean clustering algorithm is the closest in spirit for meeting our requirements for MR image segmentation. It can incorporate interactions between adjacent voxels and **as** well **as** estimate slow spatial variations of the feature characteristics of the image. By extending this algorithm to 3-dimensional, multi-valued images and incorporating specific characteristics of MR images, a 3D MR image segmentation algorithm An algorithm **,** developed for detection of PET and MR brain contours for registration, is used to extract the MR brain before our segmentation is performed. The segmentation is performed on slice by slice basis because of the large slice thickness. For the SPGR MR images, the segmentation is performed in a true **3D** fashion.

**1.4 Benefits Of Proposed System**

* Initial centroid algorithm is useful to avoid the formation of empty clusters, as the centroid values are taken with respect to the intensity value of the image. The algorithm is better for large datasets and to find initial centroid.
* The Wavelet algorithm removes the noise at the time of preprocessing.

**CHAPTER 2**

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Segmentation of brain tumor images based on atlas- registration combined with a markov-random-field lesion growth model :**

**Stefan Bauer, Lutz-P. Nolte, Mauricio Reyes**

We present an automatic method to segment brain tissues from volumetric MRI brain tumor images. The method is based on non-rigid registration of an average atlas in combination with a biomechanically justified tumor growth model to simulate soft-tissue deformations caused by the tumor mass-effect. The tumor growth model, which is formulated as a mesh-free Markov Random Field energy minimization problem, ensures correspondence between the atlas and the patient image, prior to the registration step. The method is non-parametric, simple and fast compared to other approaches while maintaining similar accuracy. It has been evaluated qualitatively and quantitatively with promising results on eight datasets comprising simulated images

**Techniques Used**

Brain Tumor, Brain Tissue Segmentation, Atlas Registration, Markov Random Field

**2.2 Atlas-based segmentation of pathological brain MR images:**

**M. Bach Cuadra, C. Pollo\*, A. Bardera, O. Cuisenaire, J.-G. Villemure\* and J.-Ph. Thiran**

We propose a method for brain atlas deformation in presence of large space-occupying tumors, based on an a priori model of lesion growth that assumes radial expansion of the lesion from its starting point. First, an affine registration brings the atlas and the patient into global correspondence. Then, the seeding of a synthetic tumor into the brain atlas provides a template for the lesion. Finally, the seeded atlas is deformed, combining a method derived from optical \_own principles and a model of lesion growth (MLG). Results show that the method can be applied to the automatic segmentation of structures and substructures in brains with gross deformation, with important medical applications in neurosurgery, radio surgery and radiotherapy.

**Techniques Used:**

Magnetic Resonance(MR), seeded atlas deformation (SAD), non-rigid

registration, spectral segmentation, demons.

**2.3 Robust Face Recognition via Sparse Representation:**

**John Wright, Student Member, Allen Y. Yang, Member, Arvind Ganesh, Student Member, S. Shankar Sastry, Fellow, and Yi Ma, Senior Member.**

We consider the problem of automatically recognizing human faces from frontal views with varying expression and illumination, as well as occlusion and disguise. We cast the recognition problem as one of classifying among multiple linear regression models, and argue that new theory from sparse signal representation offers the key to addressing this problem. Based on a sparse representation computed by `1-minimization, we propose a general classification algorithm for (image-based) object recognition. This new framework provides new insights into two crucial issues in face recognition: feature extraction and robustness to occlusion. For feature extraction, we show that if sparsity in the recognition problem is properly harnessed, the choice of features is no longer critical. What is critical, however, is whether the number of features is sufficiently large and whether the sparse representation is correctly computed. Unconventional features such as down sampled images and random projections perform just as well as conventional features such as Eigen faces and Laplacian faces, as long as the dimension of the feature space surpasses certain threshold, predicted by the theory of sparse representation. This framework can handle errors due to occlusion and corruption uniformly, by exploiting the fact that these errors are often sparse with respect to the standard (pixel) basis. The theory of sparse representation helps predict how much occlusion the recognition algorithm can handle and how to choose the training images to maximize robustness to occlusion.

**Techniques Used:**

Face Recognition, Feature Extraction, Occlusion and Corruption, Sparse Representation, Compressed Sensing, Minimization, Validation and Outlier Rejection

**2.4 Sparse Brain Network Recovery under Compressed Sensing :**

**Hyekyoung Lee, Dong Soo Lee, Hyejin Kang, Boong-Nyun Kim, and**

**Moo K. Chung**

Partial correlation is a useful connectivity measure for brain networks, especially, when it is needed to remove the confounding effects in highly correlated networks. Since it is difficult to estimate the exact partial correlation under the small-n large-p situation, a sparseness constraint is generally introduced. In this paper, we consider the sparse linear regression model with a l1-norm penalty, also known as the least absolute shrinkage and selection operator (LASSO), for estimating sparse brain connectivity. LASSO is a well-known decoding algorithm in the compressed sensing (CS). The CS theory states that LASSO can reconstruct the exact sparse signal even from a small set of noisy measurements. We briefly show that the penalized linear regression for partial correlation estimation is related to CS. It opens a new possibility that the proposed framework can be used for a sparse brain network recovery. As an illustration, we construct sparse brain networks of 97 regions of interest (ROIs) obtained from FDG-PET imaging data for the autism spectrum disorder (ASD) children and the paediatric control (PedCon) subjects. As validation, we check the network reproducibility’s by leave-one-out cross validation and compare the clustered structures derived from the brain networks of ASD and Pecan.

**Techniques Used :**

Brain Connectivity, Compressed Sensing, Partial Correlation, LASSO

**2.5 A Nonparametric Method for Automatic Correction of Intensity Non uniformity in MRI Data :**

**John G. Sled,\* Alex P. Zijdenbos, Member, IEEE, and Alan C. Evans**

A novel approach to correcting for intensity non uniformity in magnetic resonance (MR) data is described that achieves high performance without requiring a model of the tissue classes present. The method has the advantage that it can be applied at an early stage in an automated data analysis, before a tissue model is available.

Described as nonparametric non uniform intensity normalization (N3), the method is independent of pulse sequence and insensitive to pathological data that might otherwise violate model assumptions. To eliminate the dependence of the field estimate on anatomy, an iterative approach is employed to estimate both the multiplicative bias field and the distribution of the true tissue intensities. The performance of this method is evaluated using both real and simulated MR data.

