Brain tumor segmentation using Support Vector Machines

by:
Raouia Ayachi

Advisor:
Dr. Nahla Ben Amor

October 9, 2009

Abstract

One of the challenging tasks in the imaging area is image segmentation which consists on the extraction process of regions of interest from images. This is especially true in the medical field when the extraction of tumor tissues is crucial for successful diagnostics. Generally, this task is performed manually by medical experts which is not always obvious due to the high diversity in appearance of tumor tissue, among different patients and, in many cases, similarity between tumor and normal tissues. Thus, automating medical image segmentation remains a real challenge which has attracted the attention of several researchers in last years. In this work, we will focus on brain images segmentation and in particular on the extraction of tumor regions in Magnetic Resonance brain Images (MRI) which is a widely-used method of high quality medical imaging. Our idea is to consider this problem as a classification problem where the aim is to distinguish between normal and abnormal pixels on the basis of several features, namely intensities and texture ones (e.g. mean, variance, contrast, entropy, local homogeneity, etc.). More precisely, we propose to use Support Vector Machines (SVM) which is one of the most popular and well motivating classification methods. The experimental study will be carried on real and simulated datasets representing different tumor shapes, locations, sizes and image intensities.

Résumé

La segmentation des images joue un rôle crucial dans beaucoup d'applications en imagerie médicale. Elle consiste à extraire les régions d'intérêt de l'image, particulièrement les zones tumorales. Généralement, cette tâche qui est effectuée manuellement par les experts lors du diagnostic, n'est pas toujours évidente vu la grande diversité dans l'apparence des tissus tumoraux entre les différents patients et, dans de nombreux cas, la similitude entre la tumeur et les tissus normaux. Ainsi, l'automatisation de la segmentation d'images médicales demeure un défi qui a attiré l'attention de plusieurs chercheurs dans les dernières années. Dans ce travail, nous sommes intéressés à la segmentation des images de tumeurs cérébrales issues de l'IRM. Nous considérons le problème de segmentation comme un problème de classification dont le but est de distinguer entre pixel normal et pixel tumoral en se basant sur un ensemble de caractéristiques relatives aux intensités et texture (moyenne, variance, contraste, entropie, homogénéité locale, etc.). Plus précisément, nous proposons d'appliquer une des méthodes les plus populaires en classification, à savoir les séparateurs à vaste marge (SVM). L'étude expérimentale montre l'apport de cette méthode en utilisant des données réelles et simulées représentant différentes formes, localisations, tailles et intensités des zones tumorales.

Acknowledgements

The completion of this dissertation would not have been possible without the encouragement and help of many persons. It is my privilege to thank those who have supported me during this work.

I would like to express my sincere thanks to my advisor Dr. Nahla Ben Amor for introducing me to the area of research, and her availability and her real interest in this work. I thank her for providing guidance throughout my studies and for her valuable advice. Her encouragements have been of great value for me. I also thank her for reading carefully this dissertation allowing me to improve its content.

I am also grateful to Dr. Walid Barhoumi, researcher at *SIIVA (systèmes intelligents en imagerie et vision artificielle)* in *Institut Supérieur d'Informatique de Tunis* for his precious advice on everything related to medical image analysis, and for his useful remarks.

I am also grateful to *Institut Supérieur de Gestion de Tunis* for providing a comfortable environment for study and research. I thank all the members of the *LARODEC* laboratory in *Institut Supérieur de Gestion de Tunis*.

I would also thank Dr. Kais Chelaifa for providing me with the MR images and the whole staff of *El Manar clinic* who assist me in my work.

In closing, I would like to thank my family and my friends who have been of great support throughout my studies.

Contents

General introduction **2**

1 Basics on image segmentation **5**

1.1 Introduction 5

1.2 Brief refresher on imaging concepts 6

1.3 Brain images and their treatment 9

1.3.1 Magnetic Resonance Imaging 9

1.3.2 Brain tumors and their diagnosis 11

1.3.3 The diversity of brain tumors 15

1.4 Image segmentation 15

1.5 Thesis contribution 20

1.6 Conclusion 21

2 Support vector machines **23**

2.1 Introduction 23

2.2 Notations 24

2.3 SVM for binary classification 24

2.3.1 Linear classifier and linearly separable problems 25

2.3.2 Linear classifier and non separable problems 32

2.3.3 Non-linear classifier and non separable problems 33

2.4 Multi-class SVM 37

2.5 Conclusion 39

3 Pre-processing and post-processing phases **41**

3.1 Introduction 41

3.2 Pre-processing phase 41

3.2.1 Image registration 42

3.2.2 Noise reduction . . . 56
3.3 Post-processing phase . . . 58
3.4 Conclusion . . . 61

4 Features extraction phase **62**
4.1 Introduction . . . 62
4.2 Intensities features . . . 62
4.3 Texture features . . . 63
4.3.1 Notations . . . 63
4.3.2 First order texture features . . . 64
4.3.3 Second order texture features . . . 67
4.4 Conclusion . . . 76

5 Experimental study **77**
5.1 Introduction . . . 77
5.2 Experimental data . . . 77
5.2.1 Real data . . . 77
5.2.2 Simulated data . . . 80
5.3 Implementation . . . 82
5.4 Segmentation results . . . 83
5.4.1 Results on real data . . . 87
5.4.2 Results on simulated data . . . 89
5.5 Conclusion . . . 92

Conclusion and perspectives **98**

Bibliography **101**

List of Figures

1.1 Grey level image and its matrix representation. 7
1.2 Window position in a digital image. 8
1.3 Histogram of a grey level image. 8
1.4 T1 and T2-weighted images respectively. 10
1.5 T1-weighted image before and after the injection of a contrast agent. 10
1.6 Fully enhancing tumor after injection of the product of contrast [12]. 13
1.7 Partially enhancing tumors after injection of the product of contrast [12]. . . . 13
1.8 Astrocytomas of low grade. Left: T1 image. Right: T2 image [12]. 14
1.9 Glioblastoma case. Left: T2 image. Middle: T1 image. Right: T1 Gadolinium [12]. 15
1.10 Classification of image segmentation methods [37]. 18
1.11 Diagram illustrating our segmentation system. 22

2.1 Illustration of a linear hyperplane. 25
2.2 Illustration of a non-linearly separable case. 26
2.3 Examples of linearly separating hyperplanes. 26
2.4 (a) A separating hyperplane with small margin. (b) A separating hyperplane with larger margin. 27
2.5 Optimal hyperplane. 28
2.6 Illustration of data points and optimal hyperplane representations. 30
2.7 Slack variables. 33
2.8 Example of higher dimensional feature space. 34
2.9 A non-linear separator transformed into a linear one. 34
2.10 Decision surfaces given in (a) by a polynomial classifier, and in (b) by an RBF [47]. 36
2.11 Decision surface given by a polynomial classifier. 37

3.1 Different transformations applied to images. 43

3.2 Illustration of 2D rotation. 44
3.3 Illustration of β-spline representation of an image [29]. 47
3.4 Illustration of the joint histogram [51]. 50
3.5 Joint grey value histograms of an MR image with itself and their entropy values [51]. 50
3.6 Coregistration by maximization of NMI measure. Top: left to right: Template image (T1 after injection of contrast agent), T2 image, T1 image. Bottom: left to right: T2 and T1 images after coregistration. 53
3.7 The Talaraich template [62]. 54
3.8 The T1-weighted MNI template [62]. 54
3.9 Illustration of linear and non linear registration. 55
3.10 Noise reduction results on MRI data. Left to Right: Image before filtering, Image after filtering. 57
3.11 Example of a diamond structuring element. 58
3.12 Illustration of the effect of dilation using a square SE. 59
3.13 Illustration of the effect of erosion using a square SE. 60
3.14 Illustration of dilation and erosion operations. 60

4.1 Examples of first-order features for a brain tumor image. Top, left to right: Original image, mean, standard deviation. Middle, left to right: Skewness, kurtosis, entropy. Bottom left: Energy. 67
4.2 Relationship between a pair of pixels. 68
4.3 Examples of second-order (co-occurrence or "Haralick") textures for brain tumor image. Top, left to right: Original image, angular second momentum, contrast. Middle, left to right: absolute value, entropy, cluster shade. Bottom, left to right: Cluster prominence, inertia, local homogeneity. 75

5.1 Axial views of MR real images. From top to bottom: Patient 1, Patient 2 and Patient 3. From left to right: T1, contrast enhanced T1 and T2 images. 79
5.2 Axial views of MR simulated images. From top to bottom: the SimTumor1, SimTumor2, SimTumor3 and SimTumor4 MRI datasets. From left to right: T1, contrast enhanced T1 and T2 images. 81
5.3 Diagram illustrating tools used in our segmentation system. 82
5.4 SPM interface in a coregistration task. 84
5.5 Snakes interface during the training phase. 85
5.6 MIPAV interface in a features computation task. 86

5.7 Segmentations of axial slices of Patient 1. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing. 89
5.8 Segmentations of axial slices of Patient 2. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing. 92
5.9 Segmentations of axial slices of Patient 3. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing. 93
5.10 Segmentations of coronal slices of Patient 1. From left to right: Ground truth of T2 coronal slice, segmentation result before post-processing, and segmentation result after post-processing. 94
5.11 Segmentations of sagittal slices of Patient 1. From left to right: Ground truth of T2 sagittal slice, segmentation result before post-processing, and segmentation result after post-processing. 94
5.12 Segmentations of sagittal slices of Patient 2. From left to right: Ground truth of T2 sagittal slice, segmentation result before post-processing, and segmentation result after post-processing. 95
5.13 Segmentation results of Volume 1. 95
5.14 Segmentation results of Volume 2. 96
5.15 Segmentation results of Volume 3. 96
5.16 Segmentation results of Volume 4. 97

List of Tables

1.1 Tumors classification by grade [34]. 16

2.1 Commonly used kernel functions. 35

4.1 Matrix of pixels. 65
4.2 Matrix of pixels after treating border points. 66
4.3 The mean texture matrix. 67
4.4 Horizontal co-occurrence matrix. 70
4.5 Symmetric horizontal co-occurrence matrix. 70
4.6 Normalized symmetric horizontal co-occurrence matrix. 71
4.7 Vertical co-occurrence matrix. 71
4.8 Right diagonal co-occurrence matrix. 71
4.9 Left diagonal co-occurrence matrix. 72
4.10 Textural features from different co-occurrence matrices 75

5.1 Real data characteristics. 78
5.2 Simulated data characteristics. 80
5.3 Important variations on accuracy values for real data. 87
5.4 Comparison of the ground truth and our segmentation outputs for axial images. 88
5.5 Comparison of the ground truth and our segmentation outputs for coronal and sagittal images. 90
5.6 Important variations on accuracy values for simulated data. 90
5.7 Comparison of the ground truth and our segmentation results. 91

General introduction

Brain tumors are diseases characterized by an abnormal proliferation of cells. Establishing a brain tumor diagnosis is a complex process composed of several steps [12]:

1. Observation of external signs which generate a suspicion about the presence of a brain tumor.

2. Carrying out clinical examinations to confirm or deny the presence.

3. If the presence of a mass of abnormal cells is proven and the localization of this mass allows the sampling and analysis of tissue, so *biopsy* is accomplished.

In the diagnosis context, *in vivo* brain observation provides key indicators about the nature and gravity of tumors. Within these indicators we can mention localization, shape, volume, homogeneity, etc.

A great number of clinical tools such as *X-ray scanning*, *magnetic resonance imaging (MRI)*, *tomography*, *scintigraphy* can be used to observe brain areas *in vivo*. Among all these techniques, MRI is currently one of the most powerful imaging technique increasingly used in clinical routine. However, the observation and analysis of MR brain tumor images is not a simple task since brain tumors can appear in different localizations and may have different shapes and sizes and these characteristics often lead the doctor to multiply the number of acquisitions under different protocols.

The plenty of images acquired by MRI show the inside to doctors, however, these latters seek to know more details, such as emphasizing the tumor area, quantifying its size, extracting its shape, and so on. These tasks are too time consuming and tedious if they are manually covered by doctors. In fact, synthesizing mentally the large amount of information may be inaccurate and even impossible.

Thus, computer image processing and in particular *computer-aided diagnosis (CAD)* are important in radiology since their outputs are considered as a second opinion to assist doctors during abnormalities detection, disease progress survey, etc. In such systems, a close collaboration between doctors and researchers is required in order to validate results. This represents an interdisciplinary work where both medical and computer knowledge are needed.

In this context, we propose to develop a computer aided diagnosis system for brain tumors. More precisely, we propose to develop a tumor segmentation tool in order to detect and localize tumor areas automatically from MR images. This challenge is of particular interest to doctors since the knowledge of tumor regions can help them during monitoring tumor growth and radiological treatment planning.

The segmentation problem can be handled from several angles depending on the objective. In our work, we consider it as a binary classification problem where the aim is to discriminate between normal and tumor pixels depending on a set of images-based-features. Our idea is to apply a supervised method to segment tumor regions from MR images in order to take into account doctors knowledge and experience. To this end we have selected the **Support Vector Machines** as a classifier. Our choice is motivated by its robustness in generalization properties and its capacity to handle voluminous data. Thus our idea is to use a labeled training set relative to several patients, then to retain support vectors in order to classify new pixels relative to MR images.

Our segmentation system will be based on 5 phases, namely:

- *Pre-processing phase* which consists on registering images having different MR modalities and reducing image noise.
- *Features extraction phase* which aims to compute intensities and texture features for each normal and tumor pixel in the image.
- *Training and testing phase* which consists on training SVM using a labeled training set composed of tumor and normal pixels. Once the model that discrimnates between normal and tumor pixels is generated, new images are tested.
- *Segmentation phase* which groups tumor pixels, already classified by SVM, in order to obtain the tumor region.
- *Post-processing phase* which cleans segmentation results by merging connected regions

and removing isolated ones.

The remainder of this dissertation is organized as follows: Chapter 1 briefly presents several imaging concepts, medical images and a description of brain tumors and their diagnosis. Chapter 2 describes the segmentation method used in our work, namely SVM. Chapter 3 details pre-processing and post-processing phases, applied to MR images before processing and after segmentation. Chapter 4 is dedicated to the features extraction phase. Finally Chapter 5 presents experimental results evaluating our segmentation method.

Chapter 1

Basics on image segmentation

1.1 Introduction

Millions of medical images are produced each year to establish a diagnosis or to monitor a therapeutic action. These images provide information of the shape and the functioning of human body organs. Unfortunately, this information is extremely difficult to exploit using quantitatively and objectively process. Indeed, although the images are originally numerical, their exams are typically realized by observing a film or a succession of two-dimensional slices and the result is generally qualitative and subjective. So, the creation of tools dedicated for medical images analysis is important. These tools can group several problems into the following classes [5]:

- *Restoration* consists on improving the image, which contains some defects related to the physical process of acquisition,
- *Segmentation* extracts primitives (points, regions) that will be used in another complementary tasks like movement analysis, visualization, etc.,
- *Registration* is necessary to compare images acquired at different moments or images having different modalities,
- *Morphology* consists on studying the geometry of shapes, especially the calculation of average shapes and variations of these forms,
- *Movement analysis* of dynamic sequences images is a difficult problem because data are 4-dimensional (three spatial dimensions and a temporal dimension),
- *Visualization* has been within the most active tools in processing of medical images,

- *Simulation* defines geometric models in order to simulate organs deformations in real time,
- *Medical robotics* is a very active research line. It consists on moving from simulation to a concrete realization of a robot for a surgery or radiotherapy intervention.

In our work we are interested in segmentation problem, more precisely brain tumor segmentation which consists on extracting tumor regions from images, to use them subsequently in registration or visualization applications.

The remainder of this Chapter is organized as follows: as a first step, after stating some terminology related to the imaging area (Section 1.2), we will present brain images and their treatment. In particular, we will expose brain tumors and their diagnosis and the diversity of brain tumors (Section 1.3). Then, we will present the context of our task, by giving a definition of the segmentation problem (Section 1.4). Finally, Section 1.5 focuses on our segmentation process by describing its phases.

1.2 Brief refresher on imaging concepts

This Section gives some basic definitions relative to imaging concepts:

- A **2D-image** is a 2-dimensional light intensity function, $f(x, y)$, where x and y are spatial coordinates and the value of f at (x, y) is proportional to the brightness of the image at that point [49].
- A **digital image** is an image $f(x, y)$ that has been discretized in brightness. It is represented by a 2-dimensional integer array. The digitized brightness value is called *grey level* value. Each element of the array is called *pixel* derived from the term *"picture element"*. Usually, the size of such array is a few hundred pixels by a few hundred pixels. Thus, a digital image can be represented as follows:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & f(0,2) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & f(N-1,2) & \cdots & f(N-1,N-1) \end{pmatrix}$$

The number of bits, b, needed to store an image of size $N \times N$ with 2^m different grey levels is:

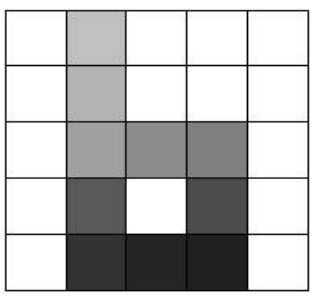
$$\text{b= N } \times N \times m \tag{1.1}$$

So, for a 512×512 image with 256 grey levels (m = 8), we need 2,097,152 bits. That is why we often try to reduce m and N, without significant loss in image quality.

- **Image dimension** corresponds to the image size. It is defined as the multiplication of the number of rows of image matrix and the number of columns.

- **Image resolution** expresses how much detail we can see in it and clearly depends on both N and m. The bigger these numbers the better the resolution.

- **Grey level or monochrome image** is an image which intensities values can take values ranging from black to white through a finite number of intermediate levels. These values can be between 0 and 255. In instance, a pixel having a grey level equal to 0 is a black pixel and a white pixel corresponds to a pixel having a grey level of 255. In this kind of images, each pixel is represented by one byte. Figure 1.1 shows a grey level image with 11 grey levels (i.e. 0, 10, 12, 14, 16, 18, 20, 40, 60, 80, 255).



255	18	255	255	255
255	16	255	255	255
255	14	12	10	255
255	80	255	60	255
255	40	20	0	255

Figure 1.1: Grey level image and its matrix representation.

- **The neighborhood of a pixel** is represented by a window containing a set of pixels surrounding a pixel. A 5×5 window means that it contains 25 pixels. Figure 1.2 defines a 5×5 window of a grey level image.

- **Grey level histogram** indicates the frequency of appearance of grey levels in a given neighborhood or in the whole image. Hence, for each image we can associate a unique histogram for the whole image and several histograms for each neighborhood. Figure 1.3 shows the histogram of the whole image where the X-axis corresponds to image grey levels and the Y-axis denotes the number of pixels for each grey level.

- **Image texture** is used to characterize the surface of an object which can be grainy, smooth, etc. Texture analysis takes into account the spatial distribution of grey levels around a pixel. Details on texture can be found in Chapter 4.

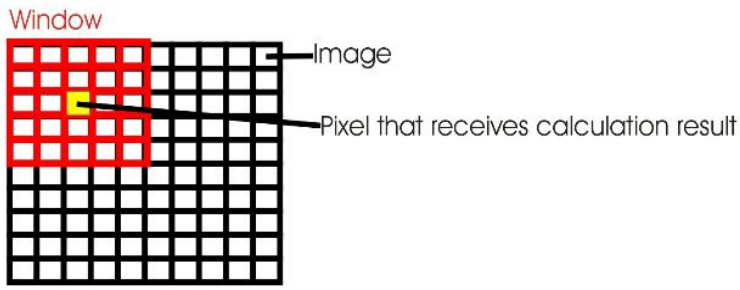


Figure 1.2: Window position in a digital image.

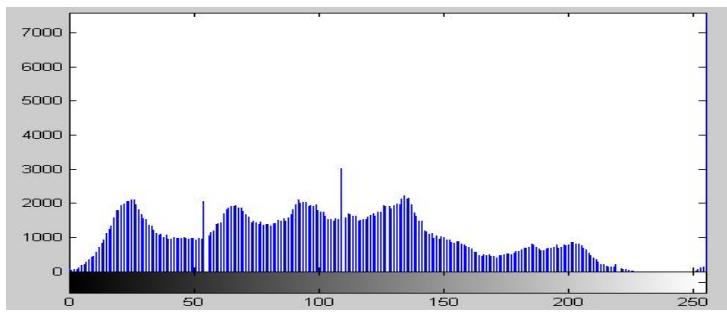


Figure 1.3: Histogram of a grey level image.

- **Image gradient**: Given an image I expressed by I : (x,y) $\rightarrow$ f(x,y), image gradient represents the derivative of f given its coordinates (x and y). It is an operator that allows us to locate areas where grey levels increase or decrease very quickly. These areas can be image contours characterized by a high variation of contrast.

- **2D and 3D images** 3-D image, also called *volume* is an image that provides the perception of depth. As we have mentioned above that the term *pixel* is reserved for 2D treatments, we use the term *voxel* for 3D treatments. In the medical area, 2D image is often called a *slice*. It is a view of the body part being visualized by the imaging technique and a volume is composed of a set of stacked and successive slices where every voxel is referenced by three coordinates: x, y, and z, such as z indicates the number of slice to which the voxel belongs, y indicates the row in the slice, and x indicates the column.

There are several categories of images such as satellite images, medical images and so on. In our work, we are interested to the medical area, more precisely to brain tumor images. So, next Section will present a brief description of brain images and brain tumors.

