**Brain Tumor Segmentation Based on Local Independent Projection-Based Classification**

Project report submitted in partial fulfillment of the requirements for the award of the degree of MSc. in IT (AI,ML/Data Science and Cloud Computing)

**Submitted by**

**JUNAIDUL ISLAM**

**REG NO : 20MSIC112**

**Under the guidance of**

**DR. ASHVINI K BABALESHWAR**

**ASSISTANT PROFESSOR**

**DEPT. OF COMPUTER SCIENCE**

**SCHOOL OF COMPUTATIONAL SCIENCE AND IT**

****

**BANGALORE – 560049**

**2022-JUNE**

**DECLARATION BY THE STUDENT**

I, JUNAIDUL ISLAM hereby declare that the project report, entitled “Brain Tumor Segmentation Based on Local Independent Projection-Based Classification”, submitted to Garden City University, in partial fulfillment of the requirements for the award of the Degree of of MSc. in IT (AI,ML/Data Science and Cloud Computing) is a record of original and independent research work done by me during 2022. The project was completed under the supervision and guidance of ASHVINI K BABALESHWAR ASSISTANT PROFESSOR, Department of COMPUTER SCIENCE and it has not formed the basis for the award of any degree/diploma/certificate or any similar title to any candidate in any University.

**Signature of the Candidate**

Place: Bangalore

Date: 23/06/2022

Register Number: 20MSIC111

**CERTIFICATE**

This is to certify that the project work entitled “Brain Tumor Segmentation Based on Local Independent Projection-Based Classification” is a bonafide work of Mr/Ms Junaidul Islam bearing University Register Number 20MSIC112 and is being submitted in partial fulfillment for the award of MSc. in IT (AI,ML/Data Science and Cloud Computing) at Garden City University.

**Date:** 23/06/2022

**Signature of the Guide**

**Place:** Bangalore

**Name:**

**Designation:**

**Signature of the Head of the Department**

**ACKNOWLEDGEMENT**

**Order:**

**Chancellor & GCU**

**VC**

**Registrar**

**Dean/HoD**

**Guide**

**\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE .NO** |
| 1 | Abstract | 2 |
| 2 | Introduction | 3 |
| 3 | Existing system | 5 |
| 4 | Disadvantages | 6 |
| 5 | Proposed system | 7 |
| 6 | Advantages | 8 |
| 7 | System Architecture | 9 |
| 8 | Flow Diagram | 10 |
| 9 | Activity Diagram | 11 |
| 10 | Use-case Diagram | 12 |
| 11 | Sequential Diagram | 13 |
| 12 | Testing Of Product | 14 |
| 13 | Modules | 18 |
| 14 | Modules Description | 18 |
| 15 | Literature Survey | 21 |
| 16 | Software Requirements | 31 |
| 17 | Hardware Requirements | 31 |
| 18 | H/W&S/W Description | 35 |
| 19 | Conclusion | 39 |
| 20 | References | 40 |

**ABSTRACT**

Now a day, Brain tumor segmentation is an important procedure for early tumor diagnosis and radiotherapy planning. Although numerous brain tumor segmentation methods have been presented, enhancing tumor segmentation methods is still challenging because brain tumor MRI images exhibit complex characteristics, such as high diversity in tumor appearance and ambiguous tumor boundaries. To address this problem, we propose a novel automatic tumor segmentation method and then segment the tumor region from the non-tumor region. By segmenting the tumor region from the non-tumor region, we have easily classify the affected region by using the scanned image itself. Additionally, the local independent projection-based classification (LIPC) method is used to classify each voxel into different classes. A novel classification framework is derived by introducing the local independent projection into the classical classification model. Locality is important in the calculation of local independent projections for LIPC. Locality is also considered in determining whether local anchor embedding is more applicable in solving linear projection weights compared with other coding methods. Moreover, LIPC considers the data distribution of different classes by learning a softmax regression model, which can further improve classification performance. In this study, 80 brain tumor MRI images with ground truth data are used as training data and 40 images without ground truth data are used as testing data. The segmentation results of testing data are evaluated by an online evaluation tool. The average dice similarities of the proposed method for segmenting complete tumor, tumor core, and contrast-enhancing tumor on real patient data are 0.84, 0.685, and 0.585, respectively. These results are comparable to other state-of-the-art methods.

**INTRODUCTION**

In Medical diagnosis, through Magnetic Resonance Images Robustness and accuracy of the Prediction algorithms are very important, because the result is crucial for treatment of Patients. Brain tumor segmentation is one of the crucial procedures in surgical and treatment planning. However, at present, brain tumor segmentation in brain tumor images is mostly performed manually in clinical practice. Apart from being time consuming, manual brain tumor delineation is difficult and depends on the individual operator. Low-level operations, such as thresholding, edge detection, and morphological techniques, are fast and can be easily adjusted. However, the tumor segmentation performance of these methods highly depends on evident difference in the intensities between tumor and non-tumor regions. Watershed and region growing approaches are simple and consistently produce complete boundaries. However, these two methods are sensitive to noise, which is a common problem in the intensity-based method. Moreover, most intensity-based methods tend to oversegment tumors because of the weak and diffused edges caused by edema. The healthy human brain is largely symmetric across the mid-sagittal plane. The asymmetric analysis method for tumor segmentation is based on the principle that tumors, which appear in one of the cerebral hemispheres, can cause asymmetry between the left and right cerebral hemispheres. This asymmetry can be detected, and tumors can be roughly located in the corresponding cerebral hemisphere. The asymmetric analysis method can hasten the tumor detection and segmentation process because tumor segmentation is implemented in one of the cerebral hemispheres. However, accurately finding the mid-sagittal plane is a challenging and time-consuming task. More importantly, asymmetry analysis may not be useful when a tumor is located across the mid-sagittal plane. Atlas-based segmentation methods have been extensively investigated. Brain atlases can provide important data prior to tumor segmentation enhancement by measuring the difference between abnormal and normal brains. However, the deformable registration of the brain atlas to brain images with tumor is an extremely challenging task because of the intensity variations around the tumor caused by edema and the deformations of healthy tissue morphology caused by the tumor mass effect. In a previous study, affine registration is used to align the atlas to the tumor image data. When a large brain structure deformation appears, the misalignment issues are noted on the aligned atlas, which may significantly decrease segmentation accuracy. The contour/surface evolution method has been widely used for the tumor segmentation of 2-D/3-D data. This method can be represented implicitly as a level set function or explicitly as an active contour model/snake function. Compared with the parametric active contour model, the level set method can represent contours with complex topology and handle topological changes, such as splitting and merging in a natural and efficient way. Furthermore, the extension of the level set method to 3D is straightforward and does not require additional machinery. However, the contour/surface evolution method does not easily determine the initializations and tune the parameters even when 3-D level set surfaces are used. Graph-based seeded segmentation framework is one of the popular methods among interactive algorithms. Graph-based seeded segmentation is a global optimization approach, which showed outstanding performance for tumor segmentation. However, this method needs manual seed selection in different tissues, and distinguishing different tissues in the tumor is difficult during the selection of initial seeds for different tissues. In a previous study, a cellular automata-based seeded method, called tumor-cut, has been presented for brain tumor segmentation. This method only requires the user to draw a line over the largest visible tumor diameter. Although this initial seed selection strategy can reduce manual interaction and decrease the sensitivity of the method to initialization, this procedure may not include all tumor areas within the volume of interest along the depth direction, thus leading to tumor undersegmentation. Unsupervised learning method, such as k-means and fuzzy clustering, has become popular for brain tumor segmentation in recent years. The fuzzy method considers that medical images are inherently fuzzy, so it is a very strong tool for medical image processing. Furthermore, the fuzzy method can capture pixel proximity in the same objective region without a training step. However, most fuzzy methods work well only for tumors that present hyper-intensity and exhibit poor performance on segmenting nonenhanced tumors. These conditions are due to the fact that these fuzzy methods typically use intensity-based method, such as thresholding and morphological operations, as pre- or postprocessing. Supervised classification learning method is widely used in tumor segmentation. Well-trained classifiers can extract discriminative information from the training data and estimate the label of each voxel in a testing volume. However, the traditional classification methods classify each voxel into different classes without considering the spatial correlation between current and nearby voxels. This method may not obtain a global optimized result. To address this problem, a classification method is generally combined with a regularization step. The regularization step can be implemented by modeling the boundary or by applying a variant of a random field spatial prior (MRF/CRF). In the previous studies, context-aware spatial features and the probabilities obtained by tissue-specific Gaussian mixture models are used as inputs for classifiers, and satisfied segmentation results are achieved without using posthoc regularization. We propose a novel classification method, named local independent projection-based classification (LIPC), for brain tumor segmentation without using explicit regularization.

**IMAGE PROCESSING**

* 1. **What is an image?**

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows.



***Figure 1: An image — an array or a matrix of pixels arranged in columns and rows.***

In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

***Figure 2: Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 tones or greyscales.***

A normal grayscale image has 8 bit colour depth = 256 grayscales. A “true colour” image has 24 bit colour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours.



***Figure 3: A true-colour image assembled from three grayscale images coloured red, green and blue. Such an image may contain up to 16 million different colours.***

Some grayscale images have more grayscales, for instance 16 bit = 65536 grayscales. In principle three grayscale images can be combined to form an image with 281,474,976,710,656 grayscales.

There are two general groups of ‘images’: vector graphics (or line art) and bitmaps (pixel-based or ‘images’).

Some of the most common file formats are:

GIF — an 8-bit (256 colour), non-destructively compressed bitmap format. Mostly used for web. Has several sub-standards one of which is the animated GIF.

JPEG — a very efficient (i.e. much information per byte) destructively compressed 24 bit (16 million colours) bitmap format. Widely used, especially for web and Internet (bandwidth-limited).

TIFF — the standard 24 bit publication bitmap format. Compresses non-destructively with, for instance, Lempel-Ziv-Welch (LZW) compression.

PS — Postscript, a standard vector format. Has numerous sub-standards and can be difficult to transport across platforms and operating systems.

PSD – a dedicated Photoshop format that keeps all the information in an image including all the layers.

Pictures are the most common and convenient means of conveying or transmitting information. A picture is worth a thousand words. Pictures concisely convey information about positions, sizes and inter relationships between objects. They portray spatial information that we can recognize as objects. Human beings are good at deriving information from such images, because of our innate visual and mental abilities. About 75% of the information received by human is in pictorial form. An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk or CD-ROM. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer. Once the image has been digitized, it can be operated upon by various image processing operations.