**Techniques Used :**

Intensity non-uniformity, magnetic resonance imaging, RF field inhomogeneity, shading artifact.

|  |  |  |
| --- | --- | --- |
| **SURVEY** | **ADVANTAGES** | **DISADVANTAGES** |
| A Nonparametric Method for Automatic Correction of Intensity Non-uniformity in MRI Data. | * High performance is obtained without requiring a model of the tissue classes present. * The method can be applied at an early stage in an automated data analysis, before a tissue model is available. | * It is not fully automatic. Needs experts advice and guidance. * Used to correct only the noisy images that are more complex in nature. |
| Sparse Brain Network Recovery under Compressed Sensing. | * Used even to correct a small set of noisy measurement. * It is very much useful in predicting autism spectrum disorder children(ASD). | * More complex. * Needs expert’s guidance and not fully automatic. |
| Robust Face Recognition via Sparse Representation | * Helps to predict how much occlusion the recognition algorithm can handle and how to choose the training images to maximize robustness to occlusion | * As there are more no. of .features, it is difficult to extract feature. |
| Atlas based segmentation of pathological brain MR images. | * Mainly used to automate the process of identifying the tumor area in the brain. * Has its application in neurosurgery | * This model is insensitive to tumor growth |
| Segmentation of Brain tumor images based on Atlas registration combined with a Markov-Random Field Lesion growth model. | * The method is based on non-rigid registration of an average atlas in combination with a biomechanically justified tumor growth model to simulate soft-tissue deformations caused by the tumor mass-effect | * This method cannot be applied to compare the images of different scan types. |

**Table 2.5.6 Comparison table**

**2.6 FEASIBILITY STUDY**

The **feasibility study** is an evaluation and analysis of the potential of a proposed project. It is based on extensive investigation and research to support the process of decision making.

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the [environment](http://en.wikipedia.org/wiki/Natural_environment), the [resources](http://en.wikipedia.org/wiki/Resources) required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are [cost](http://en.wikipedia.org/wiki/Cost) required and [value](http://en.wikipedia.org/wiki/Value_%28economics%29) to be attained.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

**Technology and system feasibility**

The assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

* A brief description of the business to assess more possible factors which could affect the study
* The part of the business being examined
* The human and economic factor
* The possible solutions to the problem

At this level, the concern is whether the proposal is both *technically* and *legally* feasible (assuming moderate cost).

**Legal Feasibility**

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

**Technical Feasibility**

The [technical feasibility](http://en.wikipedia.org/wiki/Technical_feasibility) assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, [corporate culture](http://en.wikipedia.org/wiki/Corporate_culture), and existing business processes.

**Schedule Feasibility**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

**Economic Feasibility**

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

**CHAPTER 3**

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 ARCHITECTURAL DESIGN**

**Fig 3.1 Architectural Design**

The above diagram shows the tumor extraction from brain. It should undergo several processes to find the tumor. At first scanning is done, then the two scanned image should be fused using algorithms. The fused image then undergoes segmentation. Before segmentation, centroid calculation has to be done and then image is segmented. The white matter, extracted from segmentation is called as tumor.

**3.2 DATA FLOW DIAGRAM**

**3.2.1 LEVEL 0 DATA FLOW DIAGRAM**

**Fig 3.2 Level 0 DFD Diagram**

In this process, CT and MRI scan images are taken as inputs. It undergoes two process, fusion and segmentation. The output of fusion is given to perform the segmentation. After performing segmentation, white matter which refers to tumor is extracted. For fusion wavelet and fuzzy algorithms are used and for segmentation K-means clustering method is used.

**3.2.2 LEVEL 1 DATA FLOW DIAGRAM**

Fig.3.3 Level 1 DFD Diagram

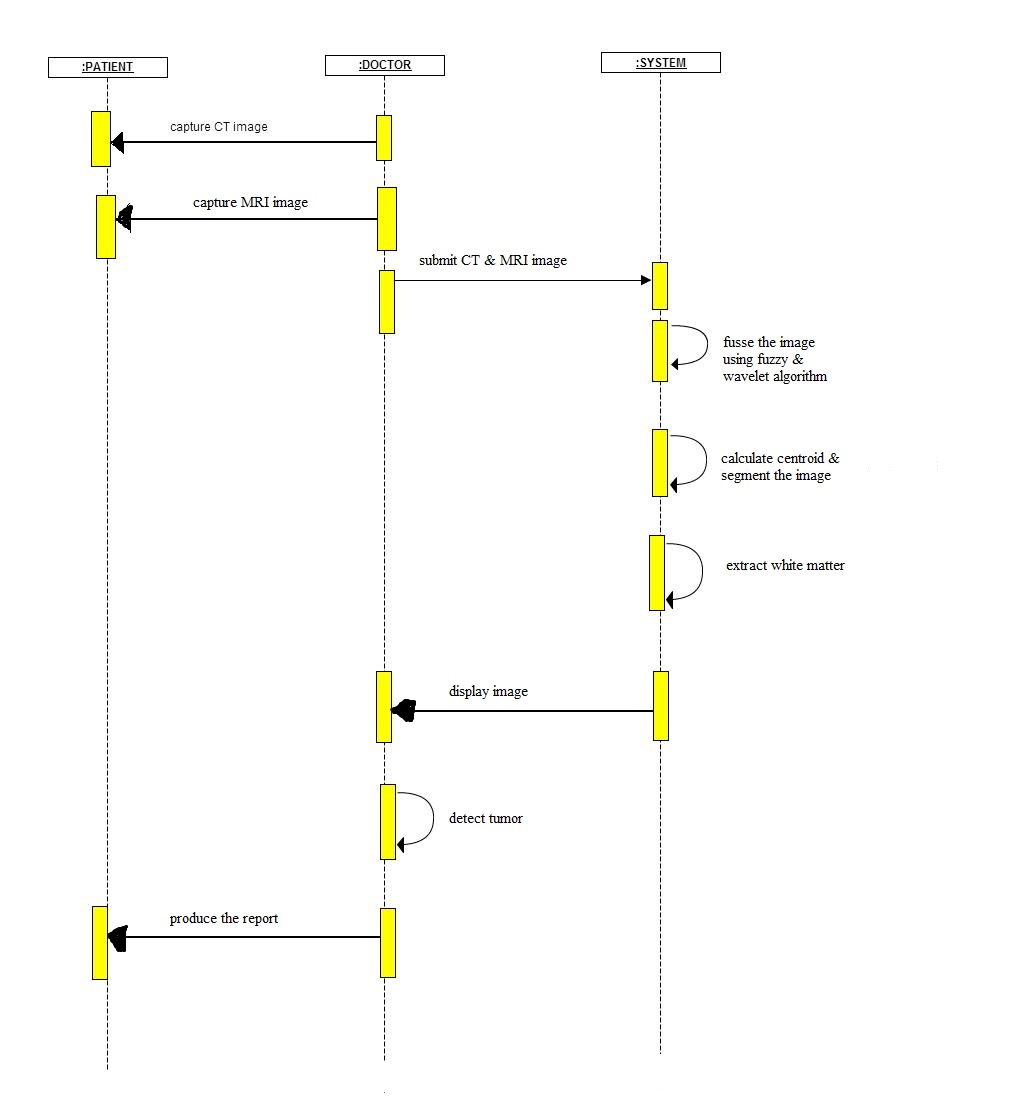
Data Flow Diagrams(DFD) helps us in identifying existing processes. The above DFD diagram shows the overall process of tumor extraction process. It shows the top down approach of entire process.