1.3 Brain images and their treatment

There are several imaging techniques commonly used in the medical area. They use different physical properties, highlight different kinds of information and are therefore used in conjunction with each other. We cite the *Computed Tomography*, *Scintigraphic Imaging*, *Magnetic Resonance Imaging*, etc. The choice of such technique depends on the target organ, also on doctors points of view. In our work we are interested in Magnetic Resonance Imaging (MRI) since it is certainly the imaging technique that allows the best distinction between tissues. and provides high-resolution images having a high accuracy. Moreover, the opportunity to rebuild volumes from images gives a point of view of the object that will be analyzed. These volumes are particularly useful for doctors during diagnosis. All these qualities make MRI the most used imaging technique in clinical routine [9]. For this reason we will adopt it in this work. In what follows we first give some details on MRI, then we present brain tumors and their characteristics.

1.3.1 Magnetic Resonance Imaging

MRI is a powerful visualization technique that allows the acquisition of internal anatomy images in a safe and non-invasive way. It is based on the principles of *Nuclear Magnetic Resonance* (NMR), and performs a vast array of different kinds of visualizations. This imaging technique has been of particular relevance in producing images of the brain, due to the ability of MRI to record signals that can distinguish between different soft tissues (such as *gray matter*[1] and *white matter*[2]) [9].

In brain imaging, two of the most commonly used MRI visualizations are T1-weighted and T2-weighted images. These weightings refer to the dominant signal (whether it be the T1 time or the T2 time) measured to produce the contrast observed in the image [9]. Since areas with high fat content have a short T1 time relative to water, T1-weighted images can be considered as visualizing locations of fat. In contrast, since areas with high water content have a short T2 time relative to areas of high fat content, T2-weighted images can be considered as visualizing locations of water. Also, a second T1-weighted image is often acquired after the injection of a contrast agent[3]. These contrast agent compounds usually contain an element whose composition causes a decrease in the T1 time of nearby tissue (*Gadolinium* is one example).

[1] A major component of the central nervous system, consisting of neuronal cell bodies, glial cells, etc.

[2] One of the three main solid components of the central nervous system, is composed of nerve cell processes (or axons), which connect various grey matter areas of the brain to each other.

[3] A substance used to enhance visualization of structures with specific anatomic properties.

More precisely, the intensity values seen on an MRI scan for a particular brain depends primarily on the content of that pixel versus neighbouring tissue and on other factors including the presence of fluids (i.e. cerebrospinal fluid) and fat. For example, on T1-weighted images, bone, air and water show as dark regions while fat has bright intensities and on T2-weighted images, water and fluid have high signal while bone, air, and fat remain dark [27] [69].

Figure 1.4 shows an example of T1 and T2-weighted images from axial[4] views of the same brain and Figure 1.5 illustrates a T1-weighted image before and after the injection of a contrast agent. In this case we propose a brain tumor image in order to show the difference between these weightings. As shown in this figure, on T1-contrast scans, tumors show as bright regions.

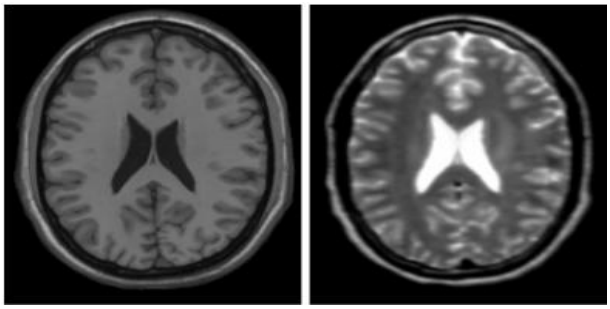


Figure 1.4: T1 and T2-weighted images respectively.

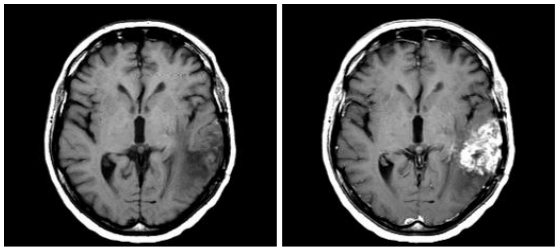


Figure 1.5: T1-weighted image before and after the injection of a contrast agent.

[4]Orientation of an MRI scan, an axial slice will be parallel to the feet of the patient with the left-hand side of the scan being the right-hand side of the patient.

1.3.2 Brain tumors and their diagnosis

Previously, we have generally detailed MRI and their visualization techniques (T1, T2, etc.). In this part, we propose some details of brain tumors and their characteristics in MR images since in our work we will focus on tumor images. So, this part will be a brief presentation of problems related to medical diagnosis of brain tumors. After explaining the medical terminology, we present the approach performed in the diagnosis of brain tumors. We note in particular the diversity of signs of brain tumors. Finally, we are interested in classification of these tumors.

A *tumor* is an abnormal mass resulting from uncontrolled and progressive multiplication of cells. A tumor can also be called *injury*, or *neoplasm*. Two kinds of tumor can be identified: the *benign* and the *malignant* tumor. The benign is not a cancer, it is characterized by a lack of growth, *invasion*[5] and *metastasis*[6]. However, it can grow worse and turns into malignant tumors. The latter is then characterized by its invasiveness and its proliferation of cells into surrounding tissues. Note that imaging techniques like MRI does not observe the dissemination of a tumor.

Our work is centered around the segmentation of *solid* brain tumors on images from MRI technique. The term 'solid' tumor means a tumor that is mainly formed of an individualized mass. However, the *soft* tissue tumors are tumors which affect blood cells and spread in the body, mainly in the bone marrow or ganglions or blood. The characterization of benign or malignant tumor is often performed during a *biopsy*[7] of pathological brain tissues. Thereafter, we use the term "tumor" regardless of their characteristic benign or malignant.

In fact, tumor diagnosis is within the complex clinical approaches leading thereafter, to a therapeutic decision. It is divided into several stages: suspicion, detection, observation, determination of its nature, etc. The clinical indications accompanying a brain tumor are varied. The symptoms vary greatly from one patient to another depending on the localization of tumor. The most common are headaches with *nausea*[8], physical and mental retardation, a muscle weakness, difficulty of speaking or visual discomfort. *Epileptic fit*[9] may be also the first manifestation of a tumor. These crisis may be limited to involuntary feelings or move-

[5]The spread of malignant cells to new sites in the body.
[6]The spread of cancer cells from the original tumor location to other parts of the body.
[7]The examination of tissues or liquids from the living body to determine the existence or cause of a disease.
[8]The state that precedes vomiting.
[9]A disorder of the central nervous system characterized by a loss of consciousness.

ments (known as partial crisis), which can sometimes cause a loss of consciousness (known as generalized crisis).

The observation of these signs leads the doctor to perform a number of examinations including radiological ones. From these examinations, signs of tumoral process are searched such as *opaque*, *clear* or *abnormal density masses*. Once this process is detected, doctors search to determine the number of tumors and their localization, then, for each of them, they look for different characteristics such as: size, shape, *infiltrant* or *limited* character, the localization in the brain and in comparison to the surrounding brain structures, composition by seeking especially if it is homogeneous or whether the tumors present cysts, fluid or *necrotic*[10] phenomena.

Also, various indirect signs may be observed namely, the presence of *edema*[11] near the tumor, topological and volumic changes of adjacent anatomic structures, etc. This last point is particularly the *central grey nuclei*[12], *white matter* and *ventricles*[13].

The observation of tumors depends on the imaging technique used to accomplish this task. For instance, the MR images offer medical information different from other techniques. More precisely, MRI visualisations techniques provide additional information of tumor characteristics. For example, in the T1-weighted images after injection of contrast, the presence of the 'enhancement'[14] can be a strong indicator of tumor location. There exist a large variety of kinds of brain tumors, and their appearance in MR images can vary considerably. Although some may be *fully 'enhancing'* (i.e. appear *hyper-intense* after the injection of a contrast agent) or may have an 'enhancing' boundary, many types of tumors display only *partial enhancement* or no enhancement at all (such as those examined in [22]).

Figures 1.6 and 1.7 show a fully enhancing tumor and a partially enhancing tumor. These two cases represent a T1-weighted slice before and after injection of Gadolinium. In the first case, the contrast product has perfectly enhanced the tumor. In the second, only a part of the

[10]The death of cells in an organ or tissue, due to injury, treatment side effects, or lack of nutrients; for example, cells at the centre of large tumors become necrotic due to the lack of nutrients.

[11]Swelling of any organ or tissue due to accumulation of excess fluid. Edema has many root causes, but its common mechanism is accumulation of fluid into the tissues. Cerebral edema is swelling of the brain which can occur as the result of a head injury, cardiac arrest, a brain tumor, etc.

[12]A set of nerve structures, which included the ganglia cells but also other sub-cortical gray structures.

[13]A cavity inside the body, which contains a fluid.

[14]The area of the tumor appears clearly visible after the injection of the contrast agent gadolinium.

tumor is highlighted. This illustrates the importance of acquisitions before and after injection of contrast agent, but also the caution when analyzing images.

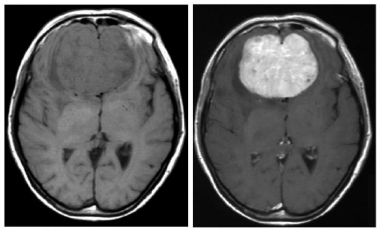


Figure 1.6: Fully enhancing tumor after injection of the product of contrast [12].

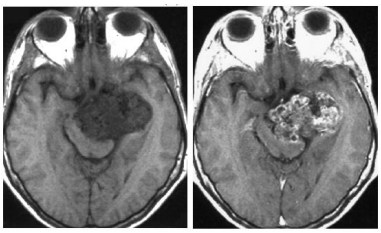


Figure 1.7: Partially enhancing tumors after injection of the product of contrast [12].

Moreover, *edema (swelling)* can also be observed in many kinds of primary tumors, and appears as hyper-intense in T2 images. The conjoint use of different acquisitions (T1 and T2) is particularly interesting in diagnosis of brain tumors. Indeed, it is common that a tumor appears only partially on MR images. The solution is trying many acquisitions and eventually acquisitions using a product of contrast. Each acquisition which provides a complementary information, but also redundant, can be considered as a particular observation of the brain.

Note that these observations can be sometimes conflicting, for example when a tumor is completely absent on one image and appears on another. The conjoint use of these images by doctors, enables them to synthesize information and to have a complete information about it.

The following examples illustrate some brain tumors:

Example 1.1. ***Astrocytomas case****: The case presented in Figure 1.8 corresponds to a low grade astrocytomas. This kind of tumor is about 5% of intracranial tumors and is usually characterized by hemispherical, encapsulated and infiltrant mass. Often accompanied by swelling, the low grade astrocytomas are hyper-intense in T2-weighted images and have a low or medium contrast in T1-weighted images. Tumor diagnosis consists of several stages. In this particular case, an MRI radiological examination is accompanied by a surgical examination and histological examination.*

In fact, images are obtained by MRI of type T1 and T2 respectively (Figure 1.8). The arrows shown in these images indicate particular interests to doctors. Each of these images provides specific and useful information for doctors during tumor diagnosis.

In the one hand, the T1-weighted image highlights the non encapsulated and the non homogeneity aspect of the tumor (β). In the other hand, the T2-weighted image presents an hyper intense tumoral mass and suggests the presence of another mass inside the main mass (γ). Thus, the radiological diagnosis emphasizes an heterogeneous and infiltrant tumor having non delimited contours. These characteristics represent the astrocytomas ones.

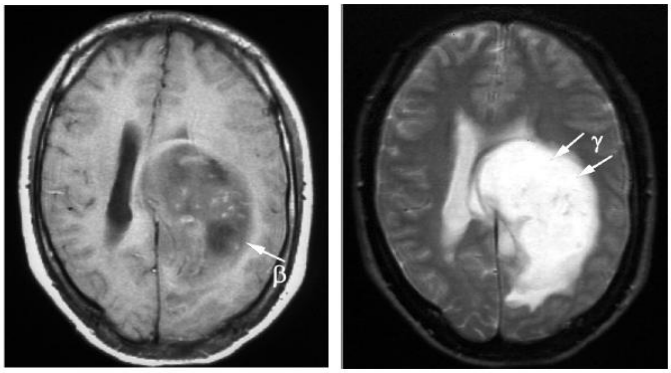


Figure 1.8: Astrocytomas of low grade. Left: T1 image. Right: T2 image [12].

This radiological examination is then completed by a surgical one. This latest shows a significant infiltration of the tumor to the callosum corpus and the vascular system. The borders of the tumor are contrasted : some of them are very well defined and the others are fuzzy. Finally, an histological examination confirmed that it is an astrocytomas of grade II.

Example 1.2. ***Glioblastoma case****: The tumor shown in Figure 1.9 corresponds to a glioblastoma. This kind of tumor, of Grade IV, is relatively common. It represents 20% of all intracranial tumors and occurs frequently in patients between 50 and 70 years. Glioblastoma masses are often heterogeneous (i.e. they have necrosis and are accompanied by edema). As in the previous case, the diagnosis is composed of the examination of the MR images, a surgical and histological examinations.*

The diagnosis of this tumor requires a T1, T2 and a T1 with injection of gadolinium (Figure 1.9). Their observation highlights the complementary and the redundant character of these images. Indeed, T2-weighted image shows a solid mass with an hyper-intense signal on the frontal area ($\alpha 1$) and an hypo-intense signal on the superior part ($\alpha 2$). Moreover, we note that the tumor is surrounded by edema ($\alpha 3$) that compresses the left lateral ventricle. The T1-weighted image indicates the presence of an hypo intense tumor(β). This low intensity

leads to the achievement of new images with adding gadolinium which show a strong contrast in the surrounding of the tumor (γ).

The observation of this clinical case is particularly interesting because it proves the need for conjoint use of different acquisitions.

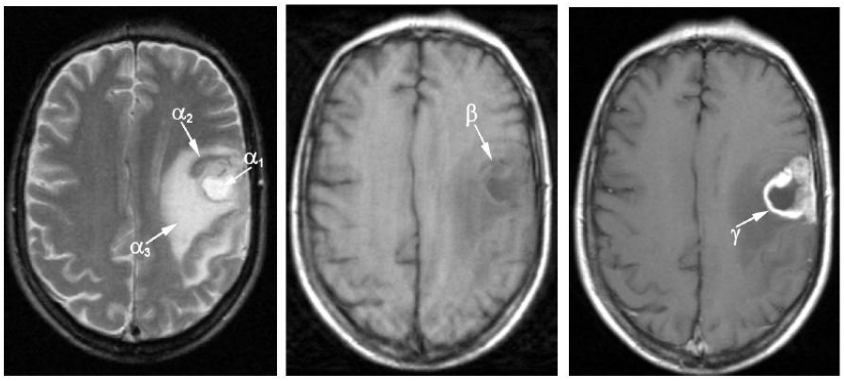


Figure 1.9: Glioblastoma case. Left: T2 image. Middle: T1 image. Right: T1 Gadolinium [12].

1.3.3 The diversity of brain tumors

The *World Health Organization* (WHO) has established a tumors classification of the central nervous system in order to lead eventually to an international standard [34]. In this classification summarized in Table 1.1, the name of each tumor indicates the origin cell. For instance, the *astrocytomas* are derived from the *ancillary cells* surrounding neurons, etc. Each kind of tumor is then linked to a grade, which varies from I to IV, reflecting the general prognosis on the future behavior of tumor. Tumors known as low grade (I and II) have the best prognosis. Tumors of grade I are limited and have a better prognosis than tumors of grade II which are not defined (often called infiltrated) and whose complete surgical excision is often impossible due to the infiltration, while tumors known as high-grade (III and IV) are associated with important malignancy.

1.4 Image segmentation

Image analysis usually refers to computer image processing with the objective of finding objects presented in these images. Image segmentation is one of the most critical tasks in automatic

	Grade I	**Grade II**	**Grade III**	**Grade IV**
Astrocyte tumors				
Pilocytic astrocytoma	×			
Astrocytoma		×		
Pleomorphic xanthoastrocytoma		×	×	
Anaplastic astrocytoma			×	
Glioblastoma				×
Oligodendrocyte tumors				
Oligodendrogliomas		×		
Anaplastic oligodendrogliomas			×	
Oligoastrocyte tumors				
Oligoastrocytoma		×		
Anaplastic oligoastrocytoma			×	
Glial and neuronal tumors				
Gangliocytoma	×			
Ganglioglioma	×	×		
Desmoplastic infantile ganglioglioma		×		
Central neurocytoma	×			
Pineal tumors				
Pinealocytom		×		
Pinealoblastoma			×	
Tumors of cranial and spinal nerves				
Schwannoma	×			
Malignant tumor of perineural sheaths			×	×
Meninges tumors				
Meningioma	×			
Papillary meningioma		×	×	
Hemangiopericytome		×	×	
Anaplastic meningioma			×	

Table 1.1: Tumors classification by grade [34].

image analysis. It plays a crucial role in many imaging applications and consists on subdividing an image into its constituent parts and extracting the regions of interest that should be homogeneous with respect to some characteristics. These regions should be relevant, i.e. should correspond to the real and significant objects in real world. Formally, the segmentation can be defined as follows:

Definition 1.1. *Let X be the domain of the image and f is the function that associates a value $f(x, y)$ to each pixel. If we define a predicate P in all parties of X, the segmentation of X is defined as the partition of X to n sub sets $R_1, ..., R_n$ such as:*

1. $X = \cup_{i=1}^{n} R_i$
2. $\forall i \in \{1, ..., n\}, \quad R_i$ *is connected*
3. $\forall i \in \{1, ..., n\}, \quad P(R_i) = true$
4. $\forall i, j \in \{1, ..., n\}^2,\ R_i$ *is adjacent to* R_j *and* $i \neq j \Rightarrow P(R_i \cup R_j) = false$
   where $\cup$ *represents the union of disjointed sets.*

The predicate P is used to test the homogeneity of R_i. These subsets constitute image regions. Thus, image segmentation is image decomposition into a set of homogeneous regions and the criterion of homogeneity P depends on the nature of problem.

The first condition implies that every pixel of the image belongs to only one region. This means that the segmentation algorithm does not end before treating all pixels. The second condition implies that every region should be connected. The connection between regions is induced by the neighborhood defined on images. The third condition implies that each region should be homogeneous. Finally, the fourth condition is a condition indicating that fusion of two regions should not be homogeneous. It is important to note that the number n of regions forming the partition of the image remains undetermined. So there can be several possible segmentations for a predicate P.

Several segmentation methods exist in the literature. The question is *how do we prefer one method or the other for a specific problem?* In fact, these methods vary widely from one application to another. A classification of these methods was proposed in [37] on the basis of five criteria namely: *region*, *contour*, *shape*, *structural approaches* and *graph theory* (see Figure 1.10).

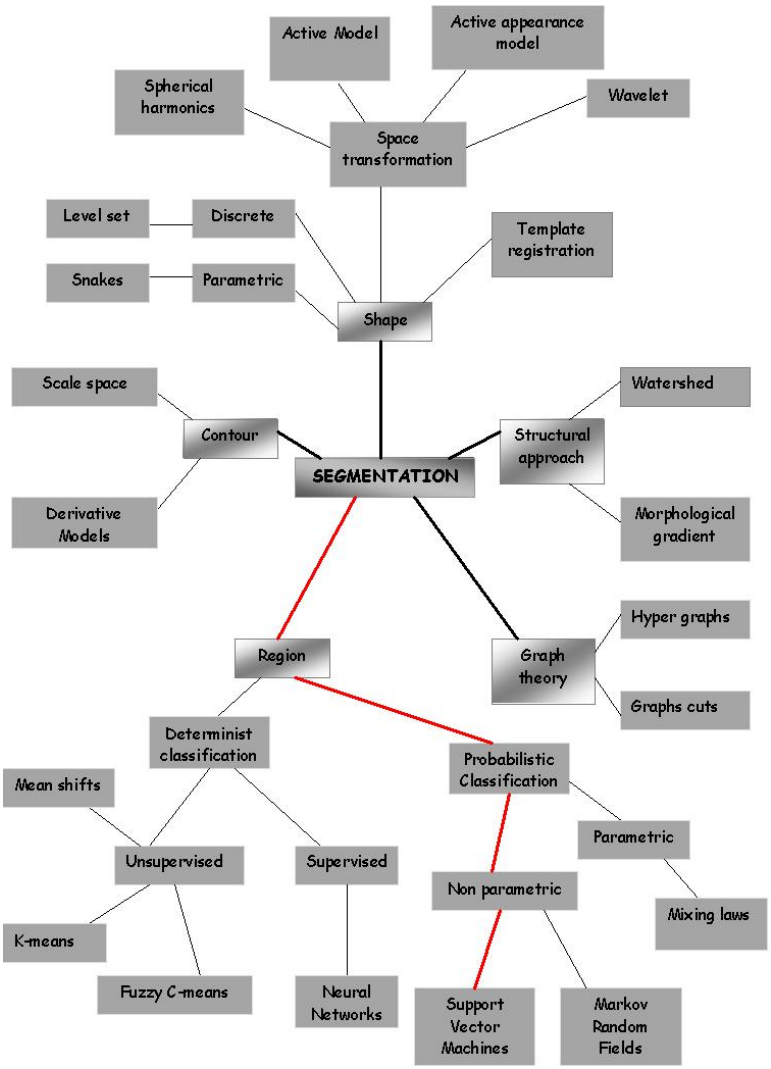


Figure 1.10: Classification of image segmentation methods [37].