Image processing operations can be roughly divided into three major categories, Image Compression, Image Enhancement and Restoration, and Measurement Extraction. It involves reducing the amount of memory needed to store a digital image. Image defects which could be caused by the digitization process or by faults in the imaging set-up (for example, bad lighting) can be corrected using Image Enhancement techniques. Once the image is in good condition, the Measurement Extraction operations can be used to obtain useful information from the image. Some examples of Image Enhancement and Measurement Extraction are given below. The examples shown all operate on 256 grey-scale images. This means that each pixel in the image is stored as a number between 0 to 255, where 0 represents a black pixel, 255 represents a white pixel and values in-between represent shades of grey. These operations can be extended to operate on colour images. The examples below represent only a few of the many techniques available for operating on images. Details about the inner workings of the operations have not been given, but some references to books containing this information are given at the end for the interested reader.

**Images and pictures**

As we mentioned in the preface, human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify and classify them, but we can scan for differences, and obtain an overall rough feeling for a scene with a quick glance. Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colors; we can process a large amount of visual information very quickly.

However, the world is in constant motion: stare at something for long enough and it will change in some way. Even a large solid structure, like a building or a mountain, will change its appearance depending on the time of day (day or night); amount of sunlight (clear or cloudy), or various shadows falling upon it. We are concerned with single images: snapshots, if you like, of a visual scene. Although image processing can deal with changing scenes, we shall not discuss it in any detail in this text. For our purposes, an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging. Even if the picture is not immediately recognizable, it will not be just a random blur.

Image processing involves changing the nature of an image in order to either

1. Improve its pictorial information for human interpretation,

2. Render it more suitable for autonomous machine perception.

We shall be concerned with digital image processing, which involves using a computer to change the nature of a digital image. It is necessary to realize that these two aspects represent two separate but equally important aspects of image processing. A procedure which satisfies condition, a procedure which makes an image look better may be the very worst procedure for satisfying condition. Humans like their images to be sharp, clear and detailed; machines prefer their images to be simple and uncluttered.

**Images and digital images**

Suppose we take an image, a photo, say. For the moment, lets make things easy and suppose the photo is black and white (that is, lots of shades of grey), so no colour. We may consider this image as being a two dimensional function, where the function values give the brightness of the image at any given point. We may assume that in such an image brightness values can be any real numbers in the range (black) (white).

A digital image from a photo in that the values are all discrete. Usually they take on only integer values. The brightness values also ranging from 0 (black) to 255 (white). A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements, or more simply pixels. The pixels surrounding a given pixel constitute its neighborhood. A neighborhood can be characterized by its shape in the same way as a matrix: we can speak of a neighborhood, Except in very special circumstances, neighborhoods have odd numbers of rows and columns; this ensures that the current pixel is in the Centre of the neighborhood.

**Image Processing Fundamentals:**

**Pixel:**

In order for any digital computer processing to be carried out on an image, it must first be stored within the computer in a suitable form that can be manipulated by a computer program. The most practical way of doing this is to divide the image up into a collection of discrete (and usually small) cells, which are known as pixels. Most commonly, the image is divided up into a rectangular grid of pixels, so that each pixel is itself a small rectangle. Once this has been done, each pixel is given a pixel value that represents the color of that pixel. It is assumed that the whole pixel is the same color, and so any color variation that did exist within the area of the pixel before the image was discretized is lost. However, if the area of each pixel is very small, then the discrete nature of the image is often not visible to the human eye.

Other pixel shapes and formations can be used, most notably the hexagonal grid, in which each pixel is a small hexagon. This has some advantages in image processing, including the fact that pixel connectivity is less ambiguously defined than with a square grid, but hexagonal grids are not widely used. Part of the reason is that many image capture systems (e.g. most CCD cameras and scanners) intrinsically discretize the captured image into a rectangular grid in the first instance.

**Pixel Connectivity**

The notation of pixel connectivity describes a relation between two or more pixels. For two pixels to be connected they have to fulfill certain conditions on the pixel brightness and spatial adjacency.

First, in order for two pixels to be considered connected, their pixel values must both be from the same set of values *V*. For a grayscale image, *V* might be any range of gray levels, *e.g.* *V= {22, 23,...40}*, for a binary image we simple have *V={1}*.

To formulate the adjacency criterion for connectivity, we first introduce the notation of *neighborhood*. For a pixel *p* with the coordinates *(x,y)* the set of pixels given by:

Eqn:eqnla1

is called its *4-neighbors*. Its *8-neighbors* are defined as

Eqn:eqnla2

From this we can infer the definition for *4-* and *8-connectivity*:

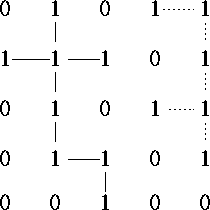
Two pixels *p* and *q*, both having values from a set *V* are *4*-connected if *q* is from the set Eqn:eqnlan4and *8*-connected if *q* is fromEqn:eqnlan8.

General connectivity can either be based on *4*- or *8*-connectivity; for the following discussion we use *4*-connectivity.

A pixel *p* is connected to a pixel *q* if *p* is *4*-connected to *q* or if *p* is *4*-connected to a third pixel which itself is connected to *q*. Or, in other words, two pixels *q* and *p* are connected if there is a path from *p* and *q* on which each pixel is *4*-connected to the next one.

A set of pixels in an image which are all *connected* to each other is called a *connected component*. Finding all connected components in an image and marking each of them with a distinctive label is called connected component labeling.

An example of a binary image with two connected components which are based on *4*-connectivity can be seen in Figure 1. If the connectivity were based on *8*-neighbors, the two connected components would merge into one.



**Figure 1** Two connected components based on *4*-connectivity.

# Pixel Values

Each of the pixels that represents an image stored inside a computer has a pixel value which describes how bright that pixel is, and/or what color it should be. In the simplest case of [binary](http://homepages.inf.ed.ac.uk/rbf/HIPR2/binimage.htm) images, the pixel value is a 1-bit number indicating either foreground or background. For a gray scale images, the pixel value is a single number that represents the brightness of the pixel. The most common motepixel format is the byte image, where this number is stored as an 8-bit integer giving a range of possible values from 0 to 255. Typically zero is taken to be black, and 255 is taken to be white. Values in between make up the different shades of gray.

To represent color images, separate red, green and blue components must be specified for each pixel (assuming an RGB color space), and so the pixel `value' is actually a vector of three numbers. Often the three different components are stored as three separate `grayscale' images known as color planes (one for each of red, green and blue), which have to be recombined when displaying or processing. Multispectral Images can contain even more than three components for each pixel, and by extension these are stored in the same kind of way, as a vector pixel value, or as separate color planes.

The actual grayscale or color component intensities for each pixel may not actually be stored explicitly. Often, all that is stored for each pixel is an index into a colour map in which the actual intensity or colors can be looked up.

Although simple 8-bit integers or vectors of 8-bit integers are the most common sorts of pixel values used, some image formats support different types of value, for instance 32-bit signed integers or floating point values. Such values are extremely useful in image processing as they allow processing to be carried out on the image where the resulting pixel values are not necessarily 8-bit integers. If this approach is used then it is usually necessary to set up a color map which relates particular ranges of pixel values to particular displayed colors.

**Pixels, with a neighborhood:**

****

**Color scale**

The two main color spaces are **RGB** and **CMYK.**

**RGB**

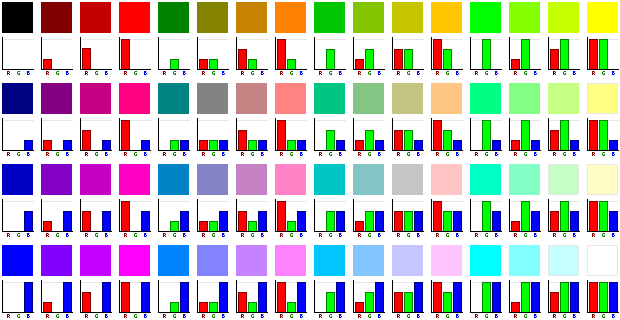
The **RGB color model** is an [additive color model](http://en.wikipedia.org/wiki/Additive_color) in which [red](http://en.wikipedia.org/wiki/Red), [green](http://en.wikipedia.org/wiki/Green), and [blue](http://en.wikipedia.org/wiki/Blue) light are added together in various ways to reproduce a broad array of [colors](http://en.wikipedia.org/wiki/Color). RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production.

The secondary colors of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colors (**red, green** or **blue**) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white. [figure4]



Figure [4]: The additive model of RGB. Red, green, and blue are the primary stimuli for human color perception and are the primary additive colors.

To see how different RGB components combine together, here is a selected repertoire of colors and their respective relative intensities for each of the red, green, and blue components:

[](http://en.wikipedia.org/wiki/Image:RGB-combinations.png)

**\*Typical uses of MATLAB include:-**

- Math and computation.

-Algorithm development

-Data acquisition

-Modeling, simulation, and prototyping

-Data analysis, exploration, and visualization

-Scientific and engineering graphics

-Application development, including graphical user interface building

**Some applications:**

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

1. Medicine

* Inspection and interpretation of images obtained from X-rays, MRI or CAT scans,
* Analysis of cell images, of chromosome karyotypes.

2. Agriculture

* Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops,
* Inspection of fruit and vegetables distinguishing good and fresh produce from old.

3. Industry

* Automatic inspection of items on a production line,
* Inspection of paper samples.

4. Law enforcement

* Fingerprint analysis,
* Sharpening or de-blurring of speed-camera images.

**Aspects of image processing:**

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.

* **Image enhancement**: This refers to processing an image so that the result is more suitable for a particular application.

Example include:

* sharpening or de-blurring an out of focus image,
* highlighting edges,
* improving image contrast, or brightening an image,
* Removing noise.
* **Image restoration**. This may be considered as reversing the damage done to an image by a known cause, for example:
* removing of blur caused by linear motion,
* removal of optical distortions,
* Removing periodic interference.
* **Image segmentation**. This involves subdividing an image into constituent parts, or isolating certain aspects of an image:
* circles, or particular shapes in an image,
* In an aerial photograph, identifying cars, trees, buildings, or roads.

These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

**An image processing task**

We will look in some detail at a particular real-world task, and see how the above classes may be used to describe the various stages in performing this task. The job is to obtain, by an automatic process, the postcodes from envelopes. Here is how this may be accomplished:

* **Acquiring the image**: First we need to produce a digital image from a paper envelope. This can be done using either a CCD camera, or a scanner.
* **Preprocessing:** This is the step taken before the major image processing task. The problem here is to perform some basic tasks in order to render the resulting image more suitable for the job to follow. In this case it may involve enhancing the contrast, removing noise, or identifying regions likely to contain the postcode.
* **Segmentation**: Here is where we actually get the postcode; in other words we extract from the image that part of it which contains just the postcode.
* **Representation and description** these terms refer to extracting the particular features which allow us to differentiate between objects. Here we will be looking for curves, holes and corners which allow us to distinguish the different digits which constitute a postcode.
* **Recognition and interpretation**: This means assigning labels to objects based on their descriptors (from the previous step), and assigning meanings to those labels. So we identify particular digits, and we interpret a string of four digits at the end of the address as the postcode.