Two types of scans are performed, they are CT scan and MRI scans. CT scan finds hard tissues and MRI scan finds soft tissues. These scanned images are preprocessed to get a clear image. Preprocessing means translation of images. For these translation Fuzzy algorithm technique is used. These translated images undergo fusion process, where both images are fused together without loss of any information. Image fusion is done based on the pixels of same type.

Segmentation process is performed on fused image in which it performs centroid calculation. Centroid calculation assigns K value for each cluster. The centroid calculation removes the clustering problem. Cluster means group of same pixels combined together. Centroid calculaton has to be done after fusion process.

Segmentation separates the vessel structure from background and it avoids self occlusion of vessels. It refers to the process of partitioning [digital image](http://en.wikipedia.org/wiki/Digital_image) into multiple [regions](http://en.wikipedia.org/wiki/Region). The segmentation process is done using the K-means clustering method. This method is applied after the centroid calculation. The White matter regions are extracted from those regions. Thus, the tumor affected brain part is extracted.

**3.2.3 SEQUENCE DIAGRAM:**

****

**Fig 3.4.Sequence Diagram**

UML sequence diagrams model the flow of logic in a visual manner, enabling us to document and validate our logic, and are commonly used for both analysis and design purposes. Sequence diagrams are UML artifact for dynamic modeling. The above diagram shows the sequence of actions which take place in the tumor extraction framework.

The doctor scans the patient with help of scanning machine. Two types of scans are taken namely CT scan and MRI scan. These scans are preprocessed and fused by system. They are fused on basis of pixels of same type. These fused image need to be segmented, by performing over the fused image. In segmentation self occlusion of vessels is removed and all clusters are formed into regions. White matter region is obtained and then tumor affected brain part is extracted.

**CHAPTER 4**

**CHAPTER 4**

**SOFTWARE AND HARDWARE REQUIREMENTS**

**4.1 Hardware Specifications:**

* Main Processor : Pentium IV 2.4 GHz
* Hard Disk : 40 GB
* RAM : 1 GB RAM
* Monitor : High Color 800 by 600
* Mouse : Logitech
* Keyboard : Standard Keyboard

**Processor**

**Pentium 4** is a line of single-core [desktop](http://en.wikipedia.org/wiki/Desktop_computer), [laptop](http://en.wikipedia.org/wiki/Laptop) and entry level [server](http://en.wikipedia.org/wiki/Server) [central processing units](http://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) introduced by [Intel](http://en.wikipedia.org/wiki/Intel). Pentium 4 processors have an integrated heat spreader (IHS) that prevents the die from accidentally being damaged when mounting and unmounting cooling solutions. Prior to the IHS, a [CPU shim](http://en.wikipedia.org/wiki/CPU_shim) was sometimes used by people worried about damaging the core. Overclockers sometimes removed the IHS from Socket 423 and Socket 478 chips to allow for more direct heat transfer. On processors using the Socket LGA 775 (Socket T) interface, the IHS is directly soldered to the die or dies, making it difficult to remove.

**Monitor**

A **monitor** or a **display** is an [electronic visual display](http://en.wikipedia.org/wiki/Electronic_visual_display) for [computers](http://en.wikipedia.org/wiki/Computer). The monitor comprises the display device, [circuitry](http://en.wikipedia.org/wiki/Electronic_circuit) and an enclosure. The display device in modern monitors is typically a [thin film transistor liquid crystal display](http://en.wikipedia.org/wiki/Thin_film_transistor_liquid_crystal_display) (TFT-LCD) thin panel, while older monitors used a [cathode ray tube](http://en.wikipedia.org/wiki/Cathode_ray_tube) (CRT) about as deep as the screen size.

Originally, computer monitors were used for [data processing](http://en.wikipedia.org/wiki/Data_processing) while [television receivers](http://en.wikipedia.org/wiki/Television_receiver) were used for entertainment. From the 1980s onwards, computers (and their monitors) have been used for both data processing and entertainment, while televisions have implemented some computer functionality. The common [aspect ratio](http://en.wikipedia.org/wiki/Aspect_ratio) of televisions, and then computer monitors, has also changed from 4:3 to 16:9. Computer monitors have traditionally possessed higher resolutions than most televisions. Many web sites and multimedia products were re-designed from the previous 800 × 600 format to the layouts optimized for 1024 × 768.

**Mouse**

In [computing](http://en.wikipedia.org/wiki/Computing), a mouse is a [pointing device](http://en.wikipedia.org/wiki/Pointing_device) that detects [two-dimensional](http://en.wikipedia.org/wiki/Two-dimensional_space) motion relative to a surface. This motion is typically translated into the motion of a [pointer](http://en.wikipedia.org/wiki/Pointer_%28graphical_user_interfaces%29) on a [display](http://en.wikipedia.org/wiki/Computer_monitor), which allows for fine control of a [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface).

A mouse typically controls the motion of a [pointer](http://en.wikipedia.org/wiki/Pointer_%28graphical_user_interfaces%29) in two dimensions in a graphical user interface (GUI). The mouse turns movements of the hand backward and forward, left and right into equivalent electronic signals that in turn are used to move the pointer.

The relative movements of the mouse on the surface are applied to the position of the pointer on the screen, which signals the point where actions of the user take place, so that the hand movements are replicated by the pointer.[[22]](http://en.wikipedia.org/wiki/Mouse_%28computing%29#cite_note-dummies-22) Clicking or hovering (stopping movement while the cursor is within the bounds of an area) can select files, programs or actions from a list of names, or (in graphical interfaces) through small images called "icons" and other elements.

**Keyboard**

A Keyboard is a panel of keys that operate a computer or typewriter. A **keyboard** is a [typewriter-style device](http://en.wikipedia.org/wiki/Typewriter#Keyboard_layouts:_.22QWERTY.22_and_others), which uses an arrangement of buttons or [keys](http://en.wikipedia.org/wiki/Push-button), to act as mechanical levers or electronic switches. Despite the development of alternative input devices, such as the [mouse](http://en.wikipedia.org/wiki/Mouse_%28computing%29), [touchscreen](http://en.wikipedia.org/wiki/Touchscreen), [pen devices](http://en.wikipedia.org/wiki/Pen_computing), [character recognition](http://en.wikipedia.org/wiki/Character_recognition) and [voice recognition](http://en.wikipedia.org/wiki/Speech_recognition), the keyboard remains the most commonly used device for direct (human) input of [alphanumeric](http://en.wikipedia.org/wiki/Alphanumeric) data into computers.