In our work, we are interested in medical image segmentation, more precisely the brain images issued from the MR device. In fact, MR images discussed above are considered as a valuable source of information allowing the analysis and the monitoring of diseases since they provide a non invasive, contrasting representations and a high resolution of the brain anatomy, allowing its observation in vivo. However, the systematic and manual study of these images is long and tedious and it also suffers from the variability of specialists interpretations. Thus, several researches have been carried out to automate image analysis; and segmentation is considered as a starting point of this process.

The context of our study is based particularly on the aid in brain tumors diagnosis. Our

aim is to precisely detect the tumor. In general, tumors segmentation problem in MR images can be reached from two angles:

- Consider the tumor as the only goal of the segmentation and using a method completely dedicated to this task,
- Consider the tumor, and possibly edema, as entities of the brain and the segmentation aim is then to isolate all brain structures.

In our work, we will focus on the first objective since it will allow us to reach the second one. This study highlights a number of difficulties:

- First, the segmentation of such images, even in the absence of brain tumor, is a non trivial task, since the brain is a complex organ composed of different anatomical structures. In addition, the physical contour between these structures is particularly difficult to define, even for an expert.
- Secondly, even if MRI is a quality visualization tool of in vivo anatomical structures, it involves defects that degrade the quality and the acquisitions of images like any imaging device. The main defect is particularly the noise which will influence and perturb the automatic digital processing of images.
- Finally, segmentation and detection of brain tumors are difficult tasks due to the complexity of the sought objects since, as shown in Table 1.1, the number of tumors is variable, their forms and their visual characteristics are varied. In addition, a brain tumor is often only partially visible on an image. Its observation requires, in most cases, multiple acquisitions which form together a synthesis information in order to detect and locate the border of tumor areas.

So to handle these characteristics, the segmentation method should be accurate and reliable. In what follows, we will treat the segmentation as a classification problem since the constraint relative to the connectivity of regions (see Definition 1.1) is removed in our case due to the fact that tumor regions can be scattered all over the brain. Therefore, the problem can be considered as a *pixel classification* problem which is often a desirable goal in medical images, especially when disconnected regions of interest belonging to the same class should be extracted [50]. For this reason, we will adopt the *region* criteria and more precisely the *Support Vector Machines (SVM)* as a segmentation method (see Figure 1.10). Our choice is especially motivated by its fastness, robustness in generalization properties and its capacity to handle voluminous data [62]. Also, many studies have indicated that SVM outperforms other two-class classifiers in most cases [32] [58] .

1.5 Thesis contribution

Our aim in this work is the extraction of tumor region from brain images. To ensure this task, images should first be pre-processed in order to improve their quality. Moreover, classification methods require an attributes set characterizing each instance which corresponds in our case to a features set relative to each pixel. This will be done via an extraction step which consists on computing image-based-features. Therefore, our segmentation method will be based on five main phases, namely *pre-processing*, *features extraction*, *SVM training and testing*, *segmentation* and *post-processing*. The whole process is illustrated by the diagram of Figure 1.11 which can be detailed as follows:

- **Pre-processing phase**: This phase is composed of six steps:

    1. *Acquisition*: this step is ensured via MRI and represents a primordial step since the remainder of the process will depend on the quality of acquired images. Note that the MR images can be axial, coronal[15] or sagittal [16], their number and their scans orientations depend on tumor localization. The output of this step is a set of 2D slices.
    2. *Reconstruction*: this step maps the 2D sequential slices of each patient to an entirely volume (i.e. 3D image).
    3. *Co-registration*: this step spatially aligns volumes of each patient having different modalities (i.e. T1 and T2).
    4. *Template registration*: having as input co-registered volumes, this step consists on aligning modalities (T1 and T2) with a template image in order to average signals from brain images of different subjects. Note that the two last steps are ensured in three dimensions, this is due to the fact that utilizing 3-D information allows a more accurate registration than the alignment of 2-D images [62].
    5. *Extraction of slices of interest*: in this step, we will choose the 'interesting' slices from the registered volumes, i.e. the slices where we can see tumor regions. The output of this step is a set of 2D pathological images.
    6. *Noise reduction*: this step improves the quality of data through the application of filters.

[15] Orientation of an MRI scan, a coronal slice will be perpendicular to the feet and parallel to the shoulder line of the patient with the left hand side of the scan being the right hand side of the patient.

[16] Orientation of an MRI scan, a sagittal slice will be perpendicular to the feet and perpendicular to the shoulder line of the patient.

- **Features extraction phase**: Given pre-processed images, this phase refers to various quantitative measurements used for decision making regarding the class of pixels (normal or abnormal). More precisely, for each pixel, we should compute a feature vector. Thus, the output of this phase will be unlabeled data set.

- **Training and testing phase**: In this phase, tumor regions for each image in the training set will be selected and validated by the radiologist. In order to select these regions, *active contour* models [15] [16] will be used to define anatomical contours of tumor regions, due to their ability to approximate accurately the random shapes of boundaries.

  Once the labeled training set containing normal and tumourous pixels is constructed, SVM can be trained, so that the output of this step will be the classification model allowing the discrimination between normal and tumor pixels.

- **Segmentation phase**: To ensure this phase, we propose two steps:

  1. Classification step: in this step, the classification model generated from the previous phase will be used to affect the tumor or the normal class for each pixel in the new images which are pre-processed.
  2. Tumor extraction step: after that, the set of tumor pixels is grouped to form the tumor region. To clarify our output segmentation, we will propose for each segmented image, a binary image where black region is image background and white region represents the tumor one.

- **Post-processing phase**: Since the classifier may not be perfect at classifying unseen test pixels, it may make errors [62]. To remove these errors, a *morphological filter* is applied as a post-processing technique merging the connected regions while removing some isolated non-tumor regions.

1.6 Conclusion

MR images are used in tumor diagnosis, monitoring tumor progression, planning treatments, and monitoring responses to treatment which often involves a combination of surgical resection, radiation therapy and chemotherapy [45]. The digitization of such acquisitions provides the development of automated systems to aid doctors in their diagnosis. In the particular case of brain tumor diagnosis, this assistance consists on detecting and locating tumors on

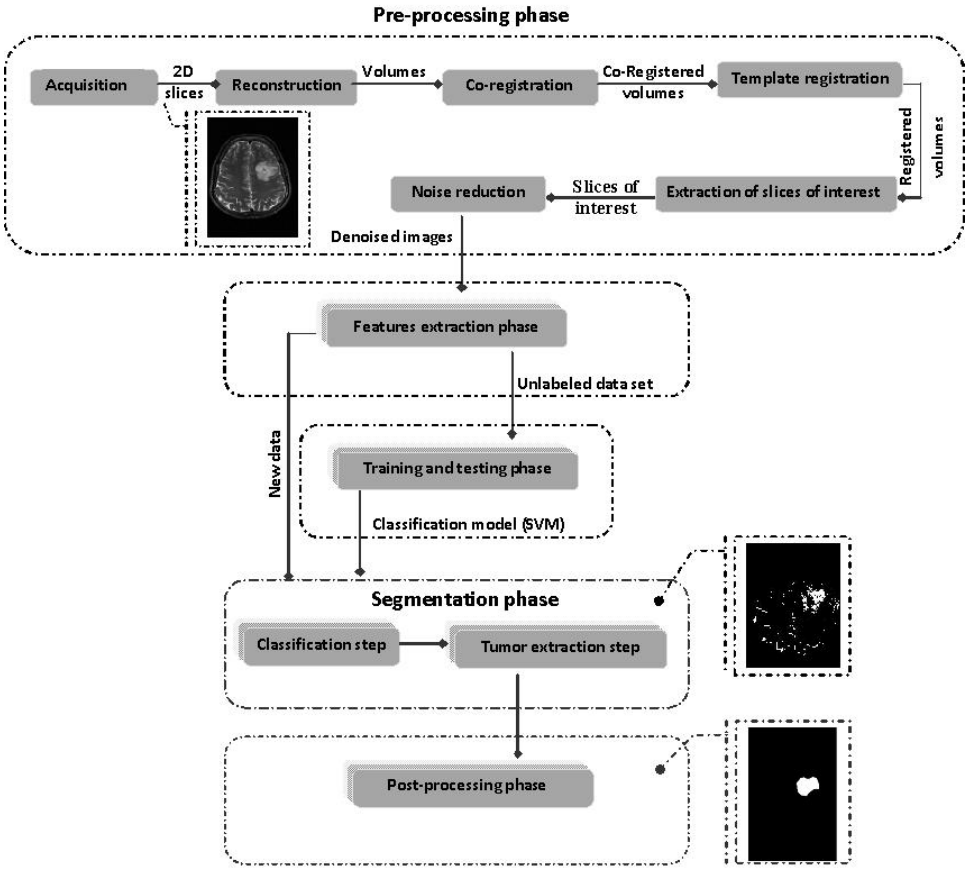


Figure 1.11: Diagram illustrating our segmentation system.

radiographic images. Therefore, our aim is to develop a segmentation method able to separate tumor areas.

In the following Chapters, we will discuss each phase of our system. Thus, in next Chapter, we start with the classification method we have selected for segmenting brain tumors, namely Support Vector Machines.

Chapter 2

Support vector machines

2.1 Introduction

Support vector machines (SVM) is a machine learning method introduced by Cortes and Vapnick [18] developed to solve both binary and multi-classification problems. It works under a supervised mode. The main idea behind this method is to separate classes with an hyperplane that maximizes some distance between them. This classifier has gained popularity due to its solid theoretical foundation and was applied to several fields such as [65]:

- Image processing:
  - Object recognition,
  - Image classification,
  - Content-based image retrieval.
- Data Mining:
  - Data classification,
  - Feature extraction,
  - Noise classification and reduction.
- Speech processing:
  - Speaker identification/verification,
  - Speech recognition, synthesis.

The aim of this Chapter is to introduce SVM. It is organized as follows: Section 2.2 introduces some notations used in this Chapter. Section 2.3 will detail binary classification by SVM. Finally, binary SVM will be extended to multi-classification SVM in Section 2.4.

2.2 Notations

We first give some notations used in this Chapter:

- n denotes the number of points in data set.
- x_i denotes a data point where $i = (1,2, \ldots \mathrm{n})$.
- y_i denotes the class label of x_i. It can be $+1$ or -1 for binary problems. In this Chapter, data having class $+1$ (respectively -1) will be represented by squares (respectively balls).
- C_i denotes the class assigned to x_i for multi-class problems.

2.3 SVM for binary classification

SVM has been initially proposed to handle binary classification problems (i.e. where y_i takes only two values). Classification by SVM is based on finding a discriminant *hyperplane* that maximizes the *margin* (or distance) to both classes in the feature space. To find this hyperplane, we have to estimate a function f using a training data which consists on a set of labeled data instances. This function f should be able to classify new examples of the testing set (data instances that don't contain target values). The application of SVM depends on the nature of training data. In fact, we can distinguish two kinds of data:

- *Linearly separable data*: are a kind of data which can be separated by a simple line. As shown in Figure 2.1, the separating line is dividing data and leaving all points with the same label on the same side of the separator.

- *Non-linearly separable data*: the separator between positive and negative points in non-linearly separable data, can not be a straight line. Figure 2.2 shows an example of these data.

In what follows we will first consider the simplest case of a linear classifier and linear separable problems. Then linear classifier and non separable problems will be handled. Finally,

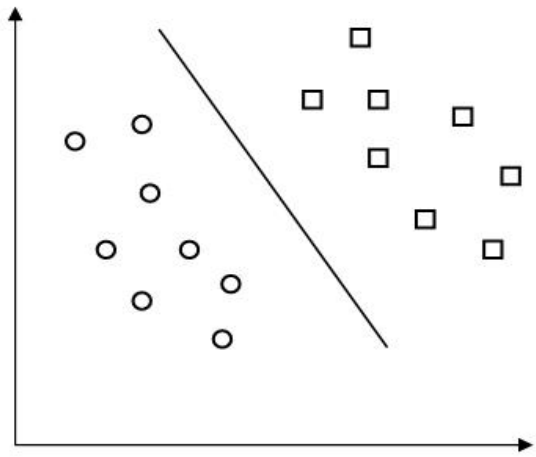


Figure 2.1: Illustration of a linear hyperplane.

the most general, interesting and useful case will be presented, namely non linear classifier and non separable problems.

2.3.1 Linear classifier and linearly separable problems

This case is the simplest one since data are linearly separable and the hyperplane that separates data will be a straight line as shown in Figure 2.3.

This means that the separating hyperplane is defined by a *weight vector* w which is normal to the hyperplane and a *bias vector* b. The points x_i which lie on the hyperplane satisfy Equation 2.1:

$$w\,\mathrm{x}_i \;+\; \mathrm{b} \;=\; 0 \tag{2.1}$$

The aim of SVM is to find the optimal separator that maximizes the margin. In Figure 2.3, we have infinite number of classification solutions, i.e. multiple separating hyperplanes, so *how to choose the optimal one?* This is the fundamental question that *support vectors* try to answer as we shall see later.

In fact, the *margin* is defined as the minimal distance of an object to the decision surface, i.e. the hyperplane. In other words, the optimal separating hyperplane separates classes and maximizes the distance between the closest points from each class. These points are called *support vectors*. Figure 2.4 shows the difference between a small margin and a larger one. A

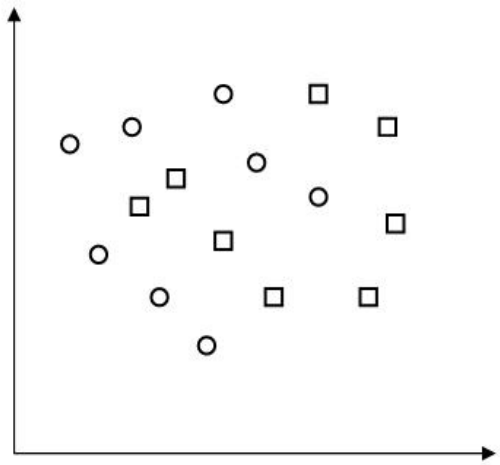


Figure 2.2: Illustration of a non-linearly separable case.

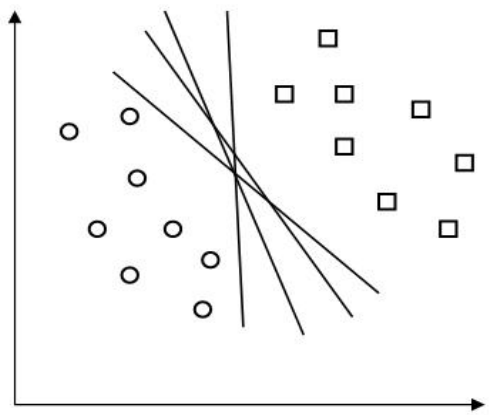


Figure 2.3: Examples of linearly separating hyperplanes.

better generalization capability is expected from a separator having larger margin (see image(b)). Note that filled squared and filled ball data points represent *support vectors*.

Support vectors which represent the key parameter of SVM appear in the hyperplanes (H_1 and H_2) parallel to the optimal one as shown in Figure 2.5 and data points lying on these hyperplanes are expressed by Equations 2.2 and 2.3 depending on the class assigned to each data point.

$$w\, x_i + b = +1 \tag{2.2}$$

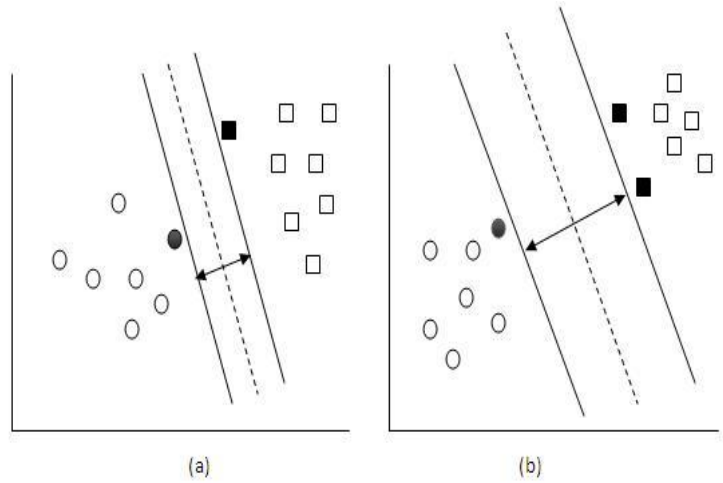


Figure 2.4: (a) A separating hyperplane with small margin. (b) A separating hyperplane with larger margin.

$$w\, x_i + b = -1 \tag{2.3}$$

In general, the equation of each hyperplane is expressed by:

$$w\, x_i + b = y_i \tag{2.4}$$

such that the value of y_i differs from one hyperplane to another. Two cases can be presented:

- Optimal hyperplane: y_i will be 0,
- Boundary hyperplane: y_i will be +1 or -1.

Now, we describe the mathematical derivation of SVM. We consider two objects x_1 and x_2 from different classes with $|wx_1 + b|$=1 and $|wx_2 + b|$=1, respectively (Figure 2.5). Then, the margin is given by the distance of these two objects, i.e. $\frac{2}{||\overrightarrow{w}||}$ (see Proof 2.1).

Proof 2.1. *[18] Given two data points x_1 and x_2 belonging to classes +1 and -1, respectively. In order to compute the margin, we have to calculate the distance between these points as follows:*

$wx_1 + b$=*1 and* $wx_2 + b$=*-1*

$\Rightarrow (w(x_1 - x_2))$=*2*

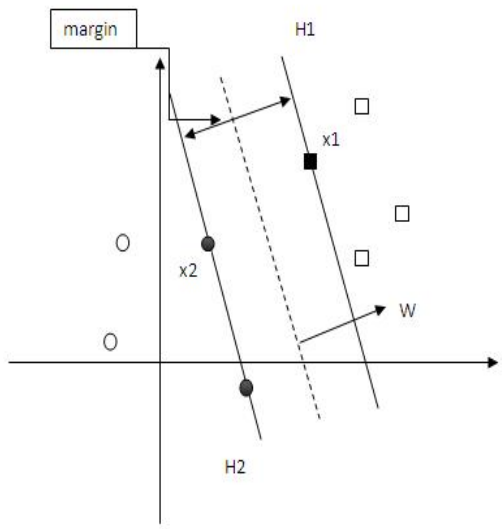


Figure 2.5: Optimal hyperplane.

$\Rightarrow \frac{w}{||\overrightarrow{w}||}.(x_1 \text{ - } x_2)) = \frac{2}{||\overrightarrow{w}||}$

So, the margin is equal to $\frac{2}{||\overrightarrow{w}||}$.

Thus, to construct the optimal hyperplane, we have to solve the following optimization problem:

$$\min \frac{1}{2} \cdot ||\overrightarrow{w}||^2 \tag{2.5}$$

$$\text{subject to } y_i(w\, x_i + b) >= 1,\ i = 1, 2, ...n$$

If $y_i = +1$ then $w\, x_i + b >= 1$, this means that the class of x_i is $+1$. Else $w\, x_i + b <= 1$, so the class will be -1. In other words, if we want to decorticate the constraint, we have:

- $w\, x_i + b >= 1\, \forall\, x_i \in\ class\ +1$,
- $w\, x_i + b <= 1\, \forall x_i\ \in class\ -1$.

The problem described in Equation 2.5 is a *convex optimization* problem that consists of a *quadratic criterion* with linear inequality constraints. To solve this problem, the *Lagragian* method is used which introduces for each constraint a new unknown scalar variable, α_i called the *Lagrange multiplier*; and forms a linear combination such as the multipliers are considered as coefficients. This reduced the constrained problem to an unconstrained problem [71]. So, the optimization problem will be expressed by a *Lagrangian primal Lp* as follows:

$$Lp = \frac{1}{2}||\overrightarrow{w}||^2 - \sum_{i=1}^{n}\alpha_i[y_i(wx_i + b) - 1] \tag{2.6}$$

where α_i are the Lagrange multipliers. The Lagragian Lp should be minimized with respect to w and b as demonstrated in Proof 2.2.

Proof 2.2. *[18] To minimize the Lagragian Lp with respect to w and b, we have to search the saddle point using Equation 2.6 as follows:*

(a) $\frac{\partial Lp}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$

(b) $\frac{\partial Lp}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$

From Equations (a) and (b), we can deduce that: $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ *and* $\sum_{i=1}^{n} \alpha_i y_i = 0$*, respectively.*

In some cases, dual problem is much easier to solve than the primal one [26]. So, instead of minimizing Lp with respect to w and b, we maximize Ld with respect to α_i. Taking the derivatives with respect to b and w as shown in Proof 2.2 and resubstituting back in the primal equation (Equation 2.6) gives the *Wolfe dual Lagrangian* expressed by Equation 2.7 [11].

$$Ld = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, x_i \, x_j \tag{2.7}$$

In the dual space, we have to maximize Ld subject to the following conditions:

$$\begin{cases} (i)\ \alpha_i >= 0 \\ (ii) \sum_{i=1}^{n} \alpha_i \, y_i = 0 \end{cases} \tag{2.8}$$

Solving Equation 2.7 with constraints of Equation 2.8 determines the *Lagrange multipliers* α_i. Two cases can be presented:

- **Case 1**: if $\alpha_i > 0$ this indicates that the input vector x_i is on the hyperplane boundaries, i.e. it is a *support vector*.
- **Case 2**: if $\alpha_i = 0$ this indicates that x_i is not on the boundaries [71].

In other words, we retain support vectors that lie on hyperplane boundaries (Figure 2.5) while all remaining training examples are considered irrelevant to the solution and they are removed from the training set. This represents a very important aspect of SVM, which makes it an effective classifier [36]. Consequently, the hyperplane is determined by a small subset of

the training set, i.e. support vectors since data points which are far away have no influence on the results, because it is obvious to which class they belong. Hence, SVM can be used to summarize information contained in a data set by the produced support vectors [26].