**EXISTING SYSTEM**

Segmentation is the process to segregate the portion in digital image process. Brain tumor is one of the common diseases which are treated in medical science. Detection of brain tumor in early stages can enhance the prevention mechanism to stronger level. Detection of brain tumor from digital image processing techniques is one of the most essential parts for work. In our research we will work on segmentation of brain tumor area for digital images. The detection of brain tumor has been done with Magnetic resonance imaging (MRI) process by doctors. We will proceed with quantization process for images and will focus on clustering process of different detecting areas of the brain and finally with ROI technique we will detect the brain tumor and image will reflect the segregated portion of brain tumor. Make decisions based on local pixel information and are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness. The image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances belong to different groups. Usually referred to as the active contour model, starts with some initial boundary shape represented in the form of spline curves, and iteratively modifies it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an “elastic” contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task. Many merging methods of segmentation use a method called region growing to merge adjacent single pixel segments into one segment. Region growing needs a set of starting pixels13 called seeds. The region growing process consists of picking a seed from the set, investigating all 4-connected neighbors of this seed, and merging suitable neighbors to the seed. The seed is then removed from the seed set, and all merged neighbors are added to the seed set. The region growing process continues until the seed set is empty. We will assume we have the „larger size‟ segments as meant above available, but we still have an over segmentation of the image, so we still need to do region merging to obtain a proper segmentation. Where region merging is an agglomerative approach, region splitting is divise. We mentioned before that this difference makes that the two approaches are not opposites, but fundamentally different problems; the merging of two segments is straightforward, but the splitting of a segment requires that suitable sub-segments are established to split the original segment into. The problem of how to split a segment is of course itself a segmentation problem, and we can treat it as such: any segmentation method can be applied to the segment to establish sub-segments. Besides the hierarchical level, there is no intrinsic difference.

**DISADVANTAGE:**

* Even if a rich set of manual segmentations are available, they may not reflect the ground truth and the true gold standard may need to be estimated.
* Validation is typically not performed for the segmentations of non-tumor structures since manual segmentations of edema and the healthy brain tissue are very challenging tasks and have a high degree of variability.

**PROPOSED SYSTEM**

The project proposes a segmenting and detecting tumor by using spatial fuzzy clustering algorithm for Magnetic Resonance (MRI) images to detect the Brain Tumor. A novel classification framework is derived by introducing the local independent projection into the classical classification model. Locality is important in the calculation of local independent projections for LIPC The artificial neural network is used to classify the stages of Brain Tumor then it is trained network. Morphologic contents of MRI frequently require segmentation of the image volume into tissue types. Manual segmentation also shows large intra- and inter-observer variability For example, accurate segmentation of MR images of the brain is of interest in the study of many brain disorders. The proposed method consists of four major steps, i.e., preprocessing, feature extraction, tumor segmentation using the LIPC method, and postprocessing. To reduce computational costs, we embedded the proposed method in a multiresolution framework. The qualitative results of the proposed method with the learned softmax regression model on different data groups. Parameters k, N, and w were set to 10, 40 000, and 5, respectively, for each class at each level. The tumor boundaries of the real patient data were more blurry than those of the synthetic data .Therefore, the tumor classification performance was better in the synthetic data than that in the real patient data. The edema boundaries of both real patient data and synthetic data were quite blurry, which led to more inaccurately classified voxels in the edema regions than those in the tumor regions. Two other typical segmentation results using the proposed method. That the tumor boundary of low-grade real patient data was more blurry than that of high-grade real patient data. The edema region was also blurry in low-grade real patient data. Thus, the classification accuracy of the tumor and edema in the low-grade patient data was lower than those in the high-grade patient data. To evaluate the Effectiveness of LIPC, both SRC and LIPC used LAE as the coding method. Moreover, the classification scores were computed. For LIPC, parameters k, N, and w were set to 10, 40 000, and 5, respectively, for each class at each level. For SRC, a dictionary containing samples from three classes was constructed for each level. This dictionary consisted of three subdictionaries and each subdictionary corresponded to a class. For a fair comparison, the size of each sub dictionary was set to 40 000. Therefore, the size of the dictionary for SRC was 120 000 at each level. The number of nonzero values in LAE for SRC was determined as follows. We first randomly selected 10 000 samples from the training data for each class at each level and computed the reconstruction error norms of all the selected samples using the dictionary. The number of nonzero values in LAE was varied from 5 to 1,200. Finally, the minimum reconstruction error was found when the number of nonzero values was set to 1000. After we investigated the results of different data groups, the mean DS of LIPC was 5.3% higher than that of SRC. The classification results with LIPC and SRC on different data groups are displayed, which shows that the proposed LIPC could be effectively used in tumor segmentation.

**ADVANTAGE:**

* The main advantage of this process is the detecting the tumor region, hence it is more helpful for the easy diagnosis.
* The accuracy of the segmentation is improved due to the thresholding.
* This method is superior when compared with SVM classifier.

**SYSTEM ARCHITECTURE**

**Input Image**

**Preprocessing**

**Classification**

**Feature Extraction**

**Segmentation**

**FLOW DIAGRAM**

**Dataset**

**Input Image**

**Database**

**Train Feature**

**Label**

**Test Feature**

**LIPC**

**Thresholding**

**Noise filtering**

**Classification**

**Feature Extraction**

**Segmentation**

**Tumor Detection**

**Preprocessing**

**UML DIAGRAM**

**ACTIVITY DIAGRAM**

****

**USE-CASE DIAGRAM**

****

**SEQUENCE DIAGRAM**

****

**TESTING OF PRODUCT**

**SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be

Accepted.

Invalid Input : identified classes of invalid input must be

Rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirement. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**MODULES:**

* Load Image
* Preprocessing
* Segmentation
* Feature Extraction
* Classification

**MODULE DESCRIPTION:**

**LOAD IMAGE:**

In this module we initially select the input image from the data set. We select the different types of MRI brain abnormal images from the data base. The database is nothing but the collection of the image which is undertaken for the process.

**PREPROCSSING:**

The test image is preprocessed by using the median filter. The median filter is a nonlinear digital filtering technique, often used to remove noise from the image. 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). The median filter is a nonlinear digital filtering technique, often used to remove noise. 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. Median filtering smoothes the image and is thus useful in reducing noise. Unlike low pass filtering, median filtering can preserve discontinuities in a step function and can smooth a few pixels whose values differ significantly from their surroundings without affecting the other pixels. It is also useful in preserving edges in an image while reducing random noise. Impulsive or salt-and pepper noise can occur due to a random bit error in a communication channel.

**SEGMENTATION:**

Segmentation is the process of segmenting the required region from the whole image. On the other hand, segmentation is the process of segmenting the Region of Interest (ROI). After the segmentation, the background other than the required region is eliminated. Only the tumor region is segmented. Although multimodal MRI images can provide complementary information in the tumor area, brain tumor segmentation is still a challenging and difficult task. Brain tumors can have various sizes and shapes and may appear at different locations. In addition to tumor heterogeneity, tumor edges can be complex and visually vague. Moreover, some tumors may deform surrounding structures in the brain because of the mass effect or edema. Additionally, artifacts and noise in brain tumor images increase the difficulty when segmenting tumors. Thus, designing of a semiautomatic or automatic brain tumor segmentation approach is necessary to provide an acceptable performance. However, the tumor segmentation performance of these methods highly depends on evident difference in the intensities between tumor and nontumor regions. Watershed and region growing approaches are simple and consistently produce complete boundaries. However, these two methods are sensitive to noise, which is a common problem in the intensity-based method. Moreover, most intensity-based methods tend to oversegment tumors because of the weak and diffused edges caused by edema.

**FEATURE EXTRACTION:**

Feature extraction is the process of collecting the data from the image. The data will be the numerical value. The collected data is saved as the “.mat” file format. Divide the examined window into cells (e.g. 16x16 pixels for each cell).For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise. Where the center pixel's value is greater than the neighbor's value, write "1". Otherwise, write "0". This gives an 8-digit binary number (which is usually converted to decimal for convenience).Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center).Optionally normalize the histogram. Concatenate (normalized) histograms of all cells. This gives the feature vector for the window .Initially we separate the image as patches. For each patch of image we apply the LBP (Local Binary Pattern).

**CLASSIFICATION:**

LIPC classification method is used to classify the input image is normal or abnormal. To achieve this we train the LIPC by using the three parameters namely LABEL, Training features and Test features. By learning, the LIPC classifies the age of the test image. Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T (that is, or a white pixel if the image intensity is greater than that constant. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming complete white. LIPC: LIPC is the algorithm used for the classification of the image, which is mean by the classification of the features extracted from the image. The test feature is compared with the already trained feature. A novel classification framework is derived by introducing the local  
independent projection into the classical classification model. Locality is important in the calculation of local independent projections  
for LIPC the artificial neural network is used to classify the stages of Brain Tumor.

**LITERATURE SURVEY**

**1) Title: Bayesian image segmentation using local iso-intensity structural orientation (2005)**

**Author: W. Wong, A. Chung**

Image segmentation is a fundamental problem in early computer vision. In segmentation of flat shaded, nontextured objects in real-world images, objects are usually assumed to be piecewise homogeneous. This assumption, however, is not always valid with images such as medical images. As a result, any techniques based on this assumption may produce less-than-satisfactory image segmentation. In this work, we relax the piecewise homogeneous assumption. By assuming that the intensity nonuniformity is smooth in the imaged objects, a novel algorithm that exploits the coherence in the intensity profile to segment objects is proposed. The algorithm uses a novel smoothness prior to improve the quality of image segmentation. The formulation of the prior is based on the coherence of the local structural orientation in the image. The segmentation process is performed in a Bayesian framework. Local structural orientation estimation is obtained with an orientation tensor. Comparisons between the conventional Hessian matrix and the orientation tensor have been conducted. The experimental results on the synthetic images and the real-world images have indicated that our novel segmentation algorithm produces better segmentations than both the global thresholding with the maximum likelihood estimation and the algorithm with the multilevel logistic MRF model.

**ADVANTAGE:**

* The advantage is that the normal vessel models (i.e., augmented vessels) are easier to manipulate as compared with the model of the complex shaped disease lumens.

**DISADVANTAGE:**

* Therefore, the local intensity statistics in the vessel and background regions may not be reliable, or the intensity gradient magnitude may not be large enough on the vessel boundary for the conventional image segmentation methods.