Standard alphanumeric keyboards have keys that are on three-quarter inch centers (0.750 inches, 19.05 mm) and have a key travel of at least 0.150 inches (3.81 mm). Desktop computer keyboards, such as the 101-key US traditional keyboards or the 104-key Windows keyboards, include alphabetic characters, [punctuation](http://en.wikipedia.org/wiki/Punctuation) symbols, numbers and a variety of [function keys](http://en.wikipedia.org/wiki/Function_Keys).

**RAM**

**Random-access memory** is a form of [computer data storage](http://en.wikipedia.org/wiki/Computer_data_storage). Today, random-access memory takes the form of [integrated circuits](http://en.wikipedia.org/wiki/Integrated_circuit). RAM is normally associated with [volatile](http://en.wikipedia.org/wiki/Volatile_memory) types of memory (such as [DRAM](http://en.wikipedia.org/wiki/DRAM) [memory modules](http://en.wikipedia.org/wiki/DIMM)), where stored information is lost if power is removed, although many efforts have been made to develop non-volatile RAM chips.[[2]](http://en.wikipedia.org/wiki/Random-access_memory#cite_note-2) Other types of non-volatile memory exist that allow random access for read operations, but either do not allow write operations or have limitations on them. These include most types of [ROM](http://en.wikipedia.org/wiki/Read_only_memory) and a type of [flash memory](http://en.wikipedia.org/wiki/Flash_memory) called [*NOR-Flash*](http://en.wikipedia.org/wiki/Flash_memory#NOR_flash).

The two main forms of modern RAM are [static RAM](http://en.wikipedia.org/wiki/Static_random_access_memory) (SRAM) and [dynamic RAM](http://en.wikipedia.org/wiki/Dynamic_random-access_memory) (DRAM). In SRAM, a [bit of data](http://en.wikipedia.org/wiki/Bit) is stored using the state of a six transistor [memory cell](http://en.wikipedia.org/wiki/Memory_cell_%28binary%29). This form of RAM is more expensive to produce, but is generally faster and requires less power than DRAM and, in modern computers, is often used as cache memory for the [CPU](http://en.wikipedia.org/wiki/Central_processing_unit). DRAM stores a bit of data using a transistor and capacitor pair, which together comprise a DRAM [memory cell](http://en.wikipedia.org/wiki/Memory_cell_%28binary%29).

**4.2 Software Specifications:**

* Operating system : Windows XP Professional
* Software used : MATLAB

**4.2.1 Operating System**

**Windows XP** is a [personal computer](http://en.wikipedia.org/wiki/Personal_computer) [operating system](http://en.wikipedia.org/wiki/Operating_system) produced by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) for use on personal computers, including home and business desktops and laptops and media centers. Windows XP was a major advance from the [MS-DOS based](http://en.wikipedia.org/wiki/Windows_9x) versions of [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) in security, stability and efficiencydue to its use of Windows NT underpinnings. It introduced a significantly redesigned [graphical user interface](http://en.wikipedia.org/wiki/Graphical_user_interface) and was the first version of Windows to use [product activation](http://en.wikipedia.org/wiki/Microsoft_Product_Activation) in an effort to reduce [software piracy](http://en.wikipedia.org/wiki/Copyright_infringement_of_software).

Upon its release Windows XP received generally positive reviews, with critics noting increased performance , a more intuitive user interface, improved hardware support, and its expanded multimedia capabilities. Despite some initial concerns over the new licensing model and product activation system, Windows XP eventually proved to be popular and widely used.

**4.2.2 MATLAB:**

**4.2.2.1 General:**

**MATLAB** is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical use include:

* Math and computation
* Algorithm development
* Data analysis, exploration and visualization
* Modelling, simulation, prototyping
* Scientific and engineering graphics
* Application development, including Graphics User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulation, in a fraction of time it would take to write a program in a scalar no interactive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB is originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together present the state-of-the-art in software for matrix computation. MATLAB has evolved over a period of years with input from many students. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collection of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system, neural networks, fuzzy logic, wavelets, simulation, and many others.

**4.2.3 The MATLAB System:**

**4.2.3.1 General:**

MATLAB system are high-performance language for technical computing. In integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. A MATLAB accepts the data‘s in a matrix form. It is an easy to use language and can perform various complex computation.

The MATLAB system consist of five main parts:

**4.2.3.2 The MATLAB language:**

This is high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create complete large and complex application programs.

**4.2.3.3 The MATLAB working environment:**

This is a set of tools and facilities that you can work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tool for developing, managing, debugging, and profiling M-files, MATLAB’s applications.

**4.2.3.4 Handle Graphics:**

This is a MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of the graphics as well as to build a complete Graphical User Interfaces on the MATLAB applications.

**4.2.3.5 The MATLAB mathematical function library:**

This is a vast range of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix values, Bessel functions, and fast Fourier transforms.

**4.2.3.6 The MATLAB Application Program Interface (API)**

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**CHAPTER 5**

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 METHODOLOGY**

**Fuzzy C Means Clustering (FCM) Algorithm**

Feature selection helps to reduce the feature space which improves prediction accuracy and minimizes the computation time. It can be done based on clustering process. Clusteringis the process of partitioning or grouping a given set of unlabelled patterns into a number of clusters such that similar patterns are assigned to one cluster. There are two main approaches to clustering-crisp (hard) and fuzzy (soft) clustering. In crisp clustering, boundary between clusters is fully defined but in some cases, the boundaries between clusters cannot be defined clearly. Some patterns may belong to more than one cluster where as fuzzy clustering provides better and more useful method to classify these patterns

To remove prevalent, redundant and noisy features (i.e.) it selects the subset of features that can achieve the best performance in terms of accuracy and also performs dimensionality reduction using **fuzzy c means clustering (FCM) algorithm**. It is based on the concept of “fuzzy c-partition”. It allows one piece of data that belong to two or more clusters then data are bound to each cluster by means of membership function which represents fuzzy behavior of algorithm.

The FCM algorithm and its derivatives have been successfully used in pattern recognition, classification, data mining and image segmentation. The FCM algorithm consists of several execution steps

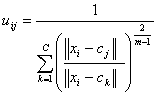
* In the first step, the algorithm selects C initial clusters from the original dataset randomly
* In the later step, after several iterations of the algorithm, final result converges to the actual cluster center

FCM is based on minimization of the following objective function:

Description: C:\Users\aparna\Documents\Clustering - Fuzzy C-means_files\image019.gif,     Description: C:\Users\aparna\Documents\Clustering - Fuzzy C-means_files\image021.gif

where m is any real number greater than 1,uij is the degree of membership of xi in the cluster j, xi is the ith of d-dimensional measured data, cj is the d-dimension center of the cluster and ||\*|| is any norm expressing the similarity between any measured data and the center.