Finally, the hyperplane decision function can be written as [71]:

$$\text{f(x)= sign } (\textstyle\sum_{i=1}^{l} \alpha_i \; y_i \; x_i \; x + \text{b} \;) \tag{2.9}$$

where l is the number of support vectors while b is computed by Equation 2.10 using any support vector (x_i, y_i) (Equation 2.10 is deduced from Equation 2.4).

$$\text{b= } y_i \text{ - } \textstyle\sum_{j=1}^{l} \alpha_j \; y_j \; (x_i \; x_j) \tag{2.10}$$

The following example illustrates the linear case of SVM:

Example 2.1. *Suppose that we have 4 data points having two dimensions, such that the positive ones have these coordinates:* $x_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, x_2 = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$ *and the negatives ones are :* $x_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, x_4 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$. *These data points can be separated by a linear hyperplane, i.e. a straight line as shown in Figure 2.6. Now we have to seek the hyperplane equation by solving*

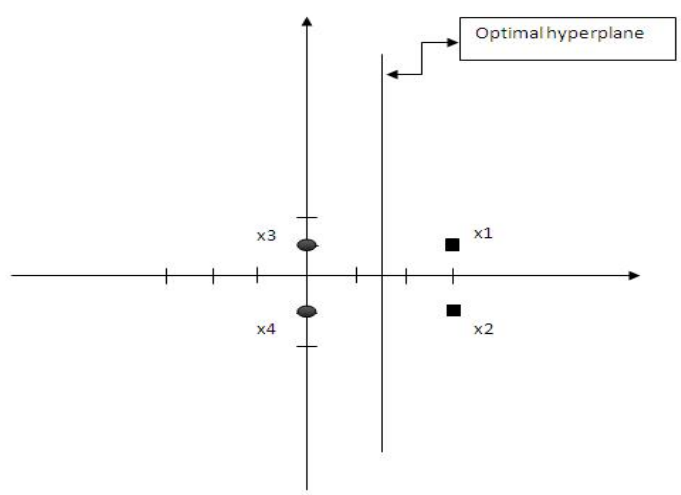


Figure 2.6: Illustration of data points and optimal hyperplane representations.

the optimization problem of Equation 2.7:

max $\sum_{i=1}^{4} \alpha_i$ - $\frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j x_i x_j$

subject to $\alpha_i >= 0$ *and* $\sum_{i=1}^{4} \alpha_i y_i = 0$.

First, we have to substitute x_i *and* y_i *by their values in the optimization problem as follows:*

max $10{\alpha_1}^2 + 10{\alpha_2}^2 + {\alpha_3}^2 + {\alpha_4}^2 + 16\ \alpha_1\ \alpha_2 + 2\ \alpha_1\ \alpha_4 + 2\ \alpha_2\ \alpha_3 - 2\ \alpha_2\ \alpha_4 - 2\ \alpha_3\ \alpha_4 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$.

Second, we take the derivative for each α_i*, and the resulting system is as follows:*

$$\begin{cases} 20\alpha_1 + 16\ \alpha_2 + 2\ \alpha_4 & = -1 \\ 16\alpha_1 + 20\ \alpha_2 + 2\ \alpha_3 - 2\ \alpha_4 & = -1 \\ 2\alpha_2 + 2\ \alpha_3 - 2\ \alpha_4 & = -1 \\ 2\alpha_1 - 2\ \alpha_2 - 2\ \alpha_3 + 2\ \alpha_4 & = -1 \end{cases}$$

Third, we shall find support vectors by solving this system. Recall that each data point having a not null value of α_i *is a support vector. In our case, we found* $\alpha_1 = -1$, $\alpha_2 = 0.8889$, $\alpha_3 = 1$ *and* $\alpha_4 = 2.3889$. *So, all data points are support vectors.*

Finally, we compute the value of the bias b using any support vector, for instance using the data point $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, *we found* $b = -0.5$ *since:* $f\begin{pmatrix} 0 \\ 1 \end{pmatrix} = -1$,

$$\Rightarrow -1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} (1) + 0.8889 \begin{pmatrix} 3 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} (1) + 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\Rightarrow (-1) + 2.3889 \begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} (-1) + b = -1,$$

$$\Rightarrow b = -0.5.$$

Therefore, the decision function will be an equation of a straight line as shown below:

$f(x) = (\sum_{i=1}^{4} \alpha_i\ y_i\ x_i\ x - 0.5\ )$

Now suppose that we want to classify a new point, it will be very simple in that case because we have to compute $f(x_i)$ *and check its polarity, i.e. positive/negative value allows us to assign*

x_i to $+1/-1$, respectively during the decision phase.

For example, we seek to classify the data point $x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. *The decision function computed above* ($\sum_{i=1}^{4} \alpha_i\ y_i\ x_i\ x$ - 0.5) *will be used and we obtain:*

$$\Rightarrow f \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 1 \begin{pmatrix} 3 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} (1) - 0.8889 \begin{pmatrix} 3 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} (1) + 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} (-1)$$
$$+ 2.3889 \begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} (-1) - 0.5$$
$$\Rightarrow f \begin{pmatrix} 0 \\ 0 \end{pmatrix} = - 0.5.$$

Once this step is assessed we can classify this data point to the negative class (-1) since the decision function value is negative. Moreover, we can check our result by the hyperplane in Figure 2.6.

2.3.2 Linear classifier and non separable problems

Now we consider the case in which we still look for a linear hyperplane but a separating hyperplane does not exist. In order to deal with this case, Cortes and Vapnick [18] introduced a new set of non negative variables ε_i called *slack variables*. In fact, these variables allow constraints violation for some objects. Therefore, these objects are permitted to be on the wrong side of the hyperplane.

As shown in Figure 2.7, error is allowed in classification. For example if $\varepsilon_i = 0$ this indicates that there is no error for x_i. In this case, the objective function differs, we have to minimize both the classifier capacity via $||\overrightarrow{w}||$ and the sum of slack variables $\sum_{i=1}^{n} \varepsilon_i$ i.e. the number of training errors. Here in the linearly non-separable case, the SVM classifier is called *soft margin classifier* [65]. It minimizes:

$$\min \tfrac{1}{2} \cdot ||\overrightarrow{w}||^2 + C \textstyle\sum_{i=1}^{n} \varepsilon_i$$
$$\text{subject to } y_i(wx_i + b) >= 1 - \varepsilon_i,\ i = 1, 2, ...n \tag{2.11}$$

where C is a constant chosen by the user that measures the penalty. This parameter is a regularization parameter that controls the trade off between maximizing the margin and minimizing the training error. A larger C corresponds to assigning a higher penalty to errors.

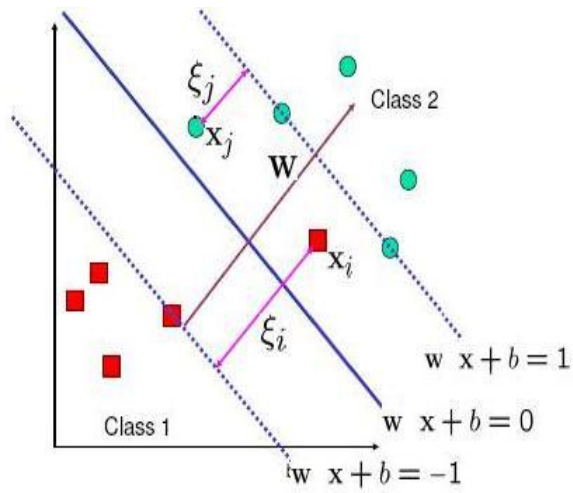


Figure 2.7: Slack variables.

Applying the *Lagrangian method* and transforming the primal problem to the dual, we find the same optimization problem (Equation 2.7). The only difference with linear separable case is that there is an upper bound C on α_i, i.e.

$$0 <= \alpha_i <= \mathrm{C} \tag{2.12}$$

2.3.3 Non-linear classifier and non separable problems

Previous Subsections have only treated linear hyperplanes, which are definitely not appropriate for complex tasks because in practise a separating hyperplane may not exist. For this reason, the extension to more complex decisions functions is important. This extension is done by mapping the input variable x_i into a higher dimensional feature space F and working with a linear classification in that space as shown in Figure 2.8 [47].

Hence, if the hyperplane separating the two classes is not linear, data points can be transformed to another high dimensional feature space where the problem is linearly separable. If the transformation to the high dimensional space is $\oint()$ then the *Lagrangian function* can be expressed by:

$$Ln = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \oint(x_i) \oint(x_j) \tag{2.13}$$

The dot product $\oint(x_i)$ $\oint(x_j)$ in that high dimensional space defines a *kernel function* $K(x_i, x_j)$ and therefore it is not necessary to know $\oint()$ as long as it is known that the kernel

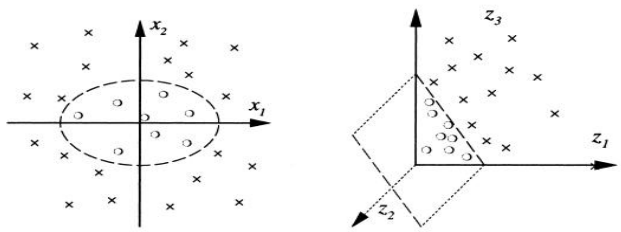


Figure 2.8: Example of higher dimensional feature space.

function is a dot product in some high dimensional feature space [65]. This case is presented in Figure 2.9 where data which can not be represented by a linear hyperplane are transformed into another space where a linear separator exists [39].

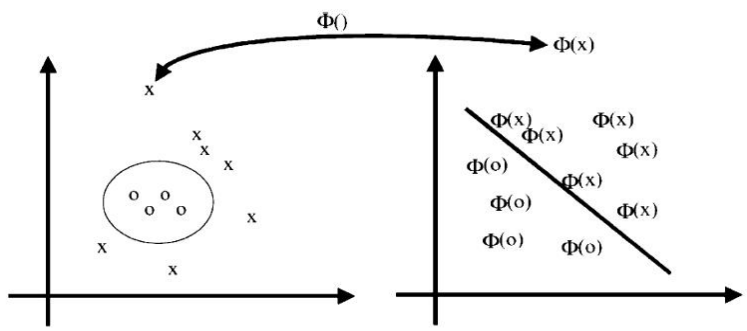


Figure 2.9: A non-linear separator transformed into a linear one.

Formally, a kernel function has an equivalent representation as expressed by Equation 2.14:

$$\mathrm{K}(\mathrm{x}_i, x_j) = \oint(x_i)\ \oint(x_j) \tag{2.14}$$

Some commonly used kernels are presented in Table 2.1 where σ, β and b are kernel parameters.

Note that every kernel function should satisty *Mercer's condition* expressed by Proposition 2.1. This condition describes the mathematical properties to check if a kernel function is a dot product in some space, but it does not tell us how to construct $\oint$ or even what F is used [10].

Kernel	$K(x, x_i)$
Linear	$x.x_i$
Polynomial of degree d	$(x.x_i + 1)^d$ or $(x.x_i)^d$
Radial Basis Function (RBF)	$exp(\frac{-\|\|x-x_i\|\|^2}{2\sigma^2})$
Multi-Layer Perceptrons	$tanh(\beta.x_i.x + b)$

Table 2.1: Commonly used kernel functions.

Proposition 2.1. *The Mercer's condition is as follows:*
there exists a mapping $\oint$ *and* $K(x_i, x_j) = \oint(x_i)\ \oint(x_j)$ *if and only if for any* $g(x)$ *such that* $\int g(x)^2 dx$ *is finite then* $\int$ *K(x,y) g(x) g(y) dx dy >= 0.*

Now we describe the mathematical derivation of the non-linear SVM. In fact to find the decision function, we have to solve the optimization problem expressed below. It is obvious that using the kernel method, we replace every dot product (x_i, x_j) in terms of the kernel k in Equation 2.7. Thus, we obtain the more general form of this equation :

$$Ld = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{2.15}$$

subject to

$$\begin{cases} \alpha_i >= 0, i = 1, 2, ...n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

And the decision function is expressed by Equation 2.16:

$$f(x) = sign(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b) \tag{2.16}$$

With a suitable kernel, SVM can separate in the feature space data that are non separable in the original space [47]. Figure 2.10 shows examples of kernel representations.

Note that in the non linearly case, we can introduce the *slack variables* as described in Section 2.3.2 and the classifier is called a *soft margin classifier* [65].

The following example illustrates the non linear case:

Example 2.2. *Let us consider 5 data points having only one dimension:* x_1 *=1,* x_2*=2,* x_3*=4,* x_4*=5 and* x_5*=6 with* x_1, x_2, x_5 *belong to the positive class*$\{+1\}$ *and* x_3, x_4 *belong to the negative*

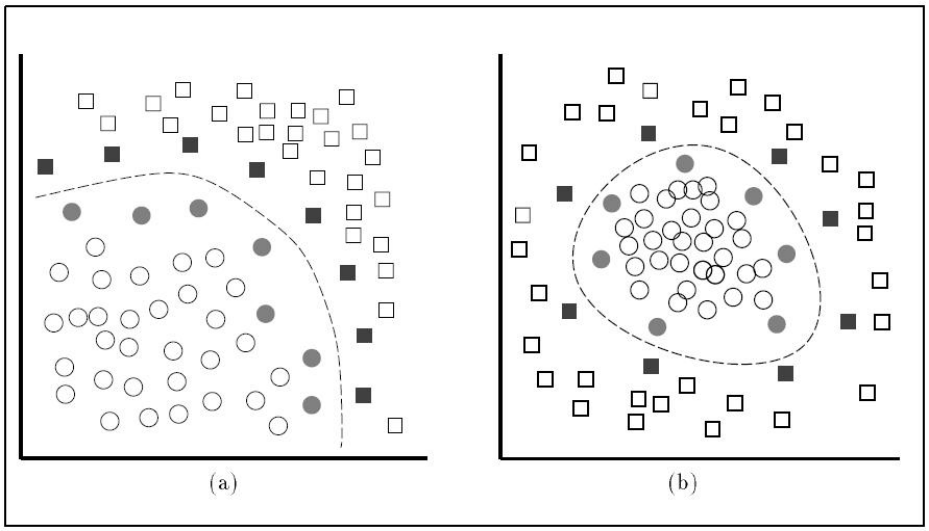


Figure 2.10: Decision surfaces given in (a) by a polynomial classifier, and in (b) by an RBF [47].

one $\{-1\}$. *Therefore we have the following points:* (x_i, y_i)=$(1,1),(2,1),(4,-1),(5,-1),(6,1)$.

Our purpose is to find the hyperplane separating these points. It is obvious that data is not linearly separable. So, we have to choose a kernel function and a value for the penalty parameter.

Suppose that the chosen kernel is the polynomial one of degree 2, that means that $K(x,y)$ *will be equal to* $(xy+1)^2$ *and the penalty parameter will be fixed to* 100. *First, we have to find the* α_i *values by solving the optimization problem of Equation 2.15:*

$max \sum_{i=1}^{5} \alpha_i$ - $\frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j +1)^2$

subject to $0 <= \alpha_i <= 100$ *and* $\sum_{i=1}^{5} \alpha_i y_i = 0$

Using a quadratic problem solver, we get: $\alpha_1 = 0$, $\alpha_2 = 2.5$, $\alpha_3 = 0$, $\alpha_4 = 7.333$ *and* $\alpha_5 = 4.833$. *We can at this moment specify support vectors which are* x_2, x_4 *and* x_5 *since their* α_i *are not null.*

Now, we can compute the decision function using Equation 2.16 as: $f(y) = 0.6667x^2 -$

$5.333x + b$ *and the value of* b *will be found by solving* $f(2) = 1$ *or* $f(5) = -1$ *or* $f(6) = 1$. *Finally,* $f(y)$ *will be equal to* $0.6667x^2 - 5.333x + 9$.

It is clear that this decision function is represented such a parabola as shown in Figure 2.11.

Once the decision function is determined, we can classify new data points. For example the class of the data point x_i*=3 is* $\{-1\}$ *since* $f(3)$ *is equal to -0.9987.*

Also, we can interpret visually using the parabola that the class of this data point class is effectively $\{-1\}$*, which confirms the computed result.*

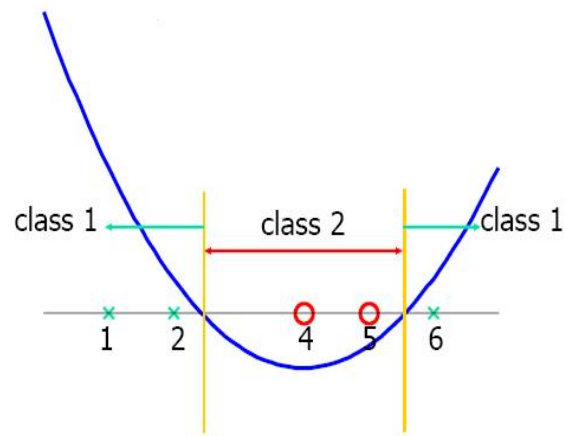


Figure 2.11: Decision surface given by a polynomial classifier.

2.4 Multi-class SVM

The standard form of SVM was originally designed for two-class problems. However, in real life situations it is often necessary to separate more than two classes at the same time. In this Section, we explore how SVM can be extended from binary problems to multi classification problems with k classes where $k > 2$. We will discuss two approaches, namely the *one-against-one* approach and the *one-against-all* approach.

In fact, most approaches for multi-class SVM decompose the data set to several binary problems. For example, the "one-against-one" approach trains a binary SVM for each two classes of data and constructs a decision function. Thus, for a k-class problem, there are

$k(k-1)/2$ decision functions [65]. Suppose that $k = 10$, we need to train 45 binary classifiers. This suggests large training times.

In the classification stage, a voting strategy is used where the testing point is designated to be in a class having the maximum number of votes. The voting approach is called the *"Max Wins"* strategy.

In order to clarify this approach, we propose an example:

Example 2.3. *Suppose that we have five records:* x_1, x_2, x_3, x_4, x_5 *such that* $x_1, x_2 \in C_1$, $x_3 \in C_2$ *and* $x_3, x_4 \in C_3$ *where* C_i *is the class assigned to each data point.*

First, we have to construct 3 decisions functions for each pairwise of classes. In other words, we have these decision functions:

$$\begin{cases} D_1(X) \text{ for the case } C_1 \text{ vs } C_2 \\ D_2(X) \text{ for the case } C_1 \text{ vs } C_3 \\ D_3(X) \text{ for the case } C_2 \text{ vs } C_3 \end{cases}$$

For each decision function, only concerned records are used. For example, for the case C_1 *vs* $C2$*, we suppose that records of the class* C_1 *are positive, so their output will be equal to* $+1$*, and the contrary for the class* C_2*. The same process is applied to* C_1 *vs* C_3 *and* C_2 *vs* C_3*.*

Second, to classify a new data point x_6*, we compute the decisions functions values for this data point.*

Suppose that we found a positive value for $D_1(X)$*. Hence,* $D_1(X)$ *suppose that the class of* x_6 *is* C_1*. But* $D_2(X)$ *and* $D_3(X)$ *judge that* x_6 *is assigned to* C_3*.*

Here, the "Max Wins" strategy is applied and x_6 *is affected to* C_3*.*

Another major method is called the "one-against-all" method. This technique requires the construction of k binary classifiers where the label $+1$ is designated to each class in its turn and the label -1 is designated to the other $k-1$ classes. In the classification stage, each data point is classified to the class having the largest decision value [65].

To explain this approach, we propose an example.

Example 2.4. *Suppose that we have the same data points of the example above.*

This approach is different. The first step consists on constructing 3 decisions functions for C_1 *vs* $\neg$ C_1, C_2 *vs* $\neg$ C_2 *and* C_3 *vs* $\neg$ C_3. *For each decision function,* C_i *is the positive class and* $\neg C_i$ *is the negative one where* $i = 1, 2, 3$.

In other words,

$$\begin{cases} \textit{SVM 1 learns } C_1 \textit{ vs } \neg\ C_1 \\ \textit{SVM 2 learns } C_2 \textit{ vs } \neg\ C_2 \\ \textit{SVM 3 learns } C_3 \textit{ vs } \neg\ C_3 \end{cases}$$

Then to predict the output for a new input, in the one hand we have to predict it with each SVM (SVM 1, SVM 2 and SVM 3). In the other hand, we have to find out which decision function that puts the prediction the furthest or into the positive region.

It was experimentally shown that for general problems various multi-classification approaches give similar accuracy [65].

2.5 Conclusion

SVM has received significant interest over the last years. It has been applied to various and complex problems with great success. This Chapter gives the basics on the SVM technique for both binary and multi classification problems. In the binary problem, we have presented the linear and separable case, the linear and non separable case and the non linear SVM. In multi classification problems, the *one-against-one* and the *one-against-all* approaches have been presented.

Like any method, SVM comes with certain advantages and disadvantages. In fact, a very good aspect of SVM is that only a small training set is needed to construct the decision function, because only *support vectors* are of importance during training. This point represents the SVM's fundamental and lets this method effective for high dimensional data. Despite that, SVM is computationally expensive in the one hand, it requires large memory and training might be slow. In the other hand, the extension to multi-class problems might not be optimum since SVM is originally designed for binary problems. Besides, users should choose the best parameters because an SVM is largely characterized by the choice of its parameters (kernel, penalty parameter) [71].

Our objective is to apply SVM to the segmentation problem where MR image is considered as input and tumor and normal classes are considered as outputs. Details of this process will be presented in Chapter 5.

Nevertheless any medical image handling needs some pre and post-processing treatments. Next Chapter is dedicated to these tasks.