**2. Title: Improved Intensity In homogeneity Correction Techniques in MR Brain image Segmentation (2002)**

**Author: C.M. Li, C. Xu, A. Anderson, J. Gore.**

Intensity inhomogeneity or intensity non-uniformity (INU) is an undesired phenomenon that represents the main obstacle for MR image segmentation and registration methods. Various techniques have been proposed to eliminate or compensate the INU, most of which are embedded into clustering algorithms. This paper proposes a pre-filtering technique for Gaussian and impulse noise elimination, and a smoothening filter that assists the fuzzy c-means (FCM) algorithm at the estimation of inhomogeneity as a slowly varying additive or multiplicative noise. The segmentation is produced by FCM algorithm together with the INU estimation. The slowly varying behavior of the bias or gain field is assured by a smoothening filter that performs a context dependent averaging, based on a morphological criterion. The experiments using 2-D synthetic phantoms and real MR images show, that the proposed method provides accurate segmentation. The produced segmentation and fuzzy membership values can serve as excellent support for 3-D registration and segmentation techniques.

**ADVANTAGE:**

* The proposed method proved to segment accurately and efficiently MR images in the presence of intensity non-uniformity.

**DISADVANTAGE:**

* This approach reduces the amount of necessary computations, but the result of the segmentation is not deterministic due to the nature of the smoothing filter.

**3. Title: Segmentation of MR images with intensity inhomogeneities. (2004)**

**Author: Jagath C. Rajapakse, Frithjof Kruggel.**

Segmentation of intensity inhomogeneous regions is a well-known problem in image analysis applications. This paper presents a region-based active contour method for image segmentation, which properly works in the context of intensity inhomogeneity problem. The proposed region-based active contour method embeds both region and gradient information unlike traditional methods. It contains mainly two terms, area and length, in which the area term practices a new region-based signed pressure force (SPF) function, which utilizes mean values from a certain neighborhood using the local binary fitted (LBF) energy model. In turn, the length term uses gradient information. The novelty of our method is to locally compute new SPF function, which uses local mean values and is able to detect boundaries of the homogenous regions. Finally, a truncated Gaussian kernel is used to regularize the level set function, which not only regularizes it but also removes the need of computationally expensive reinitialization. The proposed method targets the segmentation problem of intensity inhomogeneous images and reduces the time complexity among locally computed active contour methods. The experimental results show that the proposed method yields better segmentation result as well as less time complexity compared with the state-of-the-art active contour methods.

**ADVANTAGE:**

* The proposed segmentation algorithm is applied to synthetic and brain MR images in order to demonstrate the accuracy, effectiveness, and robustness of the algorithm.

**DISADVANTAGE:**

* The region-based model is not sensitive to initialization of the level set function and can recognize the object’s boundaries efficiently.

**4. Title: A Modified Fuzzy C-Means Algorithm for Segmentation of Magnetic Resonance Images (2002)**

**Author: M.N. Ahmed, S. Yamany, N. Mohamed, A.A. Farag, T. Moriarty.**

This paper presented a new approach for robust segmentation of Magnetic Resonance images that have been corrupted by intensity inhomogeneities and noise. The algorithm is formulated by modifying the objective function of the standard fuzzy C-means (FCM) method to compensate for intensity inhomogeneities. A additional term is injected into the objective function to constrain the behavior of membership functions with the neighborhood effect. And an adaptive K-means clustering algorithm that initializes the centroids is described. The efficacy of the

algorithm is demonstrated on both simulated and real Magnetic Resonance images.

**ADVANTAGE:**

* It can provide significant advantage for fast computation.
* Therefore, it is important to take advantage of useful date while at the same time overcoming potential difficulties.

**DISADVANTAGE:**

* Its main disadvantages are that the performance degrades significantly with increased noise and its computational complexity.

**5. Title: Automatic anatomical brain MRI segmentation combining label  
propagation and decision fusion (2009)**

**Author: Duchesne, S., Pruessner, J., Collins**

Regions in three-dimensional magnetic resonance (MR) brain images can be classified using protocols for manually segmenting and labeling structures. For large cohorts, time and expertise requirements make this approach impractical. To achieve automation, an individual segmentation can be propagated to another individual using an anatomical correspondence estimate relating the atlas image to the target image. The accuracy of the resulting target labeling has been limited but can potentially be improved by combining multiple segmentations using decision fusion. We studied segmentation propagation and decision fusion on 30 normal brain MR images, which had been manually segmented into 67 structures. Correspondence estimates were established by nonrigid registration using free-form deformations. Both direct label propagation and an indirect approach were tested. Individual propagations showed an average similarity index (SI) of 0.754 ± 0.016 against manual segmentations. Decision fusion using 29 input segmentations increased SI to 0.836 ± 0.009. For indirect propagation of a single source via 27 intermediate images, SI was 0.779 ± 0.013. We also studied the effect of the decision fusion procedure using a numerical simulation with synthetic input data. The results helped to formulate a model that predicts the quality improvement of fused brain segmentations based on the number of individual propagated segmentations combined. We demonstrate a practicable procedure that exceeds the accuracy of previous automatic methods and can compete with manual delineations.

**ADVANTAGE:**

* Accurate and reliable methods for segmentation (classifying image regions) are a key requirement for the extraction of information from images.

**DISADVANTAGE:**

* Anatomical segmentation is comparatively difficult to automate because structures that are anatomically distinct do not necessarily differ in their signal properties and can be composed of more than one tissue type.

**SYSTEM REQUIREMENTS:**

**Hardware Requirement:**

* Pentium IV – 2.7 GHz
* 1GB DDR RAM
* 250Gb Hard Disk

**Software Requirement:**

* Operating System : Windows XP, 7
* Tool : Matlab
* Version : 7.9

**SOFTWARE DESCRIPTION**:

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

Matlab is a data analysis and visualization tool which has been designed with powerful support for matrices and matrix operations. As well as this, Matlab has excellent graphics capabilities, and its own powerful programming language. One of the reasons that Matlab has become such an important tool is through the use of sets of Matlab programs designed to support a particular task. These sets of programs are called toolboxes, and the particular toolbox of interest to us is the image processing toolbox. Rather than give a description of all of Matlab’s capabilities, we shall restrict ourselves to just those aspects concerned with handling of images. We shall introduce functions, commands and techniques as required. A Matlab function is a keyword which accepts various parameters, and produces some sort of output: for example a matrix, a string, a graph. Examples of such functions are sin, imread, imclose. There are many functions in Matlab, and as we shall see, it is very easy (and sometimes necessary) to write our own.

Matlab's standard data type is the matrix all data are considered to be matrices of some sort. Images, of course, are matrices whose elements are the grey values (or possibly the RGB values) of its pixels. Single values are considered by Matlab to be matrices, while a string is merely a matrix of characters; being the string's length. In this chapter we will look at the more generic Matlab commands, and discuss images in further chapters.

When you start up Matlab, you have a blank window called the Command Window\_ in which you enter commands. Given the vast number of Matlab's functions, and the different parameters they can take, a command line style interface is in fact much more efficient than a complex sequence of pull-down menus.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement financial modeling and analysis. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

When working with images in Matlab, there are many things to keep in mind such as loading an image, using the right format, saving the data as different data types, how to display an image, conversion between different image formats.

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, spatial transformations, and image registration. Many functions in the toolbox are multithreaded to take advantage of multi core and multiprocessor computers.

MATLAB and images

• The help in MATLAB is very good, use it!

• An image in MATLAB is treated as a matrix

• Every pixel is a matrix element

• All the operators in MATLAB defined on

Matrices can be used on images: +, -, \*, /, ^, sqrt, sin, cos etc.

• MATLAB can import/export several image formats

– BMP (Microsoft Windows Bitmap)

– GIF (Graphics Interchange Files)

– HDF (Hierarchical Data Format)

– JPEG (Joint Photographic Experts Group)

– PCX (Paintbrush)

– PNG (Portable Network Graphics)

– TIFF (Tagged Image File Format)

– XWD (X Window Dump)

– MATLAB can also load raw-data or other types of image data

–

• Data types in MATLAB

– Double (64-bit double-precision floating point)

– Single (32-bit single-precision floating point)

– Int32 (32-bit signed integer)

– Int16 (16-bit signed integer)

– Int8 (8-bit signed integer)

– Uint32 (32-bit unsigned integer)

– Uint16 (16-bit unsigned integer)

– Uint8 (8-bit unsigned integer)

Images in MATLAB

Binary images: {0, 1}

• Intensity images: [0, 1] or uint8, double etc.

• RGB images: m-by-n-by-3

• Indexed images: m-by-3 color map

• Multidimensional images m-by-n-by-p (p is the number of layers)

**IMAGE TYPES IN MATLAB**

Outside Matlab images may be of three types i.e. black & white, grey scale and colored. In Matlab, however, there are four types of images. Black & White images are called binary images, containing 1 for white and 0 for black. Grey scale images are called intensity images, containing numbers in the range of 0 to 255 or 0 to 1. Colored images may be represented as RGB Image or Indexed Image.

In RGB Images there exist three indexed images. First image contains all the red portion of the image, second green and third contains the blue portion. So for a 640 x 480 sized image the matrix will be 640 x 480 x 3. An alternate method of colored image representation is Indexed Image. It actually exist of two matrices namely image matrix and map matrix. Each color in the image is given an index number and in image matrix each color is represented as an index number. Map matrix contains the database of which index number belongs to which color.

**IMAGE TYPE CONVERSION**

• RGB Image to Intensity Image (rgb2gray)

• RGB Image to Indexed Image (rgb2ind)

• RGB Image to Binary Image (im2bw)

• Indexed Image to RGB Image (ind2rgb)

• Indexed Image to Intensity Image (ind2gray)

• Indexed Image to Binary Image (im2bw)

• Intensity Image to Indexed Image (gray2ind)

• Intensity Image to Binary Image (im2bw)

• Intensity Image to RGB Image (gray2ind, ind2rgb)

**Key Features**

• High-level language for technical computing

• Development environment for managing code, files, and data

• Interactive tools for iterative exploration, design, and problem solving

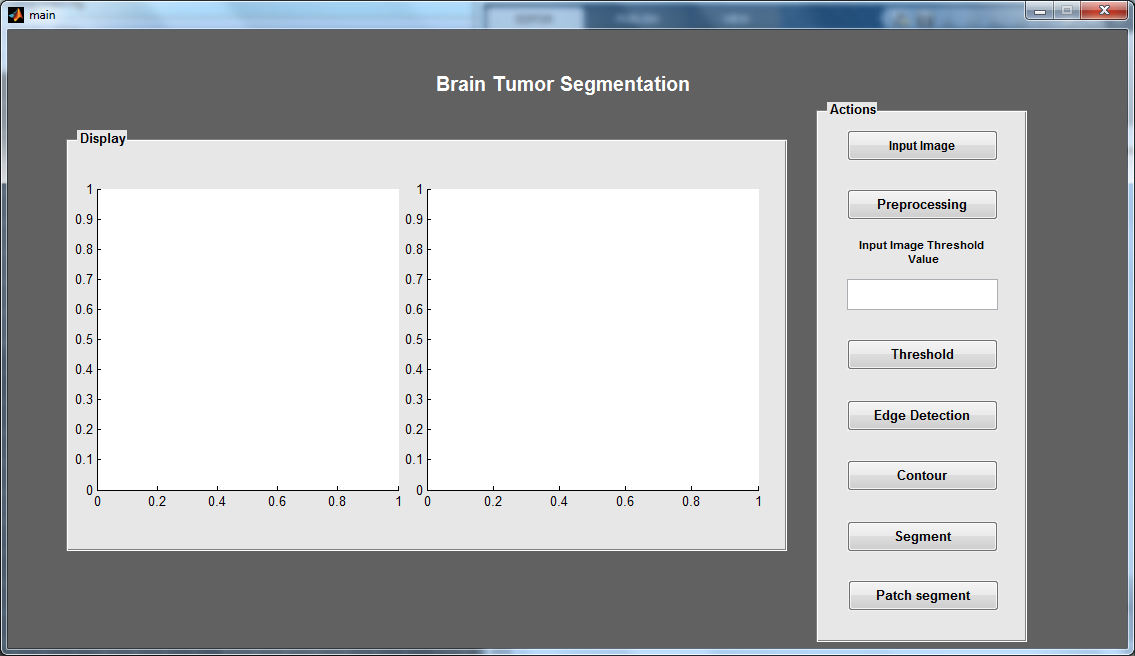
• Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration