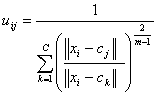
Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership *uij* and the cluster centers *cj* by:

     ,     

This iteration will stop whenDescription: C:\Users\aparna\Documents\Clustering - Fuzzy C-means_files\image027.gif, where Description: C:\Users\aparna\Documents\Clustering - Fuzzy C-means_files\image002.gif is a termination criterion between 0 and 1, whereas *k* are the iteration steps. This procedure converges to a local minimum

**5.1.1 Algorithm**

The algorithm is composed of following steps :

1. Initialize U=[uij] matrix, U(0)
2. At k-step: calculate the centers vectors C(k)=[cj] with U(k)
3. Update U(k) , U(k+1)

If || U(k+1) - U(k)||<Description: C:\Users\aparna\Documents\Clustering - Fuzzy C-means_files\image002.gif then STOP; otherwise return to step 2.

Choosing a good set of initial cluster center is very important for an FCM algorithm. However it is difficult to select a good set of initial cluster randomly. If a good set of initial cluster is chosen, the algorithm may take less iteration to find actual cluster center.

**5.2 MODULE DESIGN:**

**5.2.1PRE-PROCESSING:**

The scanned images are preprocessed. Pre-process technology enhances class 1 semantic content, before images are sent to 3D Segmentation Process. In preprocessing technique, we get four types of images from scanned image. These images undergo fusion process. Semantic content in a class 1 refers to the precise classification of a blob of pixels as a specific character, a line, a part of an image, or noise by implementing noise removal and Pixel grouping enhancement algorithms. Thus pre-processing is done.

**Fig 5.1 Flow Chart For Pre-Processing**

**5.2.2 FUSION:**

This preprocessed image is translated using Fuzzy algorithm method. The process of fusion is performed by using wavelet transformation. Fusion is the process of compressing data, by reducing the dimensions, without any kind of data loss. Since patterns are hard to find in a high dimension data, we opt for fusion. In fused image, pattern should be identified, and express the data in such a way as to highlight their similarities and differences.

**Fig 5.2 Flowchart For Fusion**

**5.2.3 SEGMENTATION**

After fusion it undergoes segmentation. Before segmentation, clustering has to be done. The process of clustering is done by centroid calculation. In centroid calculation, K value is assigned for each cluster. By undergoing some process a loop has been generated. As a result of this loop we may notice that the K value change its location until no more changes are done.

**Fig 5.3 Flowchart For Segmentation Process**

**5.2.4 TUMOR EXTRACTION**

Segmentation is done using K-clustering technique, which separates the vessel structure from background. It is the process of [partitioning](http://en.wikipedia.org/wiki/Partition) a [digital image](http://en.wikipedia.org/wiki/Digital_image) into multiple [regions](http://en.wikipedia.org/wiki/Region) ([sets](http://en.wikipedia.org/wiki/Sets) of [pixels](http://en.wikipedia.org/wiki/Pixel)). Image segmentation is typically used to locate objects and boundaries in images. The result of image segmentation is a set of [regions](http://en.wikipedia.org/wiki/Region) that collectively cover the entire image and thus the white matter region is found from set of regions. The white matter region refers to the tumor part.

**Fig 5.4 Flowchart For Tumor Extraction Process**

**5.3 CODING**

clc;

clear all;

close all;

cd Images

[filename, pathname] = uigetfile('\*.jpg;\*.bmp', 'Pick an Image File');

if isequal(filename,0) || isequal(pathname,0)

warndlg('User pressed cancel')

else

inp\_im1 = imread( filename);

inp\_im1 = imresize(inp\_im1,[256,256]);

[r c p] = size(inp\_im1);

if p==3

inp\_im1 = rgb2gray(inp\_im1);

end

end

[filename, pathname] = uigetfile('\*.jpg;\*.bmp', 'Pick an Image File');

if isequal(filename,0) || isequal(pathname,0)

warndlg('User pressed cancel')

else

inp\_im2 = imread( filename);

inp\_im2 = imresize(inp\_im2,[256,256]);

[r c p] = size(inp\_im2);

if p==3

inp\_im2 = rgb2gray(inp\_im2);

end

end

cd ..

figure('Name','Selected Images');

subplot(1,2,1); imshow(inp\_im1);

subplot(1,2,2); imshow(inp\_im2);

inp\_im1 = double(inp\_im1);

inp\_im2 = double(inp\_im2);

%%%%%%Averaging based pixel level fusion

for ii = 1:size(inp\_im2,1)

for jj = 1:size(inp\_im2,2)

Fim(ii,jj) = (inp\_im1(ii,jj) + inp\_im2(ii,jj))./2;

end

end

figure;

imshow(Fim,[]);

title('Fused Image');

out = Fim;

%%%%%%%image segmentation using clustering model

Input =double(out);

[r c] = size(Input);

Length = r\*c;

wd1=r;

wd2=c;

Dataset = reshape(Input,[Length,1]);

Clusters=4;

Cluster1=zeros(Length,1);

Cluster2=zeros(Length,1);

Cluster3=zeros(Length,1);

Cluster4=zeros(Length,1);

miniv = min(min(Input));

maxiv = max(max(Input));

range = maxiv - miniv;

stepv = range/Clusters;

incrval = stepv;

for i = 1:Clusters

K(i).centroid = incrval;

incrval = incrval + stepv;

end

for i=1:Length

for j = 1:Clusters

temp = Dataset(i);

difference(j) = abs(temp-K(j).centroid);

end

[y,ind]=min(difference);

if ind==1

Cluster1(i) =temp;

end

if ind==2

Cluster2(i) =temp;

end

if ind==3

Cluster3(i) =temp;

end

if ind==4

Cluster4(i) =temp;

end

end

AA1=reshape(Cluster1,[wd1 wd2]);

AA2=reshape(Cluster2,[wd1 wd2]);

AA3=reshape(Cluster3,[wd1 wd2]);

AA4=reshape(Cluster4,[wd1 wd2]);

figure('Name','Segmented Clusters');

subplot(2,2,1); imshow(AA1,[]);

subplot(2,2,2); imshow(AA2,[]);

subplot(2,2,3); imshow(AA3,[]);

subplot(2,2,4); imshow(AA4,[]);

se1 = strel('line',5,0);

se2 = strel('line',5,90);

se3 = strel('disk',2);

imout = imerode(im2bw(AA3),[se1 se2]);

imout1 = imerode(imout,se3);

imout2 = imdilate(imout1,se3);

imout2 = imfill(imout2,'holes');

[fout cnt] = bwlabel(imout2,8);

fout1 = bwareaopen(fout,670);

[fout1 cnt] = bwlabel(fout1,8);

prop = regionprops(fout1,'Area');

for i=1:1:cnt

if prop(i).Area > 4000

fout1(fout1==i)=0;

end

end

figure('Name','Tumor Extraction');

imshow(fout1);

%%%% Tumor Area

seg\_image = fout1;