Chapter 3

Pre-processing and post-processing phases

3.1 Introduction

Image pre-processing is important for real-life data which are often noisy and inconsistent. The purpose of this step is to improve the quality of the image by transforming it into another image that is better suited for machine analysis. This step represents a crucial phase in imaging processing applications because the effectiveness of subsequent tasks (e.g., features extraction, segmentation) depends highly on images quality. Also, it significantly improves the effectiveness of data mining techniques [60].

Moreover in imaging applications especially in segmentation systems, a post-processing phase is imposed because segmentation still remains a challenging problem and there is no solution until now [5]. Due to this fact, segmentation systems yield errors that should be reduced using post-processing operations.

In this Chapter, we consider several techniques that have proved their powerful in pre and post-processing images.

3.2 Pre-processing phase

This phase is composed of two parts i.e. image registration and noise reduction, detailed in what follows:

3.2.1 Image registration

Image registration consists in determining a one-to-one mapping of points in two different spaces. This mapping is also called *transformation*. More precisely, registration represents the process of spatially aligning two images or volumes by computing a transformation applied to an input image in order to match it to a template image that is assumed stationary.

Several registration methods exist in the literature. Their classification is based on different criteria such as *image dimensionality (2D or 3D)*, *nature of transformation (rigid, affine or curved)*, *optimization procedure*, *modalities* involved and *subject* criteria [19].

In this work, we will focus on the two last ones since we are interested to the criteria related to images and not to the image dimensionality, optimization and nature of transformation aspects. Within the existing methods handling these two criteria, we can distinguish methods registering intra-subject images, i.e. when all images are acquired of a single patient. Such methods are referred to by *intra-subject registration*. If the registration is accomplished using two images of different patients or a patient and a template, this is referred to by *inter-subject registration*.

In our work, intra-subject intra-modality is meaningless. Moreover, inter-subject inter-modality will not be treated due to the fact that we will only focus on studying one imaging technique, i.e. MRI studies while there are several imaging techniques such as X-rays scanner, tomography, scintigraphy, etc. Thus the two categories that will be detailed are:

- intra-subject inter-modality methods also called *coregistration*,
- inter-subject intra-modality methods known as the *template registration*.

These two methods share some basic operations. Thus, we will first start by introducing them.

Basic operations for registration

This part gives a brief introduction on the three basic operations of the registration task, namely transformations, spatial interpolation and optimization.

A) Transformations: Different transformations can be applied in the registration task such as the affine, rigid and curved transformations. The mentioned transformations will be detailed as follows:

- **Affine transformation**: is a transformation that includes *translations*, *rotations*, *scaling* and *shearing* (see Figure 3.1). This kind of transformation maps straight lines to straight lines and preserves the parallelism between them, but not their lengths or their angles. It has twelve degrees of freedom, i.e. parameters since the image has 3 dimensions (X, Y, Z) and this consists of one parameter for each of the three dimensions with respect to translation, rotation, scaling, and shearing [2].

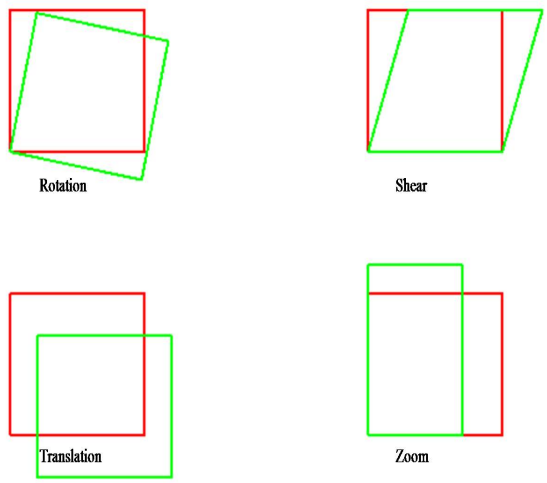


Figure 3.1: Different transformations applied to images.

1. Translation: Let (x_1, y_1, z_1) be a three dimensional point. Let (q_1, q_2, q_3) be a three dimensional translation vector. A translation from (x_1, y_1, z_1) to (x_2, y_2, z_2) via the translation vector is expressed by:

$$\begin{pmatrix} \mathrm{x}_2 \\ \mathrm{y}_2 \\ \mathrm{z}_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & q_1 \\ 0 & 1 & 0 & q_2 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{x}_1 \\ \mathrm{y}_1 \\ \mathrm{z}_1 \\ 1 \end{pmatrix} \tag{3.1}$$

2. Rotation: In two dimensions, a rotation is described by a single angle. Consider the point $(x1, y1)$ on a two dimensional plane. A rotation of this point to new coordinates $(x2, y2)$ by θ radians around the origin, can be generated by the following transformation:

$$x_2 = \cos(\theta)x_1 + \sin(\theta)y_1 \tag{3.2}$$

$$y_2 = \sin(\theta)x_1 + \cos(\theta)y_1 \tag{3.3}$$

Figure 3.2 shows an example of a 2D rotation where Φ is the angle between the X axis and $(x1, y1)$ and θ is the rotation angle between the points $(x1, y1)$ and $(x2, y2)$.

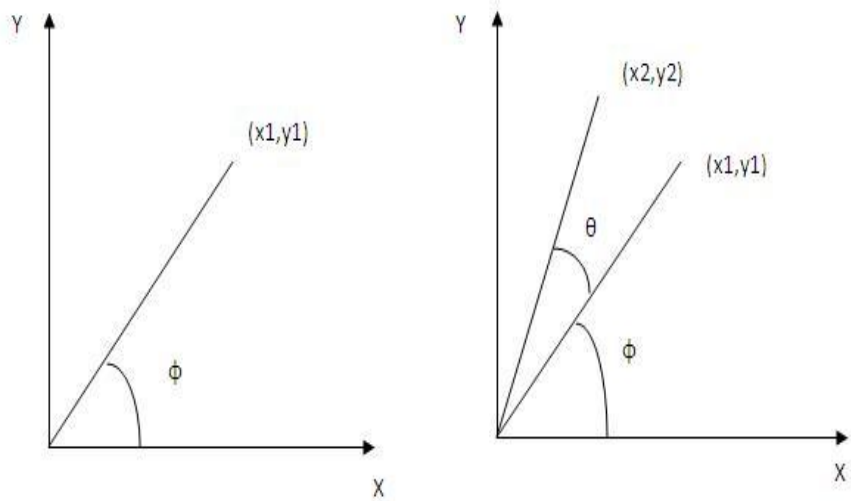


Figure 3.2: Illustration of 2D rotation.

For the three dimensional case, an object can be rotated in three different orthogonal planes, which are around the axes. For instance, a rotation of $\theta 1$ radians about the first axis (X) is called *pitch*, and is performed by:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) & 0 \\ 0 & -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \tag{3.4}$$

Similarly, rotations about the second and third axis ((Y) and (Z)) (called *roll* and *yaw*, respectively) are carried out by the following matrices:

$$\begin{pmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

and

$$\begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

In order to obtain the appropriate rotation, these matrices are multiplied together with respect to a specific order. In fact, the order is important in such transformations. In instance, a rotation of $\frac{\pi}{2}$ radians around the first and the second axis respectively produce a very different result if we reverse operations [2].

3. Zooms (Scaling): The transformations described above will generate purely *rigid* mappings. Zooms transformations are used to change the size of an image, or to work with images having *anisotropic*[1] or different voxel sizes between images.

These transformations can be expressed as follows:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \tag{3.7}$$

4. Shears: Shearing by parameters q_1, q_2 and q_3 can be performed via the following matrix:

$$\begin{pmatrix} 1 & q_1 & q_2 & 0 \\ 0 & 1 & q_3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.8}$$

- **Rigid transformation**: is considered as a special case of affine transformation restricted to *translations* and *rotations* (see Figure 3.1) [2].

  This means that it has six degrees of freedom since each degree of freedom consists of one parameter for each of the three dimensions with respect to translation and rotation.

- **Curved transformation**: allows the transformation of straight lines to curves [2].

[1] Anisotropic voxels do not have the same thickness in all dimensions while isotropic voxels have the same thickness along each dimension.

B) Spatial interpolation: Since the computed spatial transformation can change the size and location of pixels in the transformed image such that they do not correspond to the sizes and locations of pixels presented in the original image, the final stage that is called spatial interpolation or re-slicing is required.

Spatial interpolation uses the locations and intensities of the old pixels after spatial transformation to compute the intensity values at the new pixel locations.

Several methods of interpolation exist in the literature, we cite *nearest neighbour* [8], *trilinear method* [8], *polynomial method* [8], *Windowed Sinc Interpolation* [8] and *β-spline method* [29].

The choice of an effective interpolation algorithm is important, since some methods introduce more interpolation artefacts into the image than others. Evaluations of these methods had demonstrated that β-spline kernels are in general the most appropriate interpolator [44]. For this reason, they will be used in our framework.

In fact, β-spline interpolation methods model an image as a linear combination of *basis functions* as shown in Figure 3.3 such that the X and Y axis represent, respectively the locations and intensities of pixels before transformations and the solid line represents the interpolated function [29].

Given an image represented as a linear combination of such functions, the intensity value of the image at any location can be determined.

C) Optimization: Image registration is essentially an optimization. In fact, one image is spatially transformed in order to match another one (the template image), by minimizing or maximizing some criteria.

Formally, the optimization problem is expressed by Equation 3.9 where A is the template image, B is the image to be registered, T is the list of transformations, C is the measure of dissimilarity and $B\ o\ T$ represents the application of T to B.

$$\min_T C(A, B\ o\ T) \tag{3.9}$$

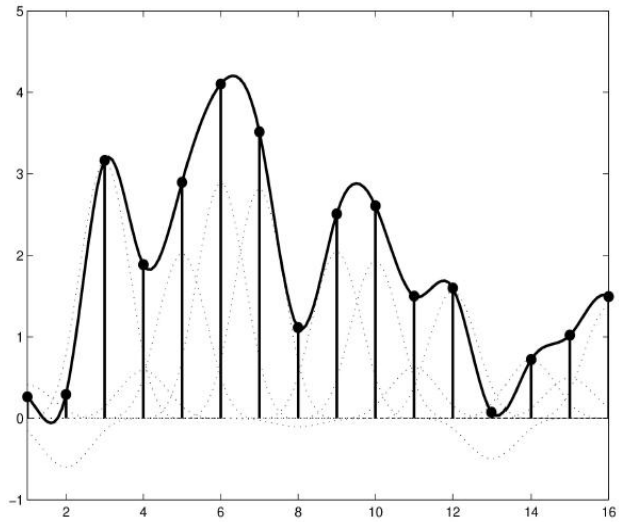


Figure 3.3: Illustration of β-spline representation of an image [29].

Finding the minimum of dissimilarity measure or the maximum of similarity measure is considered as a multidimensional optimization problem, where the degrees of freedom of the expected transformation correspond to the number of dimensions [79].

This optimization problem should be solved to obtain the transformation parameters that provide the optimum value for the similarity or dissimilarity measure. To insure this task, optimization algorithms should be fast and robust in order to provide solutions with a minimum number of local maxima which are not optimal values.

The local maxima can have two different causes. Some represent a local good match of two images. Others are imperfections related to the used interpolation method. Because of the existence of local maxima, the choice of optimization algorithm has a large influence on the registration results [40].

Within popular techniques we can mention : Powell's method [52], Gauss Newton method (see [56]), stochastic search methods, genetic methods [24], etc. More details about these techniques are given in [41] and [56]. We will not detail them since it is not the topic of this work. But we will mention some measures used in the optimization problems.

In fact, in the literature, we can distinguish several measures used to assess an optimization problem, the most used in the context of medical imaging are: the *entropy correlation coefficient*, the *normalized cross correlation*, the *normalized mutual information*, the *sum of squared difference*, etc.

The choice depends on the nature of registration. For inter-modal registration, the normalized mutual information and the entropy correlation coefficient are used and for within-modality registration, the sum of squared difference should be used [17].

In what follows, we give some details regarding the major registration measures, namely entropy measure, mutual information, normalized mutual information and sum of squared difference.

1. Entropy measure: The entropy measure introduced by Shannon [64] in 1948 has been used to measure the amount of information of an event. It is applied in different domains such as supervised classification [57], clustering [38], image registration [66], etc.

In fact, to say that an information is more informative than another, we have to measure the amount of information that they brought. In our case, the Shannon entropy can be computed for an image since its information can be represented by a probability distribution of grey values. This latter is computed by counting the number of occurrences of each image grey value and dividing them by the total number of occurrences.

The entropy values can be high or low depending on the image content. More precisely, a low entropy value corresponds to an image containing very little information, i.e. few intensities values. While an image that contains a lot of information (different intensities values) has a high entropy value. In this manner, the Shannon entropy can be considered as a measure of dispersion of a probability distribution. In fact, a distribution having a single sharp peak corresponds to a low entropy value, whereas a dispersed distribution corresponds to a high entropy value [51].

Formally, the entropy of an image A is expressed as follows:

$$H(A) = -\int p_A(I_A) \log p_A(I_A) da \tag{3.10}$$

where I_A denotes the intensity values of A, $p_A(I_A)$ denotes the probability of appearance of the intensity (I_A) in A and the term $\log p_A(I_A)$ signifies the amount of information gained from an event with a probability p_i.

In the context of registration, Studholme et al. [66] suggested to use the entropy as a measure of registration. By finding the transformation that minimizes the *joint entropy*, images should be registered. Formally, the joint entropy between two images A and B can be expressed as:

$$H(A,B) = -\int p_{A,B}(I_A, I_B) \log p_{A,B}(I_A, I_B) da \tag{3.11}$$

where (I_A, I_B) represents a pair of intensities values in A and B and $p_{A,B}(I_A, I_B)$ denotes the probability of appearance of I_A in A and I_B in B.

In fact, the *joint entropy* can be visualized using a *joint histogram* which is a two-dimensional plot showing grey values combinations for all corresponding pixels in each of the two images. It is constructed by counting the number of times a combination of grey values occurs.

The joint histogram can be visualized in Figure 3.4 which contains several histograms of an MR image with it self for different rotations. In fact, it is build as follows: each axis represents image intensities and a grey level is attributed for each point (a, b) regarding the number of pixels showing intensity a on image A and intensity b on image B.

In the case shown here (Figure 3.4), the white color indicates the pairs of intensities that correspond to none pixel, and the darkest level corresponds to the big number of pixels having the corresponding intensities values.

More explicitly, there is a pixels coincidence if the images are aligned, whereas the number of coincidences decreases and the histogram gets dispersed if images are misaligned. Thus, the joint histogram changes as the alignment of images changes.

When images are correctly registered, certain anatomical structures overlap and grey levels of these structures will be represented in the joint histogram as clusters if many pixels are in coincidence. Also, when images become misaligned, there is an overlapping between structures such as overlapped structures do not represent their anatomical counterparts in the second

image.

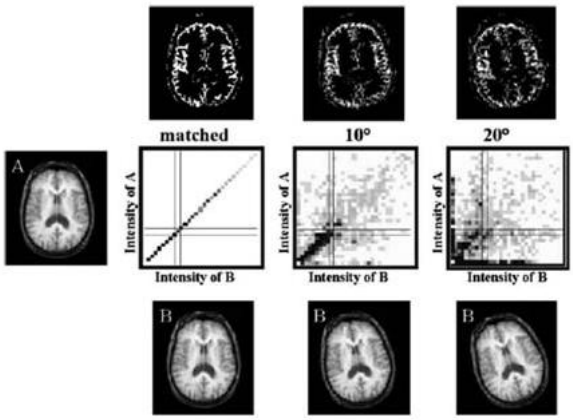


Figure 3.4: Illustration of the joint histogram [51].

Figure 3.5 contains several joint histograms of an MR image with itself for different rotations and their joint entropy values. When images are identical, all grey value correspondences lie on the diagonal as shown in the left histogram. Whereas joint histograms are dispersed if images are misaligned, as shown in the three following joint histograms. Each histogram represents an MR image rotated to the other by 2, 5 and 10 degrees, respectively. Note that the joint entropy measure need to be reduced in order to have a good registration of images.

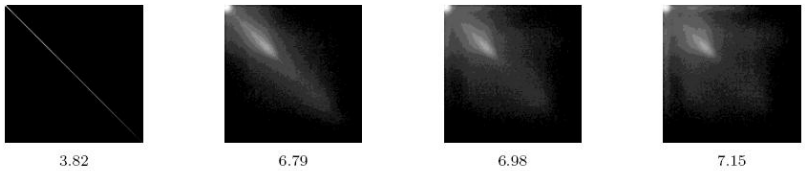


Figure 3.5: Joint grey value histograms of an MR image with itself and their entropy values [51].

2. Mutual information: Based on the entropy measure, the mutual information has been used as a registration measure in the early of 1990 [76], [77].

Collignon et al. [17] and Viola and Wells [31] [74] applied the mutual information to rigid registration. It showed great promise and it became the most investigated measure for medical image registration [51]. Formally, the mutual information of two images A and B is formulated as follows:

$$MI(A,B) = H(A) + H(B) - H(A,B) \tag{3.12}$$

where

- $H(A) = -\int p_A(I_A) \log p_A(I_A) da$
- $H(B) = -\int p_B(I_B) \log p_B(I_B) db$
- $H(A,B) = -\int p_{A,B}(I_A, I_B) \log p_{A,B}(I_A, I_B) da$

Equation 3.12 contains the term $-H(A,B)$, which means that maximizing mutual information is related to minimizing joint entropy. The advantage of mutual information over joint entropy is that it adds the entropies of the separate images. Mutual information will have low values when the overlapping part of the images represent only the background and high values when it contains anatomical structures, contrary to the joint entropy [51].

One drawback of the mutual information measure is that in special cases, it can decrease when misalignment increases i.e. images only partially overlap [66].

3. Normalized mutual information (NMI): In order to address the problem of the mutual information measure, the normalized mutual information was proposed that is less sensitive to changes in overlap.

$$\text{NMI(A,B)} = \frac{H(A)+H(B)}{H(A,B)} \tag{3.13}$$

This measure offers improved results over mutual information measure in registration tasks [62].

4. Sum of squared difference (SSD): seeks to minimize the sum of squared difference intensities for the image source and the image template. It is formally expressed by the equation below where I_x is the intensity function of the image x and i denotes a pixel:

$$\text{SSD(A,B)} = \sum_i (I_A(i) - I_{B\,o\,T}(i)) \tag{3.14}$$

3.3.1.1 Coregistration

Coregistration is the task of aligning different modalities of the same patient. This step is essential if the used modalities are not in perfect alignment which is often the case with real data [62]. It uses a rigid-body model that is parameterized by three translations and three rotations about the different axes (i.e. X,Y and Z).

This model is used since the same subject is present in the source image (input image) and the template image which show small changes in object shape or small changes in object intensities. Thus an exact correspondence should exist [17].

Various registration options should be defined, then passed to the optimization problem, such as the objective function since registration involves finding parameters that either maximize or minimize some objective function.

Currently, one of the most popular methods of coregistration in medical imaging is the maximization of the normalized mutual information (NMI) measure. For this reason, we will use it in our framework. The optimization method applied in this step will be the Powell method [52].

Figure 3.6 shows a coregistration example that maximizes NMI. Note that the T1-weighted image after contrast injection is used as the target modality.

3.3.1.2 Template registration

In the medical field, especially in the neurological area, template images are very popular. They are used to enable registering images from different subjects into roughly the same coordinate system. In fact the reason behind the existence of templates in this area is the fact that images are very different in the one hand. In the other hand, it is not possible to register images of different subjects to each other since template image should be perfect, i.e. has a minimum number of artefacts and the choice of such template from the list of images being registered will be difficult.

For these reasons, after aligning images in different modalities, i.e. coregistration, the second registration step aligns the modalities with a template image in a standard coordinate system in order to average signals from brain images of different subjects. This is often done by mapping all images into a recognized brain coordinate system.

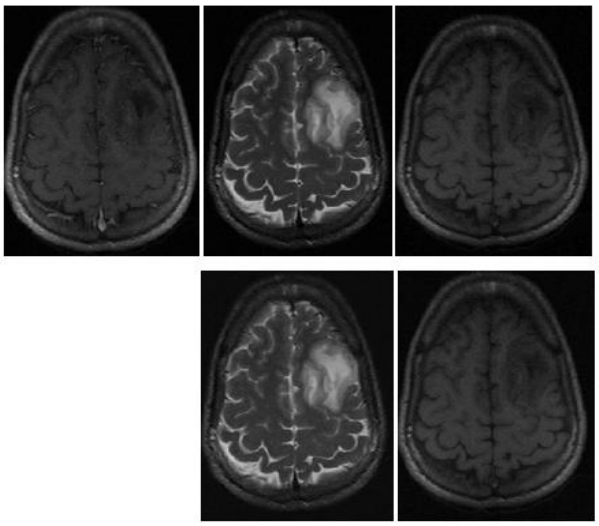


Figure 3.6: Coregistration by maximization of NMI measure. Top: left to right: Template image (T1 after injection of contrast agent), T2 image, T1 image. Bottom: left to right: T2 and T1 images after coregistration.

In the literature, we can distinguish two coordinate systems, the *Talairach system* [68] and the *MNI*[2] coordinate system inspired from the Talairach system proposed in 1988 [20]. The second coordinate system handles intensity templates in different modalities i.e. T1 and T2-weighted, that is why we will use it in the rest of this work. Figure 3.7 and Figure 3.8 show, respectively the Talairach and the axial slice of MNI for 152 T1-weighted volumes.