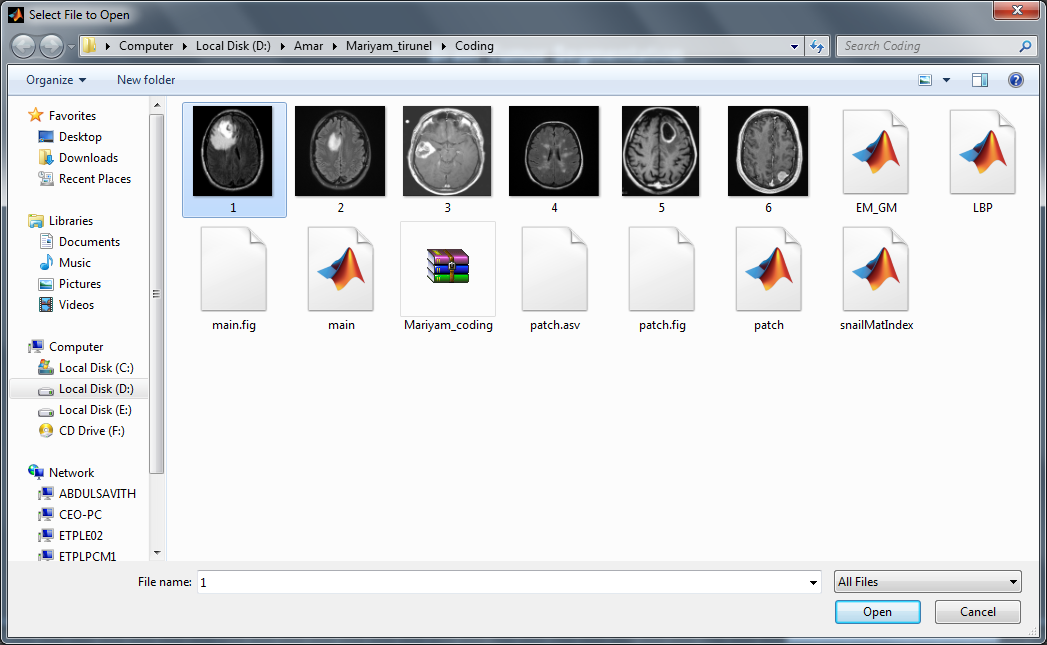
• 2-D and 3-D graphics functions for visualizing data

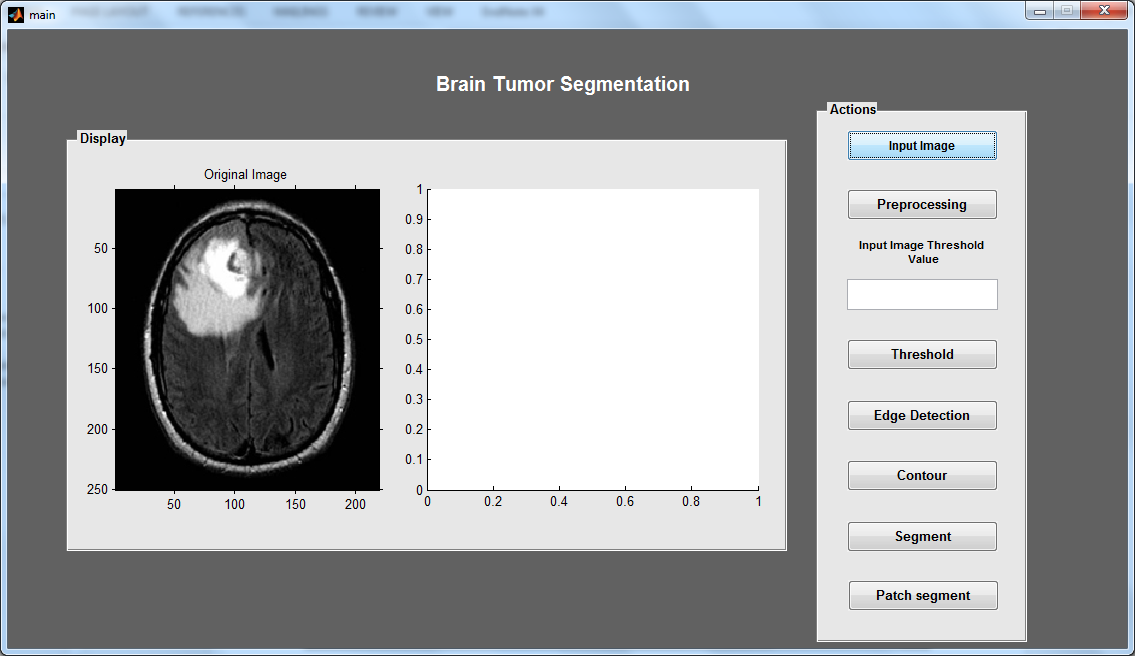
• Tools for building custom graphical user interfaces

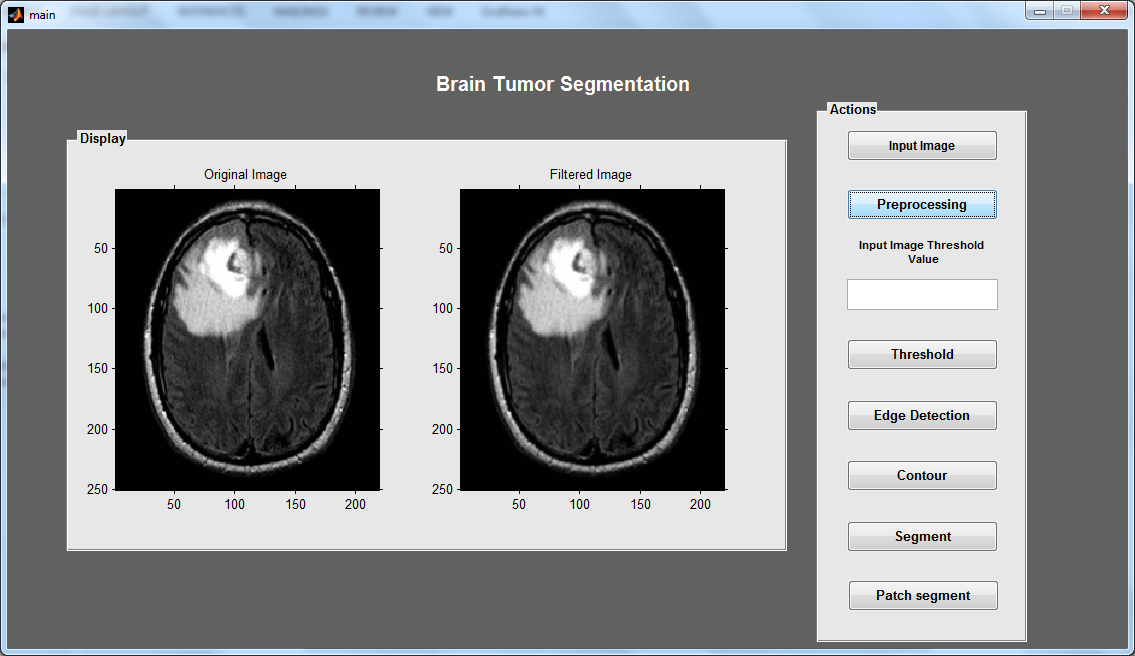
• Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, FORTRAN, Java, COM, and Microsoft Excel.