[r c] = size(seg\_image);

Pcount = 0;

for h = 1:r

for w = 1:c

temp = seg\_image(h,w);

if temp ~= 0

Pcount = Pcount+1;

end

end

end

disp('No of Defect Cells from Brain: ');

disp(Pcount);

%%%%%%% Tumor area by counting no of pixels

tarea = (sqrt(Pcount)).\* 0.264;

disp('Tumor Area (in mm.^2): ');

disp(tarea);

fid = fopen('tumor.txt','wb');

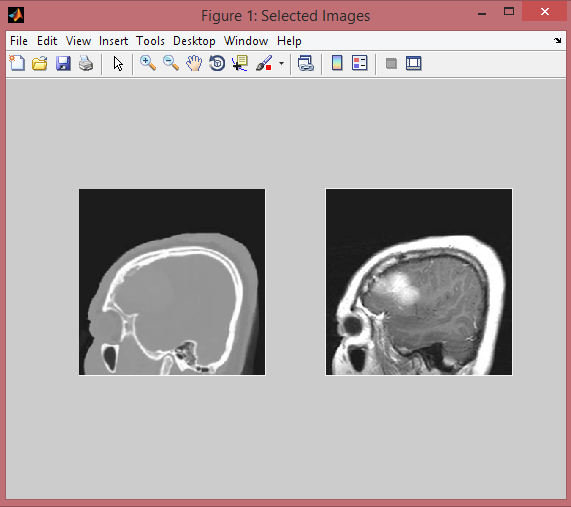
fwrite(fid,num2str(tarea),'char');

fwrite(fid,' mm^2','char');

fclose(fid);

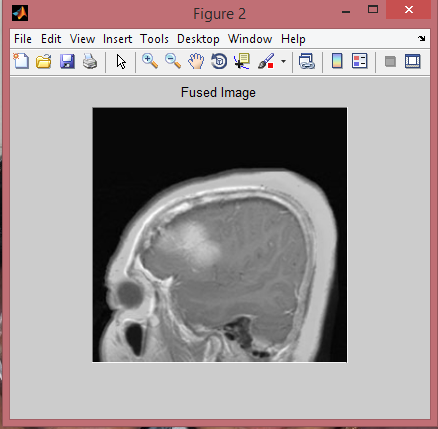
**5.4 SNAPSHOTS**

**5.4.1 Pre-processing**

****

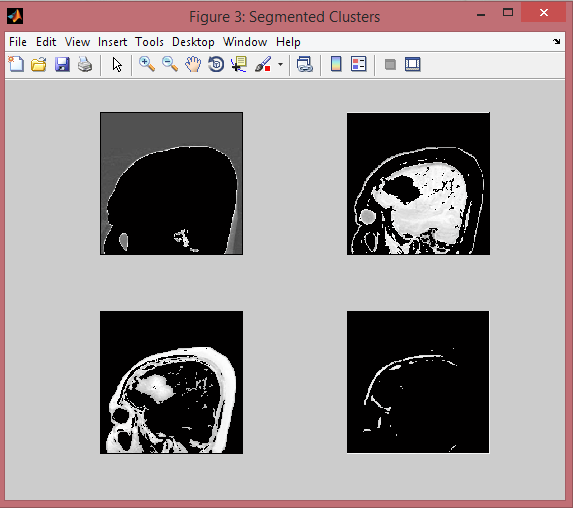
**Fig 5.5 Selected Images**

**5.4.2 Fusion**

****

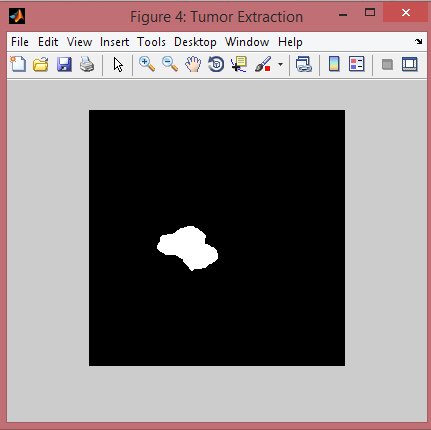
**Fig 5.6 Fused Image**

**5.4.3 Segmentation**

****

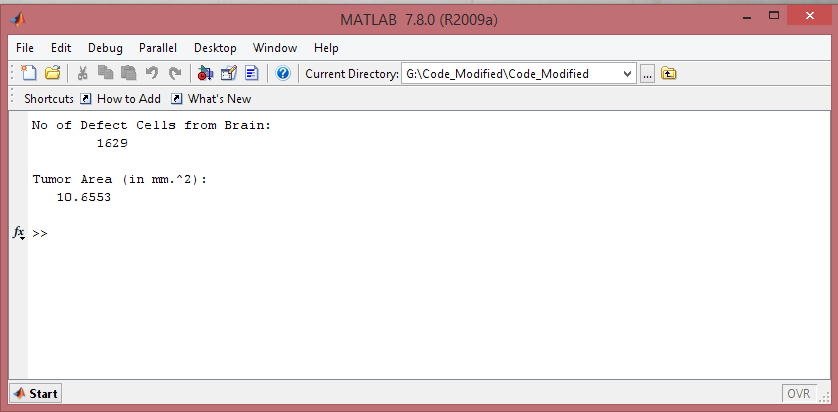
**Fig5.7 Segmented Clusters**

**5.4.4 Tumor Extraction**

****

**Fig 5.8 Tumor Extraction**

**5.4.5 Tumor Area Calculation**

****

**Fig5.9 Tumor Area Calculation**

**CHAPTER 6**

**CHAPTER 6**

**TESTING**

**6.1 INTRODUCTION**

Testing is a process used to help identify the correctness, completeness and quality of developed computer software. There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation, not merely a matter of creating and following rote procedure. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are things the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester. In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

**6.2 NEED FOR TESTING**

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.  It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible. Testing is usually performed for the following purposes:

* **To improve quality.**

In a computerized embedded world, the quality and reliability of software is a matter of life and death. Quality means the conformance to the specified design requirement. Being correct, the minimum requirement of quality, means performing as required under specified circumstances. Debugging, a narrow view of software testing, is performed heavily to find out design defects by the programmer. The imperfection of human nature makes it almost impossible to make a moderately complex program correct the first time. Finding the problems and get them fixed, is the purpose of debugging in programming phase.

* **For Verification & Validation (V&V)**

Verification and Validation (V&V) is another important purpose of testing. Testing can serve as metrics. It is heavily used as a tool in the V&V process. Testers can make claims based on interpretations of the testing results, which either the product works under certain situations, or it does not work. We can also compare the quality among different products under the same specification, based on results from the same test.

We cannot test quality directly, but we can test related factors to make quality visible. Quality has three sets of factors functionality, engineering, and adaptability. These three sets of factors can be thought of as dimensions in the software quality space.