Template registration using MNI system requires that the template used should have the same modality as the image to be registered. Templates in the MNI coordinate system include 'MNI305'[3], the average of 305 T1-weighted normal individuals [20], and the more recent and higher quality, the 'ICBM[4]152'[5]. In fact, from a dataset of 152 normal individuals, three templates are derived, we are interested to the T1 and T2-weighted average images [42].

[2]Montreal Neurological Institute and Hospital, an institute at McGill University studying the nervous system.

[3]A data set of 305 spatially registered normal individuals used in the construction of templates and prior probabilities.

[4]International Consortium for Brain Mapping: a project whose goal is the continuing development of a probabilistic reference system for the human brain. http://www.loni.ucla.edu/ICBM/

[5]A data set of 152 spatially registered normal individuals used in the construction of templates and prior probabilities

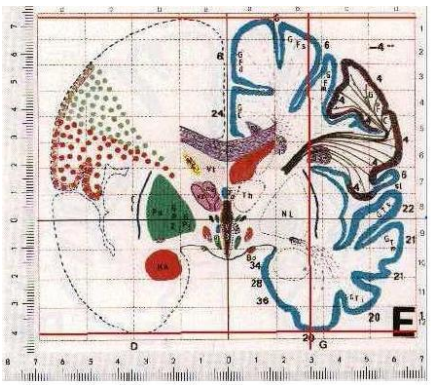


Figure 3.7: The Talaraich template [62].

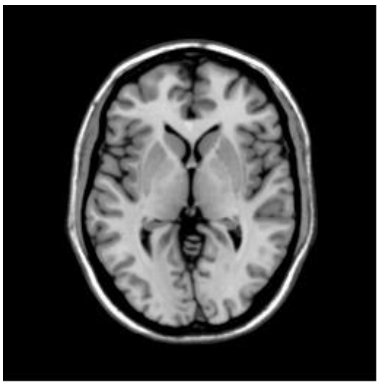


Figure 3.8: The T1-weighted MNI template [62].

In the template registration process, only a single modality (T1 or T2) needs to be registered with the template since the modalities are already aligned with each other in the coregistration step. The transformation computed to register a modality with a template can then be used to transform the other modalities.

All template registration or spatial normalization methods start with determining the optimal 12 parameters of the affine transformation. This step is performed by minimizing or maximizing some function of the image.

The template registration step can be divided into two categories: the linear registration and the non linear registration (i.e. transforms lines to curves) as shown in Figure 3.9 where the source image (a) is rotated (see image (c)), then scaled (image (d)) to match it to the template image (image(b)). The output of these transformations is represented in image (d) that is the input for the non linear registration. Image (e) illustrates the non linear mapping from the image (d) to the image (e) based on the template image (b). In fact, a deformation is applied to image (d) to match it exactly to the target (b). In our work we are interested to the simplest case, the linear one.

In fact, linear template registration is the registration of images using the rigid or the affine transformation. The choice of transformation depends on the nature of problem, i.e. the kind of registration.

Within methods associated with this topic, we will use the method outlined in [23] which

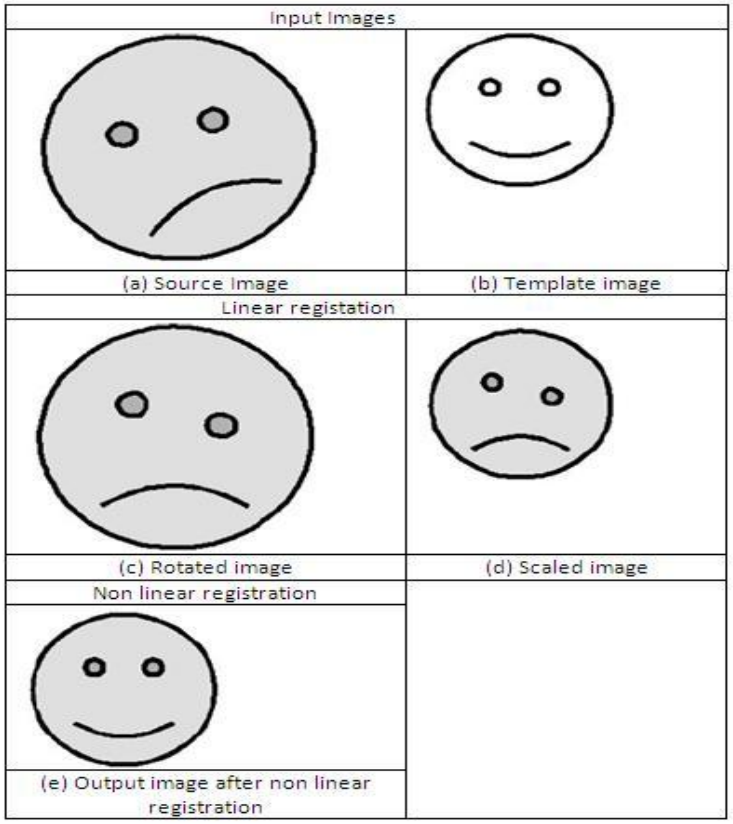


Figure 3.9: Illustration of linear and non linear registration.

is used in the popular registration tool, namely SPM[6]. This method works by minimizing the *sum of squared difference* between the image to be normalized and the template image. It computes a linear 12-parameters affine transformation. Unlike the coregistration step where the images to be matched together are from the same subject, in template registration zooms and shears are needed to register heads of different shapes and sizes.

Also, an optimization problem should be solved to find the optimum parameters that match the image to be normalized to the template image. The method used to efficiently find these parameters is the Gauss-Newton optimization .

The main contribution of [4] was the introduction of a prior knowledge of the variability

[6]Statistical Parametric Mapping, a collection of Matlab scripts developed at the University College of London for the purpose of statistical analysis of fMRI scans. This software is available at `http://www.fil.ion.bpmf.ac.uk/spm/`.

of head sizes into the registration task, especially into the optimization problem. So the algorithm simultaneously minimizes the sum of squared difference between the template and the source image in the one hand. In the other hand, it minimizes the squared distance between the parameters and the prior expected parameters. This approach is essentially a *maximum a posteriori (MAP) Bayesian* approach.

The process is as follows: when the error for any parameter is large, then that parameter will be based on the prior information. In order to adopt this approach, the prior expected parameters can be derived from the zooms and shears determined by registering a large number of brain images to the template [3].

3.2.2 Noise reduction

Image noise is an image alteration that can be caused by acquisition or storage processes. This alteration consists on a variation in image brightness information. It can be reduced due to filters. In fact *filtering* or *smoothing* is a neighborhood operation where the pixel value is replaced by a function value applied to this pixel and a set of its neighbours [25].

There are several methods used for reducing noise. In fact, linear filtering techniques apply *low-pass filters* which reduce the noise in the one hand and smooth coutours in the other hand. This yields a fuzzy and unclear image. While *Anisotropic diffusion* techniques, introduced by Perona and Malik [48] are considered very effective for image smoothing since they preserve significant discontinuities and produce noteworthy results [5]. Also, many studies [14] [48] have demonstrated that diffusion filters can be used to enhance and detect object edges within images. So we will adopt them in our work.

The idea behind anisotropic diffusion filtering (ADF) is to consider the intensity of the image as a concentration of fluid, which evolves to attend stability. It is based on using the classic heat diffusion equation. In fact, *diffusion* is a physical phenomenon that balances the concentration differences without creating or destroying masses. In the image context, concentration denotes intensity and preserving the mass corresponds to preserving image contours (edges).

Filters based on the anisotropic diffusion principle encourage intra-region smoothing while inhibiting inter-region smoothing due to the existence of an important discontinuity, relative to a region boundary or a significant contour. Therefore, such filters strengthen the difference

between regions and eliminate the noise by increasing the region homogeneity and preserving the edges.

Formally the diffusion process as Perona and Malik is expressed by the following Equation:

$$\begin{cases} \frac{\partial I(x,t)}{\partial t} = div[c(\nabla I(x,t)).\nabla I(x,t)] \\ I(t=0) = \mathrm{I}_0 \end{cases}$$

where $x = (u, v)$ represents the spatial coordinates, t refers to the iteration step, $I(x, t)$ corresponds to image intensities, div is a divergence operator, $c(\nabla I(x, t))$ is called *diffusion function* and it is a decreasing function of the image gradient magnitude $\nabla I(x, t)$. In fact, c is the operator that allows us to control the influence of diffusion and it is chosen to allow diffusion in homogeneous regions, but not through contours. If the gradient is increasing the diffusion is stopped.

The diffusion function can be expressed by Equation 3.15:

$$c(\nabla I(x,t)) = exp[\frac{-|\nabla I(x,t)|^2}{K}] \tag{3.15}$$

where K is referred to as the diffusion constant or the *flow constant*, and the behavior of the filter depends on K. The choice of its value depends on the level of noise of the image [48].

Figure 3.10 shows the ADF result on a brain tumor image.

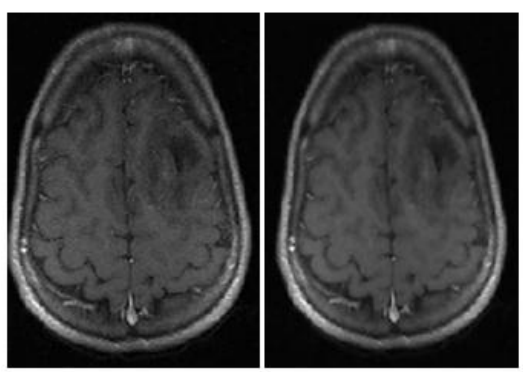


Figure 3.10: Noise reduction results on MRI data. Left to Right: Image before filtering, Image after filtering.

3.3 Post-processing phase

Mathematical morphology is a theory used especially in image analysis, whose aim is the study of objects according to their shape, size, relationships with their neighbours, their texture, and their grey scale. This theory proposes several transformations which can be at different levels of image processing (filtering, segmentation, texture analysis, etc.) [7].

In our work, we will use this theory as a post-segmentation operation in order to clean segmentation results and fill up some cavities. Morphologic operations are especially suited to binary images processing which is the case of our study.

Within the common morphological operations, we cite: *dilation* and *erosion*. In fact, dilation increases the size of small regions by adding pixels to objects boundaries, while erosion reduces the size of regions by removing pixels on objects boundaries. A primordial part of these morphological operations is the *structuring element* (SE). It consists on a matrix of 0's and 1's. The center pixel is called *origin* . It represents the pixel of interest of the structuring element. All pixels containing 1's define the neighborhood of SE. The idea behind using such geometrical set (SE) is the following: indeed, the value of each pixel in the input image is compared to SE, and depending on the transformation (dilation or erosion), this pixel remains as it is or changes and becomes within background or object pixels. Figure 3.11 shows a diamond structuring element.

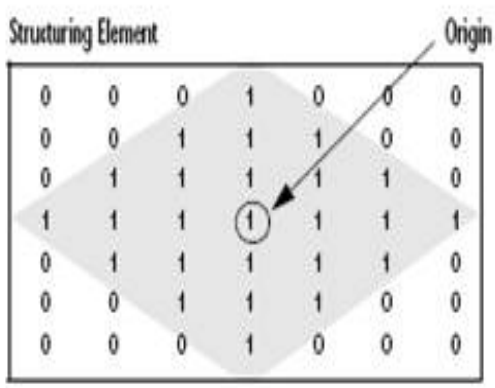


Figure 3.11: Example of a diamond structuring element.

We propose in what follows some details about dilation and erosion operations.

Dilation

Given two inputs: the binary image to be dilated composed of both background and object regions and the structuring element. Note that the result of dilation depends only on the structuring element.

The mathematical definition of dilation is the following: suppose that X is the set of points corresponding to the input image and K is the set of points of SE. Let K_x is the translation of K in position x. Then dilation of X by K is the set of points such that the intersection between K_x and X is not empty.

More precisely, for each background pixel of the input image, we superimpose it with the origin of the structuring element. Two cases can be identified:

- If at least one pixel in SE coincides with an object pixel then this pixel is set to be within object pixels.
- If all pixels of SE are background, so the input pixel remains background.

Figure 3.12 demonstrates the effect of dilation using a 3 × 3 square structuring element such as background is represented by 0 and objects are represented by 1.

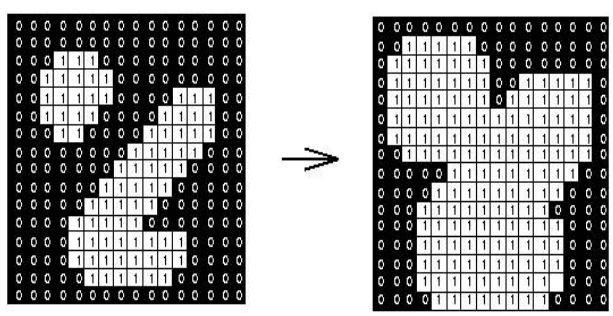


Figure 3.12: Illustration of the effect of dilation using a square SE.

Erosion

Erosion is the dual of dilation. Let X and K represent the input image and SE, respectively. The erosion of X by K is the set of points such that K_x is a subset of X. More precisely, for each object pixel, we superimpose it to the structuring element. Two cases can be presented:

- If every pixel of SE is an object pixel then the input pixel remains as it is (i.e. an object pixel).
- If any of SE pixels is background then the input pixel is set to background.

Figure 3.13 demonstrates the effect of erosion using a 3×3 square structuring element.

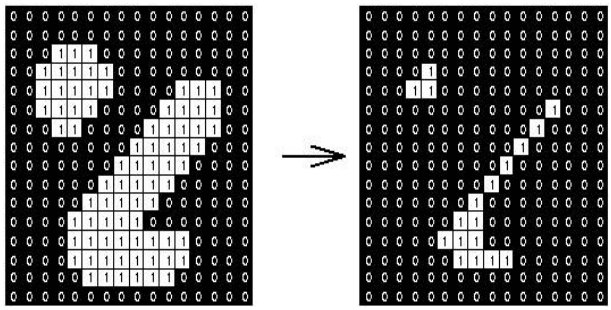


Figure 3.13: Illustration of the effect of erosion using a square SE.

Note that dilation and erosion operations change the objects shape by adding or removing boundaries pixels. Figure 3.14 shows this process before and after dilation and erosion using a circle SE.

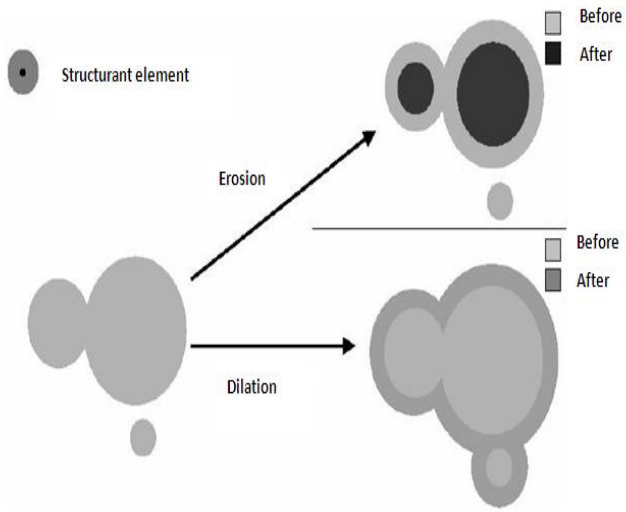


Figure 3.14: Illustration of dilation and erosion operations.

3.4 Conclusion

In this Chapter, we have presented both pre-processing and post-processing phases. In the first phase, two kinds of pre-processing steps have been investigated, namely: *image registration* and *noise reduction*. Image registration consists mainly on spatially aligning images of different modalities with each other. This is the *coregistration* step while *template registration* consists on registering co-registered images using a template image. In our work, the template that will be used is the MNI coordinate system [20]. After the registration step, images are denoised using *anisotropic diffusion filtering* in order to reduce the effect of eventual noise. In the second phase of this Chapter, we have proposed a brief introduction to morphological filters which have been used as a post-segmentation operation. These filters are based on *dilation* and *erosion* transformations which aim to change the shape of objects. This phase is essential in our work in order to clean our binary segmentation results.

Next Chapter proposes features extraction phase and presents the common features used to isolate tumor and normal regions within MR images.

Chapter 4

Features extraction phase

4.1 Introduction

Image features are considered distinguishing images attributes. They are usually defined by the visual appearance of an image, so they include the luminance of pixels and grey scale textural regions [25]. They are of major importance in the isolation of regions of common property within an image (image segmentation) and the labeling of such regions (image classification), which are the topics of our work. Effectively, since we handle the segmentation problem as a classification problem, features should be extracted and computed for each image pixel in order to discriminate between normal and abnormal regions. This Chapter describes several image features that have been proposed for medical image segmentation. We will focus our presentation on those relative to intensities and texture.

4.2 Intensities features

After the pre-processing phase, especially the image registration step (see Subsection 3.2.1), it is obvious to consider pixel intensities for each modality. As we have mentioned in Chapter 1, MRI offers different kinds of visualizations (i.e. T1-weighted, T1 after injection of contrast agent and T2-weighted images), so it is important to profit of these visualizations which provide useful information if they are used in conjunction with each other. So the idea is to consider the intensities features for T1, T1 after contrast and T2 modalities.

4.3 Texture features

Texture has been within the most important characteristics that classify and recognize objects. Image texture may be considered as something characterizing the intensity surface of an image. In fact, intensity can be measured at resolution of a single pixel, whereas texture takes into consideration the grey value of the neighbourhood surrounding the pixel. Multiple of definitions of texture have been proposed in the literature. We propose the definition of Faugeras and Pratt in [21]:

The basic pattern and repetition frequency of a texture sample could be perceptually invisible, although quantitatively present. In the deterministic formulation, texture is considered as a basic local pattern that is periodically or quasi-periodically repeated over some area.

A texture survey was done by Tuceryan and Jain [70] where they reviewed the basic concepts and various methods for texture processing. In fact, texture can be used in several areas such as texture classification (recognition of image regions), texture segmentation (finding texture boundaries), etc.

There exist a multiple of texture analysis methods which can be classified into statistical methods (*co-occurrence matrices, autocorrelation features*, etc.), geometrical methods (*Voronoi tessellation features*, *structural methods*, etc.), model-based methods (*random field methods*, *fractals*, etc.) and signal processing methods (*spatial domain filters*, *Fourier domain filtering*, *Gabor and wavelet models*, etc.) [1].

Each method characterizes texture in terms of the features it extracts from image. Therefore, the choice of any method depends on the goal behind using image texture.

Our study will focus on the statistical models, in fact one possibility to analyze image texture in the spatial domain is to compute statistical moments called *first-order texture features* and the *second-order texture features*. We first give some notations needed to the presentation of these features.

4.3.1 Notations

- m_{rc} denotes a pixel and (m_{rc1}, m_{rc2}) denotes a pair of pixels.
- $N(i)$ denotes the number of pixels having the grey level i in a specific window.

- M denotes the number of pixels of the window.
- $P(i)$ denotes the number of pixels having the same grey level i in the window divided by M, i.e. $P(i) = \frac{N(i)}{M}$.
- G denotes the maximum number of grey levels inside a window.
- d denotes the distance between a pair of pixels.
- θ denotes the angle between a pair of pixels.
- N_g denotes the number of possible grey levels for a particular image.

4.3.2 First order texture features

First order texture features are statistics calculated from the original image values and do not consider relationships between pixel neighbours. They are based on *grey level histograms* of the first order. From these histograms, parameters with statistics of different degrees can be extracted. In what follows we define the most popular ones [25]:

- ***Mean***: this feature represents the mean of grey levels values. Formally, it is expressed by:

$$\mu(m_{rc}) \ = \ \sum_{i=0}^{G-1} \ i \ P(i) \tag{4.1}$$

- ***Standard deviation***: it is a measure of the distribution of grey levels around the mean. It manifests region roughness. Formally, the standard deviation is expressed by:

$$\sigma(m_{rc}) \ =[\ \sum_{i=0}^{G-1} \ (i-\mu)^2 \ P(i) \ ]^{\frac{1}{2}} \tag{4.2}$$

- ***Skewness***: a measure of the dissimilarity of the grey level histogram (if it is well balanced around its mean or if it is more directed to the left or to the right with regard to its mean). It is a response for the question : are the intensities usually darker or lighter than the mean? This measure is defined by:

$$\mu_3(m_{rc}) \ = \sigma^{-3} \ \sum_{i=0}^{G-1} \ (i-\mu)^3 \ P(i) \tag{4.3}$$

- ***Kurtosis***: a measure that gives an indication on the sharp and disentangled aspect or rather flattened of histogram, i.e. we search how uniform is the grey level distribution. It is defined by:

$$\mu_4(m_{rc}) \ = \sigma^{-4} \ \sum_{i=0}^{G-1} \ (i-\mu)^4 \ P(i) \tag{4.4}$$

- ***Energy***: a feature that indicates how much intensity variation is there in the image. It is expressed by:

$$\mathrm{E}\ (\mathrm{m}_{rc}) \ = \ \sum_{i=0}^{G-1} \ P(i)^2 \tag{4.5}$$

- ***Entropy***: a measure that evaluates the complexity of the image, how normal or non normal is the grey level distribution. It is like standard deviation, a texture measure widely used to quantify the smoothness of image texture. The entropy is defined by:

$$\mathrm{H}\ (\mathrm{m}_{rc}) \ = \ - \ \sum_{i=0}^{G-1} \ P(i) \log \ P(i) \tag{4.6}$$

Example 4.1 illustrates the calculation of the entropy and the mean parameters.

Example 4.1. *Assume a simple digital image of size 3 × 3 such as its matrix of pixels is represented by Table 4.1.*

205	242	185
215	174	162
121	162	185

Table 4.1: Matrix of pixels.

