Brain Tumor Detection from MRI Images

Summary

Currently, the Medical image processing is one of the most challenging and emerging field in the evolution of technology. Processing of MRI images is one of the part of this field. This project describes a strategy to detect & extraction of brain tumor from patient’s MRI scan images of the brain. This method includes some noise removal functions, segmentation and morphological operations which are the basic concepts of image processing. The detection and extraction of tumor from MRI scan images is done using C++ with the help of OpenCV.

Introduction

Medical imaging plays a central role in the diagnosis of brain tumors. Early imaging methods – invasive and sometimes dangerous – such as pneumoencephalography and cerebral angiography have been abandoned in favor of non-invasive, high-resolution techniques, especially magnetic resonance imaging (MRI) and computed tomography (CT) scans. Neoplasms will often show as differently colored masses (also referred to as processes) in CT or MRI results.

From these high-resolution images, we can derive detailed anatomical information to examine human brain development and discover abnormalities. Now-a-days there are several methodology for classifying MR images, which are fuzzy methods, neural networks, atlas methods, knowledge based techniques, shape methods, variation segmentation. MRI consists of T1 weighted, T2 weighted and PD (proton density) weighted images and are processed by a system which integrates fuzzy based technique with multispectral analysis.

The primary stage in image analysis is the pre-processing of the MRI scan image which includes image enhancement methods, segmentation method, and some morphological operations. There are assumptions made about the size and shape of the tumor for the morphological operations.

Method

The method used in this project includes two stages. Firstly, the pre-processing stage, where the image is enhanced and segmentation operation is performed. And secondly, the morphological operation stage. These two stages are divided into following steps.

1. Reading input grayscale MRI image
2. Applying high pass filter for noise removal
3. Applying median filter to enhance the image
4. Converting the grayscale image to binary image
5. Performing morphological operation 1 – Erosion
6. Performing morphological operation 1 – Dilation
7. Marking the tumor extracted from the image
8. Counting the number of tumors extracted

All the above stages are explained in detail as follows.

Reading the image

This application requires the image to be in a grayscale format. A grayscale or greyscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

MRI images are magnetic resonance images which can be acquired on computer when MRI machine scans a patient. We can acquire MRI images of the part of the body which is under test or desired. Generally, when we see MRI images on computer they look like black and white images (with shades of black and white) which are grayscale.

High pass filter

A high pass filter is the basis for most sharpening methods. An image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness.

A high pass filter tends to retain the high frequency information within an image while reducing the low frequency information. The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels. The kernel array usually contains a single positive value at its center, which is surrounded by negative values.

Low pass filter

A low pass filter is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.

Using a low pass filter tends to retain the low frequency information within an image while reducing the high frequency information. An example is an array of ones divided by the number of elements within the kernel, such as a 3 by 3 kernel containing ones which are divided by the number of ones in the matrix (i.e. divided by 9)