**6.3 TAXONOMY OF TESTING:**

There is a plethora of testing methods and testing techniques, serving multiple purposes in different life cycle phases. Classified by purpose, software testing can be divided into: correctness testing, performance testing, reliability testing and security testing. Classified by life-cycle phase, software testing can be classified into the following categories: requirements phase testing, design phase testing, program phase testing, evaluating test results, installation phase testing, acceptance testing and maintenance testing. By scope, software testing can be categorized as follows: unit testing, component testing, integration testing, and system testing.

**6.3.1 Correctness testing**

Correctness is the minimum requirement of software, the essential purpose of testing. Correctness testing will need some type of oracle, to tell the right behavior from the wrong one. The tester may or may not know the inside details of the software module under test, e.g. control flow, data flow, etc. Therefore, either a white-box point of view or black-box point of view can be taken in testing software. We must note that the black-box and white-box ideas are not limited in correctness testing only.

**6.3.2 Black-box testing**

The black-box approach is a testing method in which test data are derived from the specified functional requirements without regard to the final program structure. It is also termed data-driven, input/output driven or requirements-based testing. Because only the functionality of the software module is of concern, black-box testing also mainly refers to functional testing a testing method emphasized on executing the functions and examination of their input and output data. The tester treats the software under test as a black box -- only the inputs, outputs and specification are visible, and the functionality is determined by observing the outputs to corresponding inputs. In testing, various inputs are exercised and the outputs are compared against specification to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered.

It is obvious that the more we have covered in the input space, the more problems we will find and therefore we will be more confident about the quality of the software. Ideally we would be tempted to exhaustively test the input space. But as stated above, exhaustively testing the combinations of valid inputs will be impossible for most of the programs, let alone considering invalid inputs, timing, sequence, and resource variables. Combinatorial explosion is the major roadblock in functional testing. To make things worse, we can never be sure whether the specification is either correct or complete. Due to limitations of the language used in the specifications (usually natural language), ambiguity is often inevitable.

Even if we use some type of formal or restricted language, we may still fail to write down all the possible cases in the specification. Sometimes, the specification itself becomes an intractable problem: it is not possible to specify precisely every situation that can be encountered using limited words. And people can seldom specify clearly what they want -- they usually can tell whether a prototype is, or is not, what they want after they have been finished. Specification problems contribute approximately 30% of all bugs in software.

**6.3.3 White-box testing**

Contrary to black-box testing, software is viewed as a white-box, or glass-box in white-box testing, as the structure and flow of the software under test are visible to the tester. Testing plans are made according to the details of the software implementation, such as programming language, logic, and styles. Test cases are derived from the program structure. White-box testing is also called glass-box testing, logic-driven testing or design-based testing.

There are many techniques available in white-box testing, because the problem of intractability is eased by specific knowledge and attention on the structure of the software under test. The intention of exhausting some aspect of the software is still strong in white-box testing, and some degree of exhaustion can be achieved, such as executing each line of code at least once (statement coverage), traverse every branch statements (branch coverage), or cover all the possible combinations of true and false condition predicates (Multiple condition coverage).

Control-flow testing, loop testing, and data-flow testing, all maps the corresponding flow structure of the software into a directed graph. Test cases are carefully selected based on the criterion that all the nodes or paths are covered or traversed at least once. By doing so we may discover unnecessary "dead “code that is of no use, or never get executed at all, which cannot be discovered by functional testing.

In mutation testing, the original program code is perturbed and many mutated programs are created, each contains one fault. Each faulty version of the program is called a mutant. Test data are selected based on the effectiveness of failing the mutants. The more mutants a test case can kill, the better the test case is considered. The problem with mutation testing is that it is too computationally expensive to use. The boundary between black-box approach and white-box approach is not clear-cut. Many testing strategies mentioned above, may not be safely classified into black-box testing or white-box testing. It is also true for transaction-flow testing, syntax testing, finite-state testing, and many other testing strategies not discussed in this text. One reason is that all the above techniques will need some knowledge of the specification of the software under test. Another reason is that the idea of specification itself is broad -- it may contain any requirement including the structure, programming language, and programming style as part of the specification content.

**6.3.4** **Performance testing**

Not all software systems have specifications on performance explicitly. But every system will have implicit performance requirements. The software should not take infinite time or infinite resource to execute. "Performance bugs" sometimes are used to refer to those design problems in software that cause the system performance to degrade.

Performance has always been a great concern and a driving force of computer evolution. Performance evaluation of a software system usually includes: resource usage, throughput, and stimulus-response time and queue lengths detailing the average or maximum number of tasks waiting to be serviced by selected resources. Typical resources that need to be considered include network bandwidth requirements, CPU cycles, disk space, disk access operations, and memory usage. The goal of performance testing can be performance bottleneck identification, performance comparison and evaluation, etc. The typical method of doing performance testing is using a benchmark -- a program, workload or trace designed to be representative of the typical system usage.

**6.3.5 Reliability testing**

Software reliability refers to the probability of failure-free operation of a system. It is related to many aspects of software, including the testing process. Directly estimating software reliability by quantifying its related factors can be difficult. Testing is an effective sampling method to measure software reliability. Guided by the operational profile, software testing (usually black-box testing) can be used to obtain failure data, and an estimation model can be further used to analyze the data to estimate the present reliability and predict future reliability. Therefore, based on the estimation, the developers can decide whether to release the software, and the users can decide whether to adopt and use the software. Risk of using software can also be assessed based on reliability information. Advocates that the primary goal of testing should be to measure the dependability of tested software.

There is agreement on the intuitive meaning of dependable software: it does not fail in unexpected or catastrophic ways. Robustness testing and stress testing are variances of reliability testing based on this simple criterion.The robustness of a software component is the degree to which it can function correctly in the presence of exceptional inputs or stressful environmental conditions.  Robustness testing differs with correctness testing in the sense that the functional correctness of the software is not of concern. It only watches for robustness problems such as machine crashes, process hangs or abnormal termination.

The oracle is relatively simple, therefore robustness testing can be made more portable and scalable than correctness testing. This research has drawn more and more interests recently, most of which uses commercial operating systems as their target. Stress testing, or load testing, is often used to test the whole system rather than the software alone. In such tests the software or system are exercised with or beyond the specified limits. Typical stress includes resource exhaustion, bursts of activities, and sustained high loads.

**6.3.6 Security testing**

Software quality, reliability and security are tightly coupled. Flaws in software can be exploited by intruders to open security holes. With the development of the Internet, software security problems are becoming even more severe.Many critical software applications and services have integrated security measures against malicious attacks. The purpose of security testing of these systems include identifying and removing software flaws that may potentially lead to security violations, and validating the effectiveness of security measures. Simulated security attacks can be performed to find vulnerabilities.