To compute the entropy of this image, 3 steps should be applied:

1. *First, we should handle points of border which are pixels located in the extremity of the image matrix. There are several ways to treat these points, we cite zeros method, circular symmetry method and the symmetric method. In the last one, we attribute for border points the grey levels of their symmetries being in the image. In this example, we have applied the symmetric method. Table 4.2 shows the matrix of pixels after applying the symmetric method to the points of border.*
2. *Second, we have to choose the size of neighborhood, for example 3-by-3 or 9-by-9. In this example, we consider the 3-by-3 neighborhood. Hence, the window size will be 3 × 3.*
3. *Third, we compute the entropy value for each pixel in the original image in order to obtain the entropy matrix. In fact, each output pixel in the entropy matrix contains the entropy value of the neighborhood around the corresponding pixel in the input image.*

174	215	174	162	174
242	205	242	185	242
174	215	174	162	174
162	121	162	185	162
174	215	174	162	174

Table 4.2: Matrix of pixels after treating border points.

Hence, to compute the entropy value for the pixel having as grey level 205, we should compute P(i) for each pixel neighborhood. Indeed, we start with P(174), P(215), P(242) and P(205).

- *P(174)* $= \frac{N(174)}{M} = \frac{4}{9}$; *P(215)* $= \frac{N(215)}{M} = \frac{2}{9}$; *P(242)* $= \frac{N(242)}{M} = \frac{2}{9}$; *P(205)* $= \frac{N(205)}{M} = \frac{1}{9}$;
- *Applying Equation 4.6, H(205)= - [P(174)* log *P(174) + P(215)* log *P(215) + P(242)*log *P(242)+ P(205)* log *P(205)] = 0.*

The same steps are repeated for the remaining pixels and we obtain a matrix of zeros, i.e. a black entropy image.

The same principle is applied to compute any texture feature. In instance, to compute the mean texture, only the third step changes. In fact, by applying Equation 4.1 the mean of the grey value 205 will be:

$$\mu(205) = [\ 174\ \ P(174) + 215\ \ P(215) + 242\ \ P(242) + 205\ \ P(205)\ ] = 201.$$

As for the grey level 205, we will follow the same steps in order to obtain the mean texture matrix. The idea behind computing the mean value of the next pixel is to change the position of the window in pixels matrix such that the window size stills always 3×3.

- $\mu(242) = [215 \times \frac{2}{9} + 174 \times \frac{2}{9} + 162 \times \frac{2}{9} + 205 \times 19 + 185 \times \frac{1}{9} + 242 \times \frac{1}{9}] = 192.$
- $\mu(185) = [174 \times \frac{4}{9} + 162 \times \frac{2}{9} + 242 \times \frac{2}{9} + 185 \times \frac{1}{9}] = 187.$
- $\mu(215) = [242 \times \frac{2}{9} + 205 \times \frac{1}{9} + 174 \times \frac{2}{9} + 215 \times \frac{1}{9} + 162 \times \frac{2}{9} + 121 \times \frac{1}{9}] = 184.$
- $\mu(174) = [205 \times \frac{1}{9} + 242 \times \frac{1}{9} + 185 \times \frac{2}{9} + 162 \times \frac{2}{9} + 174 \times \frac{1}{9} + 215 \times \frac{1}{9} + 121 \times \frac{1}{9}] = 183.41.$
- $\mu(162) = [242 \times \frac{2}{9} + 185 \times \frac{2}{9} + 174 \times \frac{2}{9} + 162 \times \frac{3}{9}] = 187.54.$
- $\mu(162) = [215 \times \frac{2}{9} + 185 \times \frac{1}{9} + 174 \times \frac{2}{9} + 162 \times \frac{3}{9} + 121 \times \frac{1}{9}] = 174.42.$

- $\mu(121) = [174 \times \frac{4}{9} + 215 \times \frac{2}{9} + 162 \times \frac{2}{9} + 121 \times \frac{1}{9}] = 170.$
- $\mu(185) = [174 \times \frac{4}{9} + 162 \times \frac{4}{9} + 185 \times \frac{1}{9}] = 165.$

Hence, the mean texture matrix is as follows:

201	192	187
184	183.41	187.54
170	174.42	165

Table 4.3: The mean texture matrix.

Figure 4.1 shows examples of first-order (statistical moment) features for a brain tumor image issued from MRI.

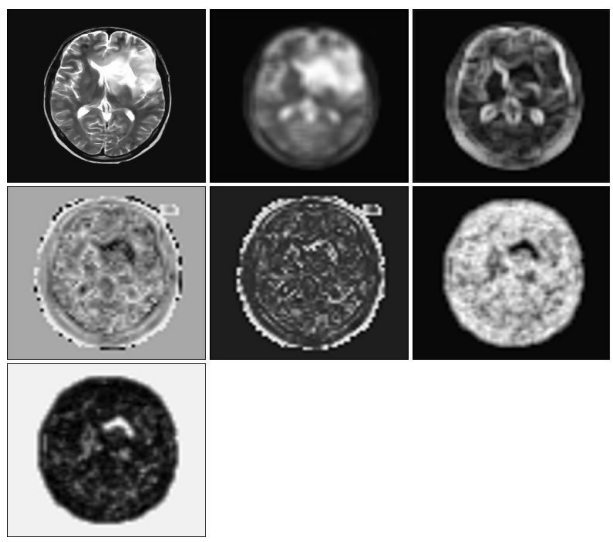


Figure 4.1: Examples of first-order features for a brain tumor image. Top, left to right: Original image, mean, standard deviation. Middle, left to right: Skewness, kurtosis, entropy. Bottom left: Energy.

4.3.3 Second order texture features

First order statistical methods allow analysis based on histograms description. Hence, information about pixel localization is not provided. To insure a precise analysis, second order methods are applied. In fact, these methods take into account the relationship between two pixels (usually neighbouring) in the image.

To account for second order texture features, one of the most well-known methods of texture analysis is *grey-level co-occurrence matrices (GLCM)* proposed by Haralick et al [28].

GLCM is a powerful technique for measuring texture features. It takes into account the spatial distribution of grey levels around a given pixel of the image. This matrix contains frequencies $P(i, j, d, \theta)$ such as i and j are two neighbouring pixels separated by displacement d and an angle θ. Note that GLCM can be considered as a two dimensional grey level histogram for a pair of pixels separated by a given distance [1].

The GLCM is expressed by Equation. 4.7:

$$\begin{aligned} \mathrm{P(i,\ j,\ d,}\ \theta) = \#(\mathrm{m}_{rc1}, m_{rc2}) \mid f(m_{rc1}) = i,\ f(m_{rc2}) = j,\ |(m_{rc1}) - (m_{rc2})| \ = d, \\ \angle((m_{rc1}), (m_{rc2})) = \theta \end{aligned} \tag{4.7}$$

where # denotes the number of occurrences inside the window where intensity of a pair of pixels changes from i to j given a distance d between (m_{rc1}, m_{rc2}) and an angle θ between (m_{rc1}, m_{rc2}), i.e. the orientation. In this Chapter, $P(i, j, d, \theta)$ will refer to $P(i, j)$ in order to simplify the representation of equations.

Different orientations can be used in the GLCM computation. The most used are the horizontal $P0$, vertical $P90$, left and right diagonals orientations ($P135$ and $P45$, respectively) . In fact, the horizontal orientation corresponds to an angle of 0° between the pair of pixels which represents a left-right transition of grey level i to grey level j. The vertical one is a 90°. However, the diagonally orientation that can be a right or a left orientation is 45° or 135°, respectively as shown in Figure 4.2.

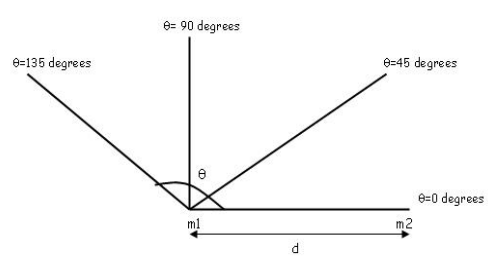


Figure 4.2: Relationship between a pair of pixels.

The co-occurrence matrices so defined are not symmetric. A symmetric one can be computed by the following expression:

$P(i,j,d,\theta) = \frac{P(i,j,d,\theta)+P(i,j,d,\theta)^T}{2}$ where $P(i,j,d,\theta)^T$ is the transpose of $P(i,j,d,\theta)$.

After making the GLCM symmetrical, it should be normalized by dividing each entry in the matrix by the sum of all possible GLCM entries [67].

Example 4.2 shows a co-occurrence matrix computation with the same digital image used in Example 4.1.

Example 4.2. *Recall that the digital image is a 3 × 3 image having 7 different grey levels which are 121, 162, 174, 185, 205, 215 and 242. So, the size of the co-occurrence matrix will be 7 × 7.*

Four matrices should be computed, namely $P0$, $P90$, $P45$ *and* $P135$. *Each one corresponds to a specific orientation. So, we will explain how to construct the horizontal matrix, i.e. P0 having a distance of 1 between the pair of pixels.*

Note that, by default, the distance that is usually used in the construction of co-occurrence matrices is 1.

Once P0 is explained, the remaining matrices have the same principles. The only difference is the orientation between the pair of pixels.

First, we construct a matrix having 7 rows and 7 columns. Each row and column correspond to a grey level.

The question here is how to compute the number of frequencies in the original image given a distance d *and an angle* θ. *In fact, to calculate an entry of the co-occurrence matrix, for example* $(121, 121)$ *where 121 designs the pixel* m_{rc1} *and the second 121 is* m_{rc2}, *we have to find the number of frequencies of this pair of pixels in the horizontal orientation. In our case, we have: #(121,121)=0; #(121,162)=1; #(121,174)=0; #(121,185)=0; #(121,205)=0; #(121,215)=0; #(121,242)=0;*

This procedure is applied for each pair of pixels and the result is as follows in the horizontal

matrix below:

0°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	1	0	0	0	0	0
162	0	0	0	1	0	0	0
174	0	1	0	0	0	0	0
185	0	0	0	0	0	0	0
205	0	0	0	0	0	0	1
215	0	0	1	0	0	0	0
242	0	0	0	1	0	0	0

Table 4.4: Horizontal co-occurrence matrix.

After the construction of the matrix, we have to check if it is a symmetric one and we have to normalize it by dividing each matrix entry by the sum of entries.

The symmetric matrix is obtained by computing the number of times a value a is adjacent to value b and the value b is adjacent to a, i.e. both a,b and b,a pairings. In our case, the symmetric matrix is expressed by Table 4.5 and the normalized matrix is expressed by Table 4.6:

0°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	1	0	0	0	0	0
162	1	0	1	1	0	0	0
174	0	1	0	0	0	1	0
185	0	1	0	0	0	0	1
205	0	0	0	0	0	0	1
215	0	0	1	0	0	0	0
242	0	0	0	1	1	0	0

Table 4.5: Symmetric horizontal co-occurrence matrix.

The demonstration above shows the steps of construction of horizontal co-occurrence matrix. The same steps will be applied to co-occurrence matrices having respectively an orientation of 90°, 45° and 135° and the resulting matrices are expressed by Tables 4.7, 4.8 and 4.9:

Once co-occurrence matrices are constructed, a feature calculation step is primordial in order to use the information contained in these matrices. Hence, Haralick et al. [28] defined

0°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	0.0833	0	0	0	0	0
162	0.0833	0	0.0833	0.0833	0	0	0
174	0	0.0833	0	0	0	0.0833	0
185	0	0.0833	0	0	0	0	0.0833
205	0	0	0	0	0	0	0.0833
215	0	0	0.0833	0	0	0	0
242	0	0	0	0.0833	0.0833	0	0

Table 4.6: Normalized symmetric horizontal co-occurrence matrix.

90°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	0	0	0	0	0.1	0
162	0	0	0.1	0	0	0	0
174	0	0.1	0	0	0	0	0.1
185	0	0.2	0	0	0	0	0
205	0	0	0	0	0	0.1	0
215	0.1	0	0	0	0.1	0	0
242	0	0	0.1	0	0	0	0

Table 4.7: Vertical co-occurrence matrix.

45°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	0	0.1428	0	0	0	0
162	0	0.1428	0	0	0	0	0
174	0.1428	0	0	0.1428	0	0	0
185	0	0	0.1428	0	0	0	0
205	0	0	0	0	0	0	0
215	0	0	0	0	0	0	0.1428
242	0	0	0	0	0	0.1428	0

Table 4.8: Right diagonal co-occurrence matrix.

135°	**121**	**162**	**174**	**185**	**205**	**215**	**242**
121	0	0	0	0	0	0	0
162	0	0	0	0	0	0.1428	0
174	0	0	0	0.1428	0.1428	0	0
185	0	0	0.1428	0	0	0	0
205	0	0	0.1428	0	0	0	0
215	0	0.1428	0	0	0.1	0	0
242	0	0.1428	0	0	0	0	0

Table 4.9: Left diagonal co-occurrence matrix.

14 statistical measures which measure textural characteristics.

The statistical values computed from co-occurrence matrices represent the texture parameters, and are typically calculated for a pixel by considering a square window around the pixel.

Within the most popular statistical features, we cite: *angular second momentum (ASM), contrast (CON), entropy (ENT), cluster shade (CS), cluster prominence (CP), inertia (IN), and local homogeneity (LH)* [62]. These features are called second order texture features since they take into account the spatial information represented in GLCM.

The spatial co-occurrence texture parameters (defined for the G by G neighborhood surrounding the pixel) are as follows:

- ***Angular second momentum*** also called *energy*. It is a measure of homogeneity of an image. In fact, an homogeneous image contains only a few grey levels, giving a GLCM with only a few but relatively high values of $P(i,j)$. Thus, the sum of squares will be high. So the energy of an image is high when we have an homogeneous image. Formally, energy is expressed by:

$$\text{ASM} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i,j)^2 \tag{4.8}$$

- ***Contrast*** is also called *sum of squares variance*. This measure presents local variations in grey levels. If they are important (i.e. if there are few homogeneous regions), thus contrast will be high. This setting also characterizes the dispersion of matrix values around the diagonal. It favours contributions of $P(i,j)$ away from the diagonal, i.e.

$i \neq j$. It is defined by:

$$\text{CON} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} |i-j|^k \; P(i,j) \tag{4.9}$$

Typically k =2.

- ***Absolute value*** is called the *dissimilarity feature*. It is expressed by:

$$\text{ABS} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} |i-j| P(i,j) \tag{4.10}$$

- ***Entropy***: this descriptor measures the randomness of the GLCM elements. When all elements are maximally random, entropy has its highest value. So, an homogeneous image has a lower entropy than an inhomogeneous image. In fact, when energy (ASM) gets higher, entropy should get lower. Formally, the entropy is defined by:

$$\text{ENT} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} -P(i,j) \ln P(i,j) \tag{4.11}$$

- ***Cluster shade*** and ***Cluster prominence***: measure the skewness of GLCM, in other words the lack of symmetry. When CS and CP are high, the image is not symmetric. Formally, CS and CP are expressed by:

$$\text{CS} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i+j-\mu_x-\mu_y)^3 \; P(i,j) \tag{4.12}$$

$$\text{CP} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i+j-\mu_x-\mu_y)^4 \; P(i,j) \tag{4.13}$$

where μ_x denotes the average of the row's sum of the co-occurrence matrix. It is defined as $\sum_i \; i \sum_j P(i,j)$ and μ_y denotes the average of the column's sum of the co-occurrence matrix. It is defined as $\sum_j \; j \sum_i P(i,j)$.

- ***Local homogeneity***: also called *inverse difference moment (IDM)*. It has a relatively high value when the high values of GLCM are near the main diagonal. This is because the squared difference $(i-j)^2$. If it is smaller, the value of $\frac{1}{1+(i-j)^2}$ increases. This feature is influenced by the homogeneity of the image. Because of the weighting factor $\frac{1}{1+(i-j)^2}$, LH will get small contributions from inhomogeneous areas (i $\neq$ j). The result is a low LH value for inhomogeneous images, and a relatively higher value for homogeneous images. Formally, this feature is expressed by:

$$\text{LH} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} \; P(i,j) \tag{4.14}$$

- ***Inertia*** : This feature gives the opposite effect. When the high values of the matrix are further away from the main diagonal, the value of inertia becomes higher [59].

$$\text{IN} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 \; P(i,j) \tag{4.15}$$

In what follows we present Example 4.3 demonstrating the procedure of computation of textural features from horizontal co-occurrence matrix.

Example 4.3. *Using Matlab, the energy feature is equal to 0.0828. The aim of this example is to explain how did we find this value from the horizontal co-occurrence matrix.*

Recall that the principle of ASM is computing the square of each entry in the co-occurrence matrix. Then, the output matrix is summed to obtain the energy value.

In our case, the output matrix will be the following:

$$\begin{pmatrix} 0 & 0.0069 & 0 & 0 & 0 & 0 & 0 \\ 0.0069 & 0 & 0.0069 & 0.0069 & 0 & 0 & 0 \\ 0 & 0.0069 & 0 & 0 & 0 & 0.0069 & 0 \\ 0 & 0.0069 & 0 & 0 & 0 & 0 & 0.0069 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0069 \\ 0 & 0 & 0.0069 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0069 & 0.0069 & 0 & 0 \end{pmatrix}$$

If we sum this matrix, we found 0.0828. This value represents the ASM feature for the image. Applying this procedure for each co-occurrence matrix, we will obtain 4 values of ASM. The question here, what is the best value that will be representative of energy?

In this case, Haralick et al [28] propose to compute the average of each feature since several matrices exist and each one corresponds to a specific orientation and resumes their textural features.

Table 4.10 presents the textural features and their average from each co-occurrence matrix.

Figure 4.3 shows examples of second-order (co-occurrence or "Haralick") textures for a brain tumor image issued from MRI. From these images, we can see that the relationship between features and pixels classes (tumor or normal) is not obvious. This is can be argued by the

Angle	**ASM**	**CON**	**LH**
0°	0.0828	3.8333	0.375
90°	0.1111	6.8333	0.375
45°	0.1563	2.75	0.5625
135°	0.1563	8.25	0.475
Average	0.1266	5.41665	0.446875

Table 4.10: Textural features from different co-occurrence matrices

fact that tumor regions can occur for example on both low and high entropy. So combining features can be effective in order to discriminate between tumor and normal pixels [62]. For example, the classifier can learn that a high T2 intensity and a low entropy are indications of tumor. For this reason, we have chosen several features in our work.

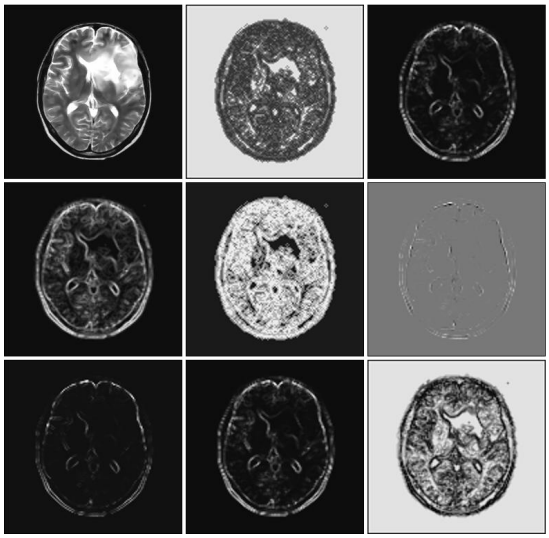


Figure 4.3: Examples of second-order (co-occurrence or "Haralick") textures for brain tumor image. Top, left to right: Original image, angular second momentum, contrast. Middle, left to right: absolute value, entropy, cluster shade. Bottom, left to right: Cluster prominence, inertia, local homogeneity.

4.4 Conclusion

In this Chapter, we have presented the features extraction phase. The first features, used in our work, are pixels intensities in different MR modalities. These features are important due to the fact that images have been registered in the previous phase (i.e. pre-processing phase). The second features are texture features. We have explored both first-order and second-order texture parameters. The first-order parameters which are called (*statistical moments*) are *mean*, *variance*, *skewness*, *kurtosis*, *energy*, and *entropy*. While the second-order ones, we have chosen the most commonly used features; the *Haralick* features which are *angular second momentum*, *contrast*, *dissimilarity*, *entropy*, *cluster shade*, *cluster prominence*, *local homogeneity* and *inertia*.

Next Chapter provides experimental results of our segmentation system using two kinds of databases, having different tumor characteristics and different tumor localizations.

Chapter 5

Experimental study

5.1 Introduction

The evaluation of the performance of brain segmentation methods using classification approaches is not a trivial task, since many challenges should be taken into consideration, namely the construction of a training set, the lack of a gold standard and the comparison to other segmentation methods. In fact comparing to other domains, real datasets relative to brain tumors are not available (free) and most of existing segmentation methods only provide a summary description of their tools and their experimental results.

This Chapter first describes our experimental data, then it presents the implementation of our segmentation system. Finally, it presents different experimental results allowing us to evaluate the performance of our system.

5.2 Experimental data

The first challenge in the experimental study is to find the adequate data since brain tumors are rare and real MR images are not available. This leads us to use two kinds of data i.e. real data collected from *El Manar clinic*, relative to 3 patients and simulated data.

5.2.1 Real data

Real data, obtained by MRI device, have several characteristics, namely slices dimensions (304 $\times$ 512, 333 $\times$ 320, etc.), number of slices, inter-slice distance, kind of scans orientations (coronal, sagittal, axial), kind of modalities (T1, T2, etc.). These parameters vary differently from one patient to another depending on tumor requirements. In our study, due to the lack of

data, we will use only axial slices during the training phase, while the whole scans orientations will be tested during testing (i.e. we train our system by axial views then we test it by axial, coronal and sagittal views).