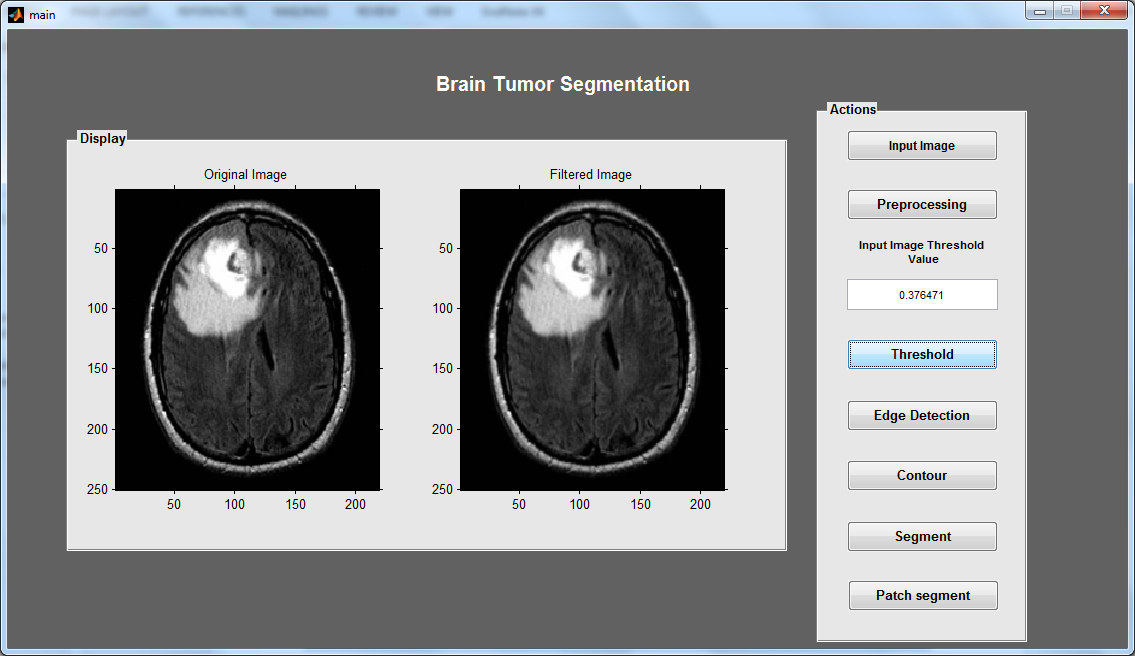
**SCREENSHOT**

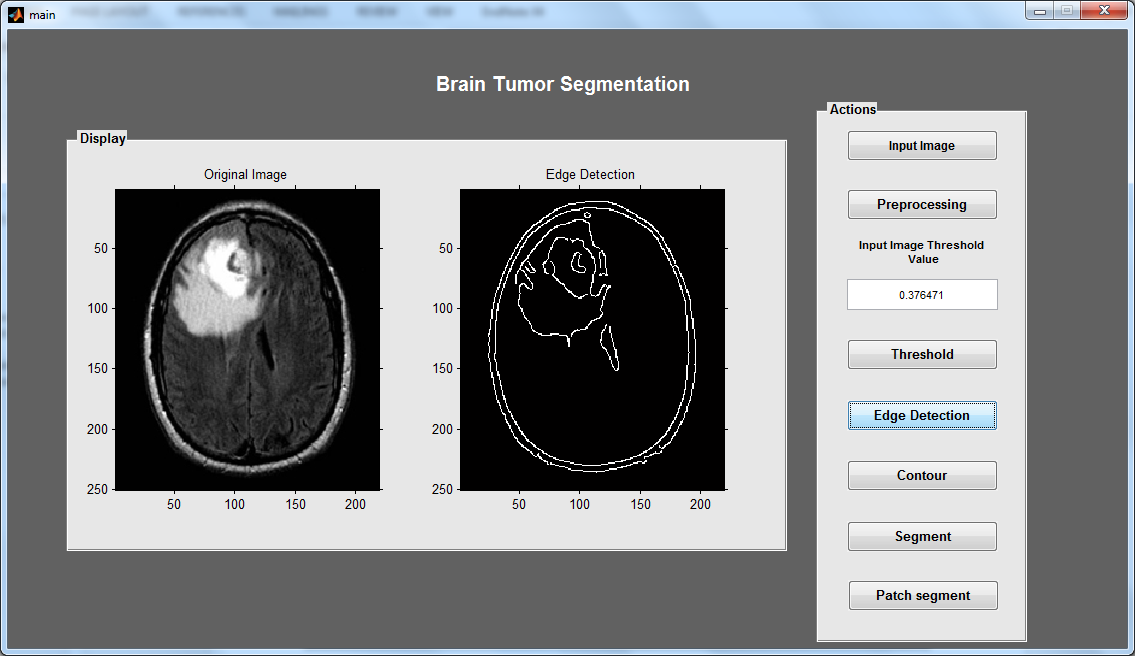
****

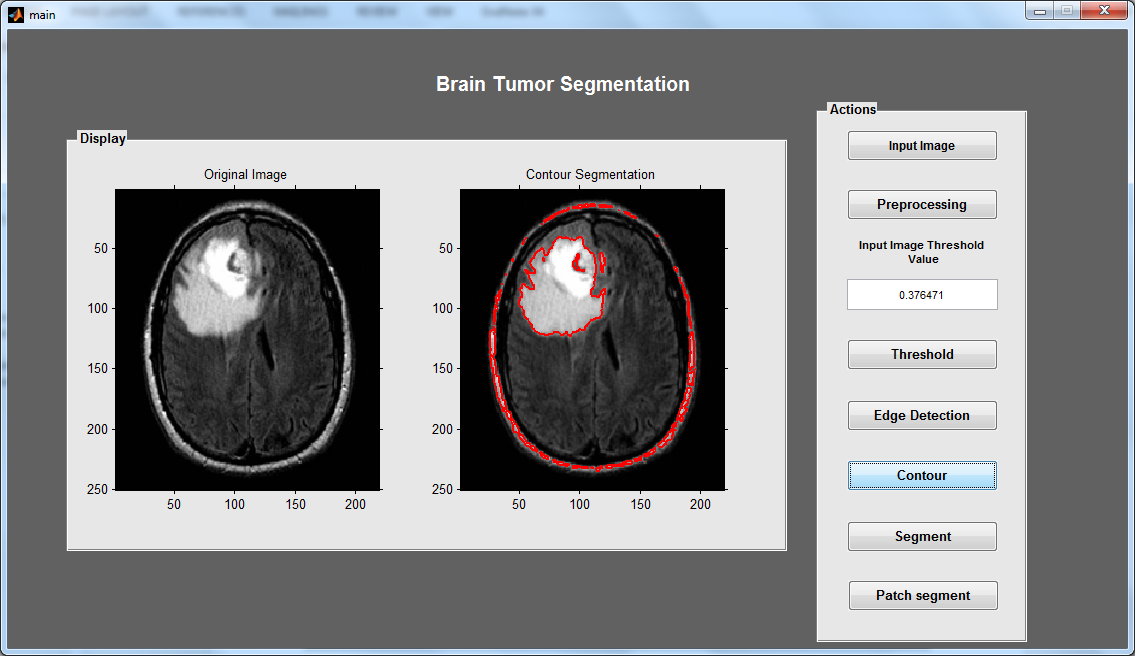
****

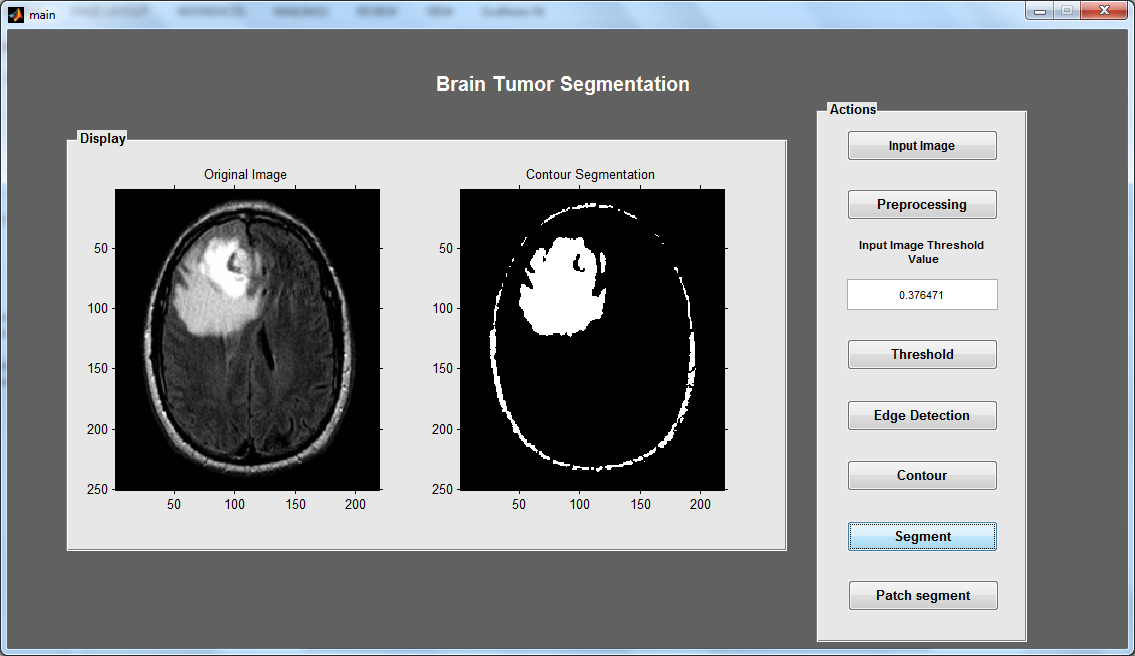
****

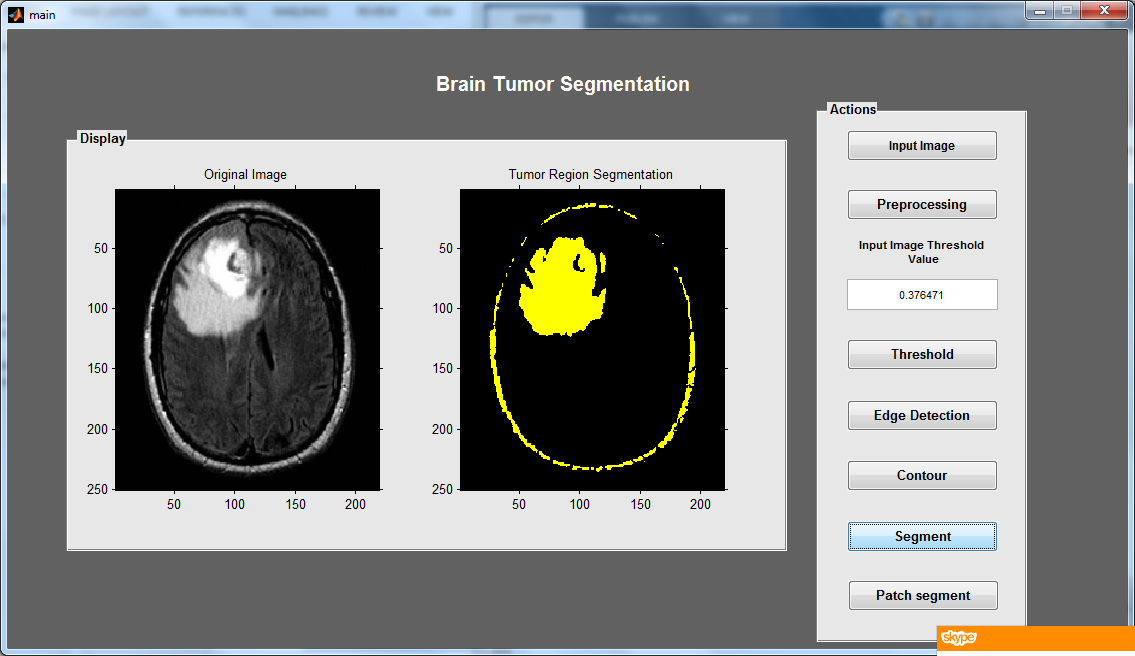
****

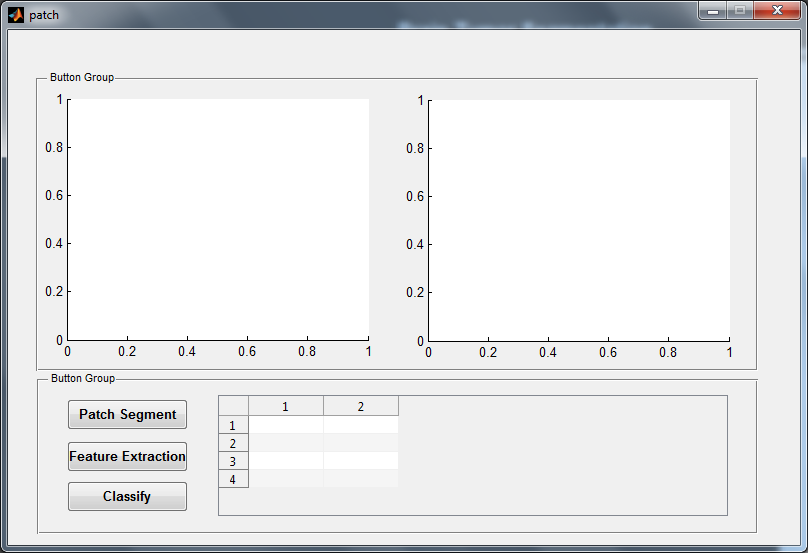
****

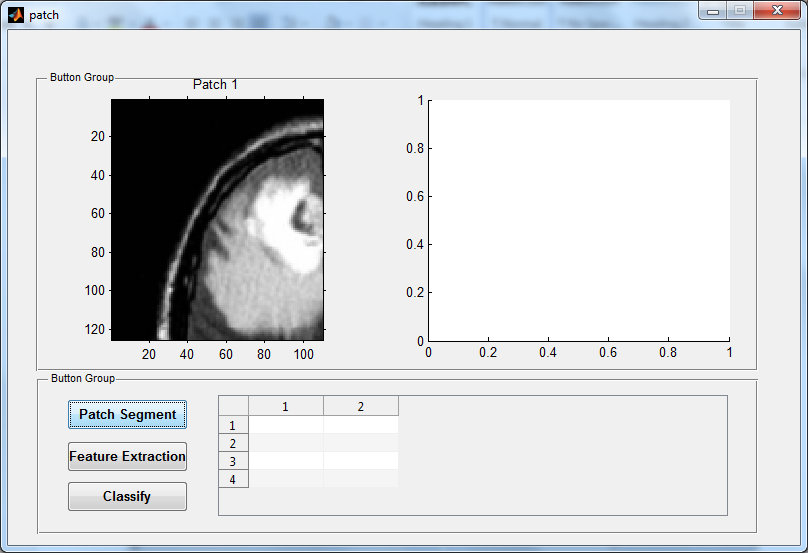
****

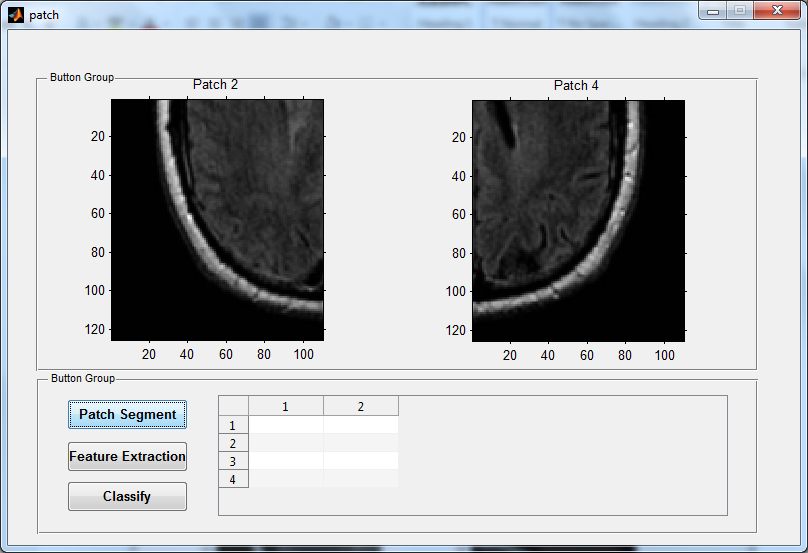
****

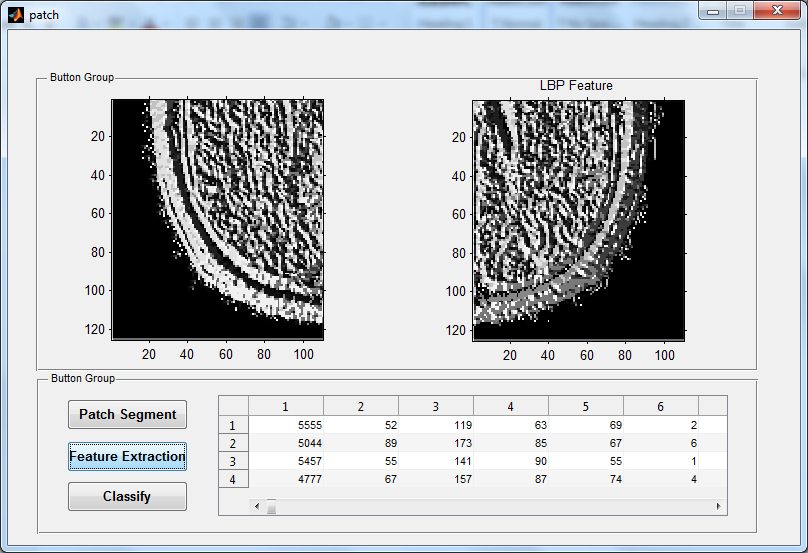
****

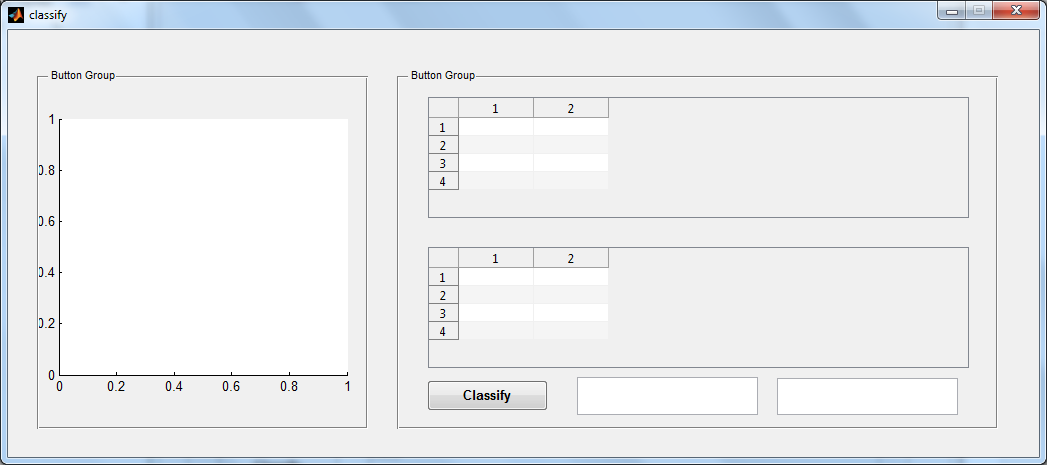
****

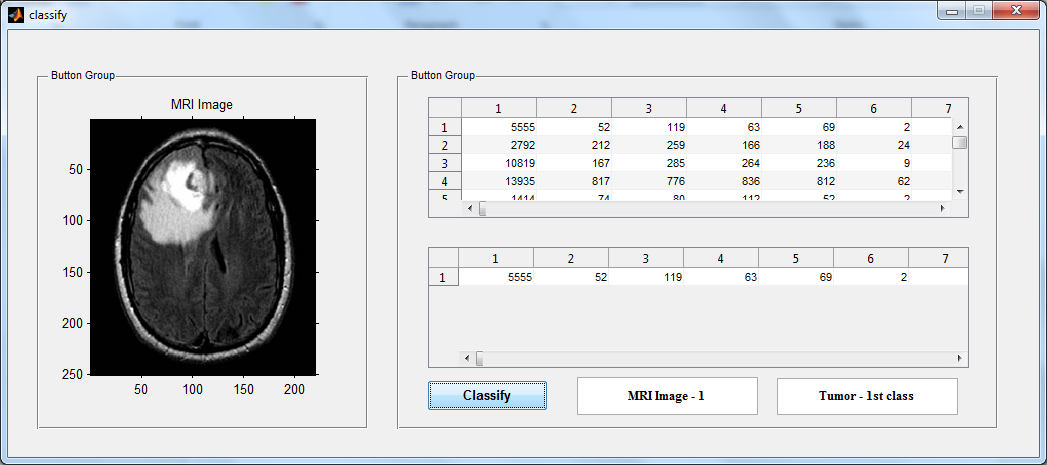












**CONCLUSION**

Hence, we introduce new method to Segment the MRI Brain tumors and Classify the image is normal or abnormal. Simulation results shows that our Classifier and segmentation outperforms than other Techniques. An automatic method is proposed for brain tumor segmentation in MRI images. An LIPC-based method was introduced to solve the tumor segmentation problem. The proposed LIPC used local independent projection into the classical classification model, and a novel classification framework was derived. Compared with other coding approaches, the LAE method was more suitable in solving the linear reconstruction weights under the locality constraint. The data distribution in each sub manifold was important for the classification, and we used a softmax model to learn the relationship between the data distribution and reconstruction error norm. We evaluated the proposed method using both synthetic data and public available brain tumor image data. In both problems, our method outperformed competing methods.