**6.3.7 Unit testing**

Unit testing of software applications is done during the development (coding) of an application. The objective of unit testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing increases confidence in changing/maintaining code. If good unit tests are written and if they are run every time any code is changed, the likelihood of any defects due to the change being promptly caught is very high. If unit testing is not in place, the most one can do is hope for the best and wait till the test results at higher levels of testing are out. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

**6.3.8 Integration testing**

Integration testing is a software testing methodology used to test individual software components or units of code to verify interaction between various software components and detect interface defects. Components are tested as a single group or organized in an iterative manner. After the integration testing has been performed on the components, they are readily available for system testing. Integration is a key software development life cycle (SDLC) strategy. Generally, small software systems are integrated and tested in a single phase, whereas larger systems involve several integration phases to build a complete system, such as integrating modules into low-level subsystems for integration with larger subsystems. Integration testing encompasses all aspects of a software system's performance and functionality.

**6.3.9 System testing**

The [process](http://www.businessdictionary.com/definition/process.html) of performing a variety of [tests](http://www.businessdictionary.com/definition/test.html) on a [system](http://www.businessdictionary.com/definition/system.html) to explore functionality or to identify the system problems. System [testing](http://www.businessdictionary.com/definition/testing.html) is usually [required](http://www.businessdictionary.com/definition/required.html) before and after a system is put in place. A series of [systematic](http://www.businessdictionary.com/definition/systematic.html) [procedures](http://www.businessdictionary.com/definition/procedure.html) are referred while testing is being performed. These procedures tell the tester how the system should perform and where [common mistakes](http://www.businessdictionary.com/definition/common-mistake.html) may be found. Testers usually try to "break the system" by entering [data](http://www.businessdictionary.com/definition/data.html) that may cause the system to malfunction or [return](http://www.businessdictionary.com/definition/return.html) incorrect [information](http://www.businessdictionary.com/definition/information.html). For example, a tester may put in a city in a [search engine](http://www.businessdictionary.com/definition/search-engine.html) designed to only accept [states](http://www.businessdictionary.com/definition/state.html), to see how the system will respond to incorrect [input](http://www.businessdictionary.com/definition/input.html).

**6.4 TEST REPORT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case number | Input | Expected Output | Output Obtained | Pass/Fail | Remark |
| 1. | Description: C:\Users\DELL\Desktop\Image Fusion\Image Fusion\fusionimage\ct.jpg  Description: C:\Users\DELL\Desktop\Image Fusion\Image Fusion\fusionimage\mri.jpg |  |  | Pass | Tumour is extracted. |
| 2. | **Description: C:\Users\DELL\Desktop\Brain\3d brain\3d brain\segment3dbrain\segment3dbrain\bin\Debug\mri8.jpg** |  |  | Pass | Tumour is extracted. |
| 3. | **Description: C:\Users\DELL\Desktop\Brain\3d brain\3d brain\segment3dbrain\segment3dbrain\bin\Debug\3D.bmp** |  |  | Pass | Tumour is extracted. |
| 4. | Description: C:\Users\DELL\Desktop\Image Fusion\Image Fusion\fusionimage\ct.jpg  Description: C:\Users\DELL\Desktop\Image Fusion\Image Fusion\fusionimage\mri.jpg |  |  | Fail | Tumour is not extracted properly. |

**Table 1- Test Report**

**CHAPTER 7**

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSION:**

This algorithm is completely automated which does not need any user interaction, not even to identify a start point. The main branches of the vessel structure is identified by randomly selecting the seed point. The precise segmentation is not obtained by random selection.. Experimental result shows that selecting centroid by our algorithm can lead to a better clustering**.** Our ongoing research will focus the development of methods to segment coronary arteries in a sequence of angiographic images while preserving the topology of the vessel structure.

**7.2 FUTURE ENHANCEMENT:**

The improved K-means algorithm provides a better solution for handling large scale data. It is done by selecting the initial clustering purposefully by reducing the sensitivity to isolated point. It avoids clustering problem. The initial seed point is located easily using this technique and will enhance the output more accurate with high-resolution. By analysing various techniques we can find out which technique will gives higher resolution.

**REFERENCES**

[1] Meiyan Huang, Wei Yang, Yao wu, Jun Jiang, Wufan Chen, Senior Member, IEEE, and Qianjin Feng\*, Member, IEEE “Brain tumor Segmentation Based on Local Independent Projection-based Classification,” IEEE Transactions on Biomedical Engineering Sep2014.

[2]J.G.Sled, A.P.Zijdenbos, and A.C.Evans,” A Non-parametric method for automatic correction of intensity non-uniformity in MRI data,” IEEE Transactions on Medical Imaging, vol. 17, no. 1, pp. 87-97, Feb1998

[3]H. Loe, D.S. Lee, H. Kang,B.N. Kim,and M.K.Chung,” Sparse brain network recovery under compressed sensing,” IEEE Transactions on Medical Imaging, vol.30, no.5,pp. 1154-1165, May 2011.

[4]M.B.Cuadra, C.Pollo, A.Bardera, O.Cuisenaire, J.G.Villemure, and J.P.Thiran, “Atlas-based segmentation of pathological MR brain images using a lesion growth,” IEEE Transactions on Medical Imaging, vol. 23, no. 10,pp. 1301-1314, 2004.

[5] Dina Aboul Dahab, Samy S. A. Ghoniemy, Gamal M. Selim,”Automated Brain Tumor Detection and Identification Using Image Processing and Probabilistic Neural Network Techniques,”vol. 1, Oct2012

[6] J.Wright, A.Y.Yang, A.Ganesh, S.S.Sastry, and Y.Ma, “Robust face recognition via sparse representation,” IEEE Trans Pattern anal mach intell ,vol. 31, no. 2, pp. 210-227, Feb 2009

[7] S.Bauer, C. Seiler, T. Bardyn, P. Buechler, and M.Reyes, “Atlas-based segmentation of brain tumor images using a markov random field-based tumor growth model and non-rigid registration,” in Proceedings of the 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina,2010.

[8] M. Qiguang and W. Baoshu, “The contourlet transform for image fusion,” Proc. SPIE, vol. 6242, pp. 62420Z-1–62420Z-8, Apr.2006.

[9] M. H. Asmare, V. S. Asirvadam, and L. Iznita, “Multi-sensor image enhancement and fusion for vision clarity using contourlet transform,” inProc. IEEE Int. Conf. Inf. Manage. Eng., Apr. 2009, pp. 352–356.

[10] D.A. Gutman, L.A. Cooper, S.N. Hwang, C.A. Holder, J. Gao, T.D. Aurora, W.D. Dunn, Jr., L. Scarpace, T. Mikkelsen, R. Jain, et al., “MR imaging predictors of molecular profile and survival: multi-institutional study of the TCGA glioblastoma data set, “ Radiology, vol. 267, no. 2,pp. 560-9, May2013.