**REFERENCES**

[1] 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, M. Wintermark, M. Jilwan, P. Raghavan, E. Huang, R. J. Clifford, P. Mongkolwat, V. Kleper, J. Freymann, J. Kirby, P. O. Zinn, C. S. Moreno, C. Jaffe, R. Colen, D. L. Rubin, J. Saltz, A. Flanders, and D. J. Brat, “MR imaging predictors of molecular profile and survival: multi-institutional study of the TCGA glioblastoma data set,” Radiology, vol. 267, no. 2, pp. 560–569, May 2013.

[2] D. Bhattacharyya and T. H. Kim, “Brain tumor detection using MRI image analysis,” Commun. Comput. Inform. Sci., vol. 151, pp. 307–314, 2011.

[3] C. L. Biji, D. Selvathi, and A. Panicker, “Tumor detection in brain magnetic resonance images using modified thresholding techniques,” Commun. Comput. Inform. Sic., vol. 4, pp. 300–308, 2011.

[4] P. Gibbs, D. L. Buckley, S. J. Blackband, and A. Horsman, “Tumour volume determination from MR images by morphological segmentation,” Phys. Med. Biol., vol. 41, no. 11, pp. 2437–2446, Nov. 1996.

[5] R. Ratan, S. Sharma, and S. K. Sharma, “Brain tumor detection based on multi-parameter MRI image analysis,” Int. J. Graphics, Vision Image Process., vol. 9, no. 3, pp. 9–11, 2009.

[6] T. M. Hsieh, Y. M. Liu, C. C. Liao, F. Xiao, I. J. Chiang, and J. M. Wong, “Automatic segmentation of meningioma from non-contrasted brain MRI integrating fuzzy clustering and region growing,” BMC Med. Informat. Decision Making, vol. 11, p. 54, 2011.

[7] B. N. Saha, N. Ray, R. Greiner, A. Murtha, and H. Zhang, “Quick detection of brain tumors and edemas: A bounding box method using symmetry,” Comput. Med. Imag. Graphics, vol. 36, no. 2, pp. 95–107, 2012.

[8] Z. Iscan, Z. Dokur, and T. Olmez, “Tumor detection by using Zernike moments on segmented magnetic resonance brain images,” Expert Syst. Appl., vol. 37, no. 3, pp. 2540–2549, 2010.

[9] H. Khotanlou, O. Colliot, J. Atif, and I. Bloch, “3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models,” Fuzzy Sets Syst., vol. 160, no. 10, pp. 1457–1473, 2009.

[10] Z. J. Wang, Q. M. Hu, K. F. Loe, A. Aziz, and W. L. Nowinski, “Rapid and automatic detection of brain tumors in MR images,” in Proc. SPIE, 2004.

[11] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig, “A brain tumor segmentation framework based on outlier detection,” Med. Image Anal., vol. 8, no. 3, pp. 275–283, Sep. 2004.

[12] 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 model of lesion growth,” IEEE Trans. Med. Imag., vol. 23, no. 10, pp. 1301–1314, Oct. 2004.

[13] N. Moon, E. Bullitt, K. V. Leemput, and G. Gerig, “Model-based brain and tumor segmentation,” in Proc. 16th Int. Conf. Pattern Recog., 2002, pp. 528–531.

[14] M. R. Kaus, S. K. Warfield, A. Nabavi, P. M. Black, F. A. Jolesz, and R. Kikinis, “Automated segmentation of MR images of brain tumors,” Radiology, vol. 218, no. 2, pp. 586–591, Feb. 2001.

[15] J. Sachdeva, V. Kumar, I. Gupta, N. Khandelwal, and C. K. Ahuja, “A novel content-based active contour model for brain tumor segmentation,” Magn. Resonance Imag., vol. 30, no. 5, pp. 694–715, Jun. 2012.

[16] S. Taheri, S. H. Ong, and V. F. H. Chong, “Level-set segmentation of brain tumors using a threshold-based speed function,” Image Vision Comput., vol. 28, no. 1, pp. 26–37, 2010.

[17] T. Wang, I. Cheng, and A. Basu, “Fluid vector flow and applications in brain tumor segmentation,” IEEE Trans. Biomed. Eng., vol. 56, no. 3, pp. 781–789, Mar. 2009.

[18] A. Hamamci, N. Kucuk, K. Karaman, K. Engin, and G. Unal, “TumorCut: segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications,” IEEE Trans. Med. Imag., vol. 31, no. 3, pp. 790–804, Mar. 2012.

[19] J. Jiang, Y. Wu, M. Huang, W. Yang, W. F. Chen, and Q. J. Feng, “3D brain tumor segmentation in multimodal MR images based on learning population- and patient-specific feature sets,” Comput. Med. Imag. Graphics, vol. 37, no. 7–8, pp. 512–521, Jun. 28, 2013.

[20] Y. Wu, W. Yang, J. Jiang, S. Q. Li, Q. J. Feng, and W. F. Chen, “Semi-automatic segmentation of brain tumors using population and individual information,” J. Digital Imag., vol. 26, no. 4, pp. 786–796, 2013.

[21] D. Zikic, B. Glocker, E. Konukoglu, J. Shotton, A. Criminisi, D. H. Ye, C. Demiralp, O. M. Thomas, T. Das, R. Jena, and S. J. Price, “Contextsensitive classification forests for segmentation of brain tumor tissues,” presented at the Med. Image Comput. Comput.-Assisted Intervention Conf.–Brain Tumor Segmentation Challenge, Nice, France, 2012.

[22] D. Zikic, B. Glocker, E. Konukoglu, A. Criminisi, C. Demiralp, J. Shotton, O. M. Thomas, T. Das, R. Jena, and S. J. Price, “Decision forests for tissuespecific segmentation of high-grade gliomas in multi-channel MR,” in Proc. Med. Image Comput. Comput. Assist. Interv. Conf., 2012, pp. 369– 376.

[23] S. Bauer, T. Fejes, J. Slotboom, R. Wiest, L.-P. Nolte, and M. Reyes, “Segmentation of brain tumor images based on integrated hierarchical classification and regularization,” presented at the Med. Image Comput. Comput.-Assisted Intervention Conf.–Brain Tumor Segmentation Challenge, Nice, France, 2012.

[24] S. Bauer, L. P. Nolte, and M. Reyes, “Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization,” in Proc. Med. Image Comput. Comput. Assist. Interv., 2011, pp. 354–361.

[25] M. Wels, G. Carneiro, A. Aplas, M. Huber, J. Hornegger, and D. Comaniciu, “A discriminative model-constrained graph cuts approach to fully automated pediatric brain tumor segmentation in 3-D MRI,” in Proc. Med. Image Comput. Comput. Assist. Interv., 2008, pp. 66– 75.

[26] C. H. Lee, S. Wang, A. Murtha, M. R. Brown, and R. Greiner, “Segmenting brain tumors using pseudo-conditional random fields,” in Proc. Med. Image Comput. Comput. Assist. Interv., 2008, pp. 359–366.

[27] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille, “Efficient multilevel brain tumor segmentation with integrated Bayesian model classification,” IEEE Trans. Med. Imag., vol. 27, no. 5, pp. 629–640, May 2008.

[28] G. P. Beyer, R. P. Velthuizen, F. R. Murtagh, and J. L. Pearlman, “Technical aspects and evaluation methodology for the application of two automated brain MRI tumor segmentation methods in radiation therapy planning,” Magn. Resonance Imag., vol. 24, no. 9, pp. 1167–1178, Nov. 2006.

[29] C. H. Lee, M. Schmidt, A. Murtha, A. Bistritz, J. Sander, and R. Greiner, “Segmenting brain tumors with conditional random fields and support vector machines,” in Proc. First Int. Conf. Comput. Vision Biomed. Image Appl., 2005, pp. 469–478.

[30] V. Harati, R. Khayati, and A. Farzan, “Fully automated tumor segmentation based on improved fuzzy connectedness algorithm in brain MR images,” Comput. Biol. Med., vol. 41, no. 7, pp. 483–492, Jul. 2011.

[31] T. Logeswari and M. Karnan, “An enhanced implementation of brain tumor detection using segmentation based on soft computing,” presented at the Int. Conf. Signal Acquisition Process., Bangalore, India, 2010.

[32] M. M. Ahmed and D. B. Mohammad, “Segmentation of brain MR images for tumor extraction by combining kmeans clustering and Perona–Malik anisotropic diffusion model,” Int. J. Image Process., vol. 2, no. 1, pp. 27–34, 2008.

[33] W. B. Dou, S. Ruan, Y. P. Chen, D. Bloyet, and J. M. Constans, “A framework of fuzzy information fusion for the segmentation of brain tumor tissues on MR images,” Image Vision Comput., vol. 25, no. 2, pp. 164–171, 2007.

[34] J. Liu, J. K. Udupa, D. Odhner, D. Hackney, and G. Moonis, “A system for brain tumor volume estimation via MR imaging and fuzzy connectedness,” Comput. Med. Imag. Graphics, vol. 29, no. 1, pp. 21–34, Jan. 2005.

[35] L. M. Fletcher-Heath, L. O. Hall, D. B. Goldgof, and F. R. Murtagh, “Automatic segmentation of non-enhancing brain tumors in magnetic resonance images,” Artif. Intell. Med., vol. 21, no. 1–3, pp. 43–63, Jan.–Mar. 2001.

[36] J. E. Iglesias, M. R. Sabuncu, and K. Van Leemput, “A unified framework for cross-modality multi-atlas segmentation of brain MRI,” Med. Image Anal., vol. 17, no. 8, pp. 1181–1191, Dec. 2013.

[37] H. Wang, J. W. Suh, S. R. Das, J. Pluta, C. Craige, and P. A. Yushkevich, “Multi-atlas segmentation with joint label fusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 3, pp. 611–623, Jun. 22, 2013.

[38] A. Gooya, G. Biros, and C. Davatzikos, “Deformable registration of glioma images using em algorithm and diffusion reaction modeling,” IEEE Trans. Med. Imag., vol. 30, no. 2, pp. 375–390, Feb. 2011.

[39] S. Zhang, Y. Zhan, and D. N. Metaxas, “Deformable segmentation via sparse representation and dictionary learning,” Med. Image Anal., vol. 16, no. 7, pp. 1385–1396, Oct. 2012.

[40] H. Lee, D. S. Lee, H. Kang, B. N. Kim, and M. K. Chung, “Sparse brain network recovery under compressed sensing,” IEEE Trans. Med. Imag., vol. 30, no. 5, pp. 1154–1165, May 2011.

[41] 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.

[42] Y. Gao, S. Liao, and D. Shen, “Prostate segmentation by sparse representation based classification,” Med. Phys., vol. 39, no. 10, pp. 6372–6387, Oct. 2012.

[43] W. Liu, J. F. He, and S.-F. Chang, “Large graph construction for scalable semi-supervised learning,” in Proc. Int. Conf. Mach. Learning, 2010, pp. 679–686.

[44] Y. Wu, G. Liu, M. Huang, J. Jiang, W. Yang, W. Chen, and Q. Feng, “Prostate segmentation based on variant scale patch and local independent projection,” IEEE Trans. Med. Imag.,, vol. 33, no. 6, pp. 1290–1303, Jun. 2014.

[45] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. Avants, N. Ayache, P. Buendia, L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. Iftekharuddin, R. Jena, N. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. Riklin-Raviv, S. Reza, M. Ryan, L. Schwartz, H. C. Shin, J. Shotton, C. Silva, N. Sousa, N. Subbanna, G. Szekely, T. Taylor, O. M. Thomas, N. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput, “The multimodal brain tumor image segmentation benchmark (BRATS),” IEEE Trans. Med. Imag., 2014, to be published.

[46] J. Macqueen, “Some methods for classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symp. Math. Statist. Prob., 1967, pp. 281–297.

[47] J. J. Wang, J. C. Yang, K. Yu, F. J. Lv, T. Huang, and Y. H. Gong, “Locality-constrained linear coding for image classification,” presented at the Comput. Vision Pattern Recog., San Francisco, CA, USA, 2010.

[48] K. Yu, T. Zhang, and Y. H. Gong, “Nonlinear learning using local coordinate coding,” presented at the Neural Inf. Process. Syst., Vancouver, BC, Canada, 2009.

[49] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Norwell, MA: Kluwer Academic Publishers, 2003.

[50] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, “A nonparametric method for automatic correction of intensity nonuniformity in MRI data,” IEEE Trans. Med. Imag., vol. 17, no. 1, pp. 87–97, Feb. 1998.

[51] C. Cortes and V. Vapnik, “Support-Vector Networks,” Mach. Learning, vol. 20, no. 3, pp. 273–297, 1995.

[52] N. Cordier, B. Menze, H. Delingette, and N. Ayache, “Patch-based Segmentation of Brain Tissues,” presented at the Med. Image Comput. Comput.-Assisted Intervention Conf.–Brain Tumor Segmentation Challenge, 2013, Nagoya, Japan.