Our real database is relative to 69 axial images of 3 patients collected from August 2008 to March 2009 such that patients' ages are 34, 42 and 54 years having the *Gliomas*[1] tumor of grade I, II and IV, respectively. From each patient, a set of images are chosen for training and the remaining tumor slices will be used during testing as shown in Table 5.1. This choice is justified by the fact that slices are scanned from the top of the patients' brain and downward and a small number of slices contain tumor regions.

Patient	**N. of axial slices**	**N. of tumor slices**	**Training slices**	**Testing slices**
1	23	6	3	3
2	23	5	3	2
3	23	8	4	4

Table 5.1: Real data characteristics.

Note that a subset of tumor types (Gliomas) has been studied in this work since recently this tumor has been the focus of many studies [33], [53], [55] due its complexity. Figure 5.1 shows one axial tumoral view of the 3 patients having Gliomas in different grades.

Once images of interest are chosen, training is started. Tumor regions are selected by manual segmentation of MR images of the head. In fact, axial T2-weighted images are displayed on the computer screen, one slice at a time. Next, representative regions of interest (ROI's) are selected interactively on the computer screen using a mouse-driven interface and pixels within any of the defined regions of interest are labeled as tumor, while pixels outside of tumor contours are labeled as normal. So data containing the labels of the regions, the corresponding intensities in different modalities (T1 and T1 after injection of contrast agent) and texture values are created and stored in a file. The created file that represents a set of tumor and normal pixels extracted from the chosen training images is then used to train SVM.

[1]Gliomas are diffuse brain tumors that originate from a single glial cell and spread to adjacent healthy tissue as the number of tumor cells increases. Malignant gliomas are the most common primary tumors of the central nervous system.

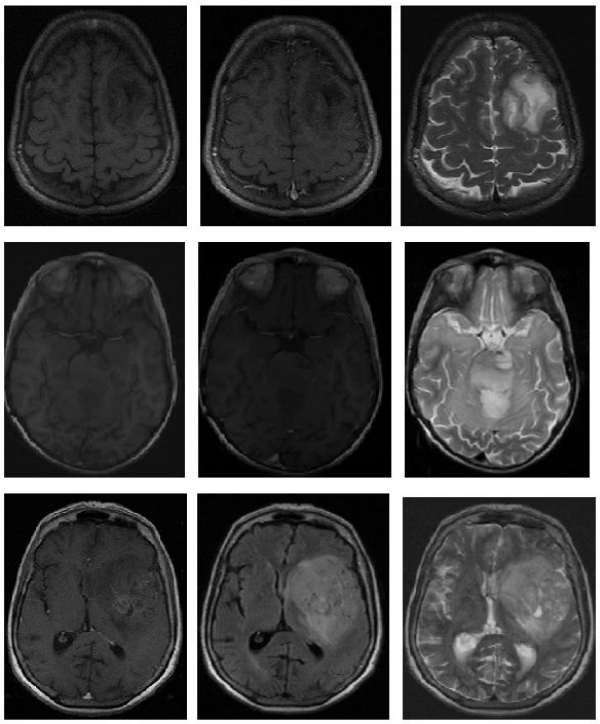


Figure 5.1: Axial views of MR real images. From top to bottom: Patient 1, Patient 2 and Patient 3. From left to right: T1, contrast enhanced T1 and T2 images.

This file consists of 90000 pixels (45 000 and 45 000 are respectively considered as tumor and normal pixels), each with 18 features values (class label, intensities in different modalities and texture). Prior to running our method, we apply a normalization procedure to transform pixels features between -1 and 1.

5.2.2 Simulated data

Recently, brain tumor data[2] [54] have been simulated and the main idea behind them is the insertion of pathology into a healthy brain MRI and the simulation of tumor effects. Within tumor effects, we cite: contrast enhancement, local distortion of healthy tissue, infiltrating edema adjacent to tumors, etc.

These data show significant mass effect, a large surrounding edema and a ventricle displacement as shown in Figure 5.13 where one axial view for each simulated volume is displayed.

Simulated volumes (3D images) are characterized by a slices dimension of 333×320, an inter-slice distance of 3 mm, a thickness of 4 mm and a number of slices between 100 and 180 for each volume. We note that only axial scans are simulated in these data. So, our training and testing images are selected from a large series of axial images of the whole head. In fact, we have chosen 16 training slices and 16 testing slices as shown in Table 5.2.

Volume	**N. of axial slices**	**N. of tumor slices**	**Training slices**	**Testing slices**
1	180	20	4	4
2	180	25	4	4
3	180	22	4	4
4	180	20	4	4

Table 5.2: Simulated data characteristics.

The training file which corresponds to simulated data is based on 4 variables (class label and intensities in three modalities) describing 147 680 observations (73 840 and 73 840 are respectively considered as tumor and normal observations). Note that texture is not taken into consideration in these data since simulated images are uniform and they are not textured.

[2]This data set is available at http://www.ucnia.org/softwaredata/5-tumordata/10-simtumordb.html.

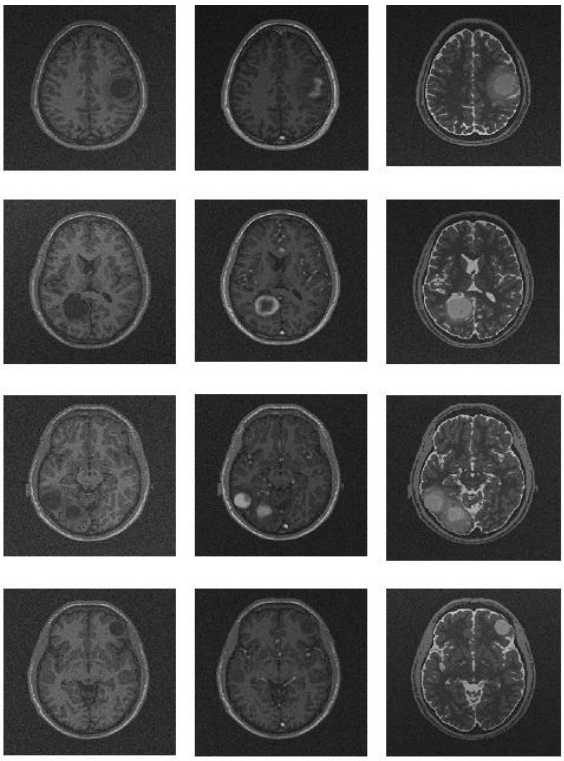


Figure 5.2: Axial views of MR simulated images. From top to bottom: the SimTumor1, SimTumor2, SimTumor3 and SimTumor4 MRI datasets. From left to right: T1, contrast enhanced T1 and T2 images.

5.3 Implementation

The implementation of our whole system needs several tools and consecutively several parameterizations. Figure 5.3 summarizes different steps and tools used in this implementation. Our choices can be summarized as follows:

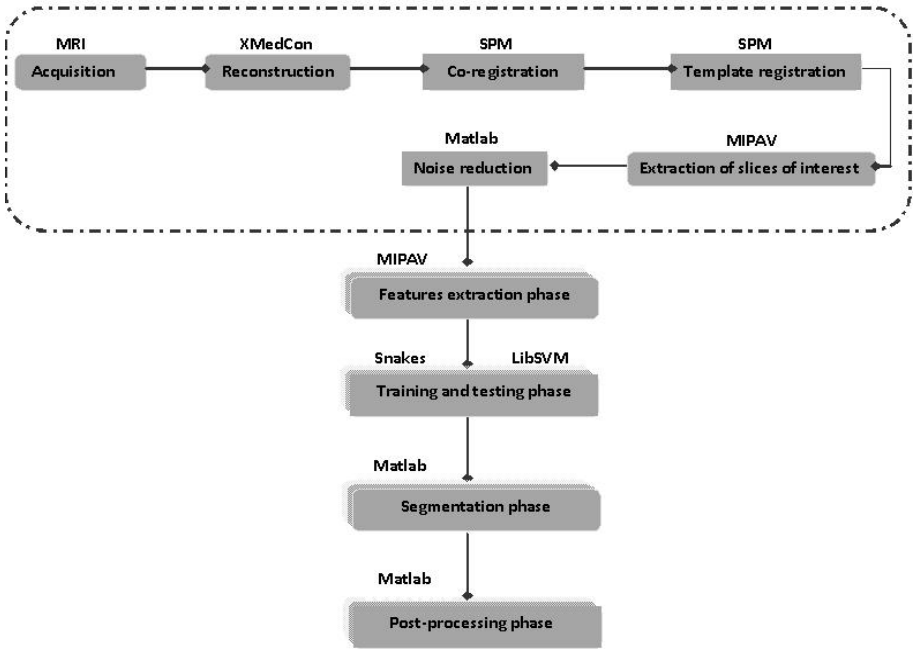


Figure 5.3: Diagram illustrating tools used in our segmentation system.

- **Acquisition**: 2D images having different orientations (i.e. axial, coronal and sagittal images) are acquired by an MRI device.
- **Reconstruction**: 2D images for each patient should be stacked into one volume. This phase is accomplished by XMedCon[3] which requires a sequence of images sorted alphabetically. This sequence is used to construct 3D images [46]. We note that a specialized doctor brings a great help if he validates the image sequence. Only real data have been reconstructed since simulated data are in three dimensions (i.e. volumes).
- **Registration**: the rigid-body model that can be parameterized by only translations and rotations is used and the objective function that is maximized in this case is the *normalized mutual information* (NMI) measure. While the measure used in template

[3] An open source toolkit for medical image conversion, available at `http://xmedcon.sourceforge.net/`.

registration is *the sum of squared difference* which needs the computation of a linear 12-parameters affine transformation. The tool used in this step is the popular registration tool SPM (see Chapter 3).

- **Extraction of slices of interest**: to ensure this task, we have used the MIPAV[4] tool which is a medical imaging software ensuring volumes visualization. The output of this phase is 2D images.
- **Noise reduction**: 2D images are denoized using the *anisotropic diffusion filtering* that is implemented in Matlab [35].
- **Features extraction**: for each pixel, the features vector is restricted to intensities values and texture values. For texture values, especially Haralick features the parameters used in computation of co-occurrence matrices are as follows: a distance of 1 between pixels pair and directions of north-south, northeast-southwest, east-west, southeast-northwest are taken into account (i.e. 0°, 45°, 90° and 135°). This step is ensured by MIPAV.
- **Training and testing**: during training, *Snakes*[5] has been used in order to assist doctors in tumors delimitation, and the package of SVM that has been used in our work is the LibSVM[6] package.
- **Segmentation**: for each image, we have used the Image Processing Toolbox of Matlab to construct a binary image in order to underline the segmentation result.
- **Post-processing**: in this step, we have used morphological filters implemented in the Image Processing Toolbox of Matlab. Note that tumor regions have been dilated and eroded by square and diamond structuring elements.

Figures 5.4, 5.5 and 5.6 are relative, respectively to SPM interface in a co-registration task, Snakes interface during the training phase and MIPAV interface in a features computation process.

5.4 Segmentation results

An important difficulty we have to face in segmentation methods is the lack of a gold standard, which can be handled by drawing tumor regions boundaries, then a radiologist should

[4]Medical Image Processing, Analysis, and Visualization is an application that enables quantitative analysis and visualization of medical images of numerous modalities, available at `http://mipav.cit.nih.gov/`.

[5]Snakes tool, available at `http://iacl.ece.jhu.edu/projects/gvf/`.

[6]An integrated software for support vector classification, available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

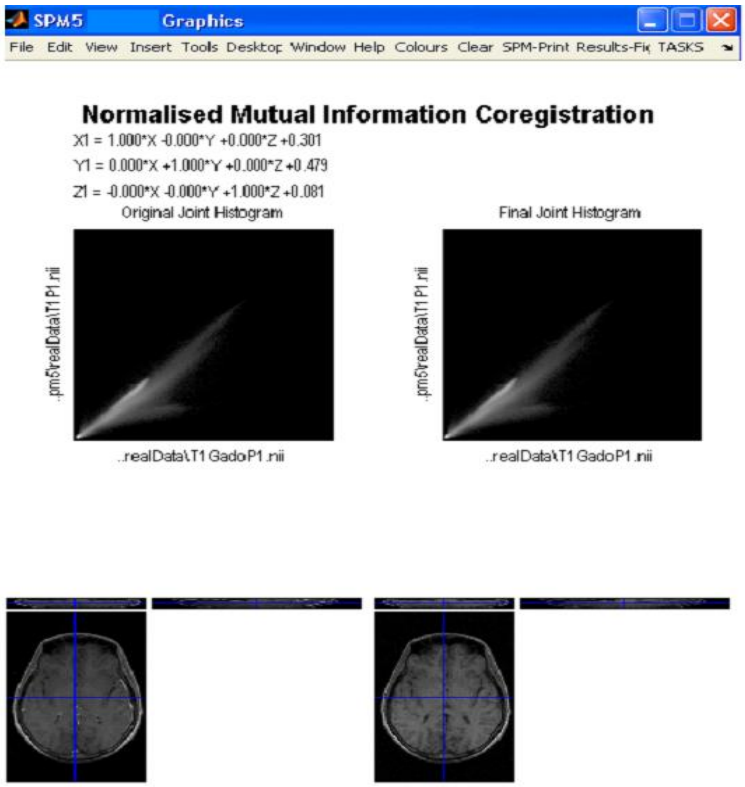


Figure 5.4: SPM interface in a coregistration task.

check them. However, the task of hand-labeling ground truth is very complex [75]. Thus in our experiments we propose to use the *Snakes* segmentation method [15]. It is a user-guided tool used to delimit regions of interest. In fact, we should fix several points of interest from the tumor region, then these points will be linked then deformed in order to obtain tumor contours. The output of this process will be considered as the segmentation ground truth.

Another challenge in such problems is the choice of evaluation measures. In fact, our work consists mainly on applying a learning machine to a segmentation problem. So it seems interesting to handle measures used in machine learning experimentations, namely the *Percentage of Perfect Classification*, denoted PCC and given by :

$$PCC = \frac{\text{Number of well classified tumor and normal pixels}}{\text{Total number of classification pixels}} \times 100 \tag{5.1}$$

However, we should not forget that our aim is the extraction of the tumor region, so a criteria that takes into consideration only tumor pixels should be used. In our experiments the

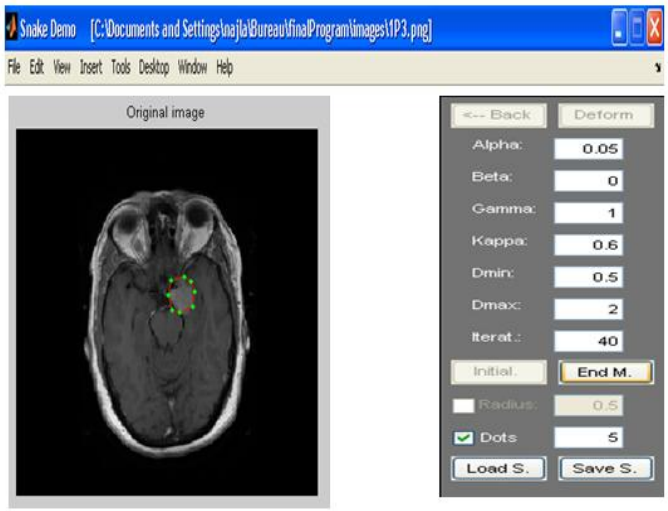


Figure 5.5: Snakes interface during the training phase.

Match Percent (MP) which is the commonly used criteria in segmentation systems [43] will be used. This criteria is calculated as the direct ratio of the tumor pixels found algorithmically to the number of ground truth tumor pixels. It is expressed by Equation 5.2:

$$MP = \frac{\#TP}{\#GT} \tag{5.2}$$

where symbol # denotes the cardinality of a set. Therefore, MP is the percentage of correct match between our segmentation results TP and the ground truth GT. It measures the similarity between the two segmentations. It will be 1 if segmentations are identical, while it approaches to 0 for completely dissimilar segmentations. It is used to give an objective evaluation of our segmentation results with respect to the ground truth GT.

Also, the choice of parameters should be discussed, especially kernels choices and their parameters. Recent studies have pointed out that kernels functions are not equally useful. Among them, the preferable choice is the RBF kernel [61]. Moreover, Vapnik argued in [72] that on the basis of experiments the choice between kernels does not make a big difference in empirical performance. The more important, and usually the more difficult problem is the selection of the parameters of kernel function.

So in our experiments, we have used the RBF kernel which owns two parameters, namely: the penalty parameter C and the RBF width γ. To know the best values of C and γ for our problem, a model selection should be applied. The goal behind such models is to identify

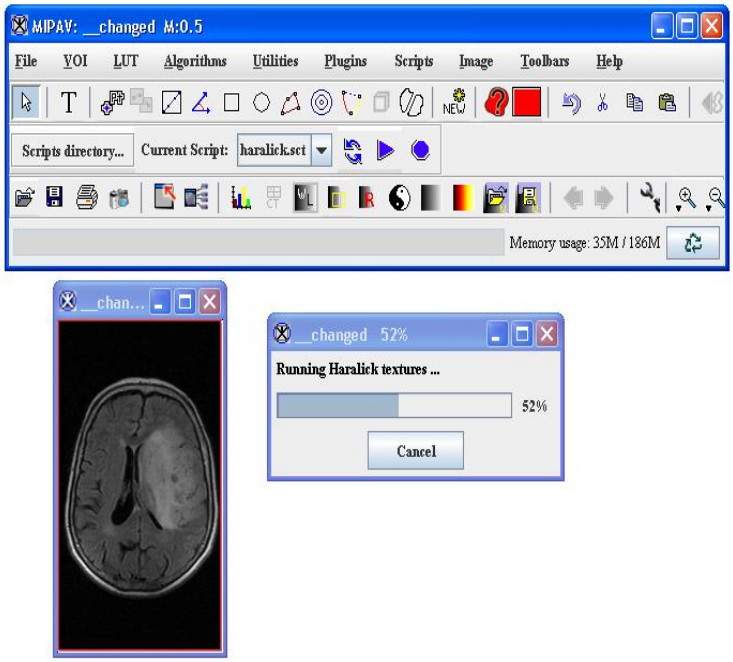


Figure 5.6: MIPAV interface in a features computation task.

"good" parameters that achieve a high *accuracy* (PCC).

A common way to do it is to dispatch training data into several parts of which a part is considered unknown in training the classifier. Then the prediction accuracy on this part can reflect the performance on classifying unknown data. This procedure is called *cross-validation*. More precisely, in v-fold cross-validation, the training set is first divided into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining $v-1$ subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. In our experiments, we have used v=10.

To obtain the best parameters, a *grid-search* on C and γ parameters using cross-validation can be applied [30]. Basically pairs of (C; γ) are tried and the one with the best cross-validation accuracy is picked. The main question here: *what is the interval of values that should be used in a grid search procedure?* In fact, many experiments have found that trying exponentially growing sequences of C and γ is a practical method to identify good parameters (for example, $C = 2^{-5}; 2^{-3}; \ldots; 2^{15}, \gamma = 2^{-15}; 2^{-13}; \ldots; 2^{3}$) [30]. This process is implemented by an *anytime algorithm* which generates an approximation of the best values using the available information. It can be stopped at any time and even it stills running, accuracy can be

improved. In fact, for each iteration, a set of parameters values is tested and their accuracy is computed. Note that these values change depending on a step chosen by the user. In our experiments, we have chosen a step of 1 for the transition from one parameter value to another.

In what follows we propose results obtained on real and simulated data, respectively.

5.4.1 Results on real data

Applying the grid search to real data provides 79.6% as the best SVM accuracy during training for C=16 and γ=2 as shown in Table 5.3 where only important variations on PCC values are presented.

C	γ	**Training accuracy** (%)
1	0.0078125	57.1677
1	0.5	66.6017
8	0.0078125	68.06857
8	0.5	71.955
8	1	72.98
16	2	79.6

Table 5.3: Important variations on accuracy values for real data.

Using these parameters, we train SVM, then we use the generated model (i.e. support vectors, in our case we have 56 743 support vectors extracted from 90 000 samples) in order to test unlabeled data. More precisely, we compute the value of the decision function for each testing pixel, then we affect it to the normal or tumor class depending on the function polarity. After that we construct a binary image where the tumor pixels, judged by SVM, are white and black pixels are normal pixels. Note that only brain pixels (the object of interest) are taken into consideraction and all background pixels are not handled.

Table 5.4 shows the comparison between quantitative measurement of segmentation accuracy with respect to the ground truth. Listed in this table are SVM accuracy (PCC), ground truth tumor pixels (GT), number of tumor pixels found algorithmically after post-processing (TP) and *Match Percent* (MP).

In our experiments, the comparison results are in terms of PCC, $\# \, TP$, $\# \, GT$ and MP.

Patient number	**PCC** (%)	**GT**	**TP**	**MP** (%)
1	74.5808	2538	2063	81.28
1	78.7723	2265	1587	70.06
1	66.7731	2611	1731	66.29
2	73.456	2152	2085	96.88
2	91.3326	360	263	73.05
3	73.0202	3919	2873	73.3
3	67.5441	4607	4200	91.16
3	64.4454	4218	3628	86.01
3	78.3686	3040	2432	80
Average	74.2547	-	-	79.7811

Table 5.4: Comparison of the ground truth and our segmentation outputs for axial images.

The higher the MP value, the larger $\# \ TP$ (i.e. correct tumor segmentation) contained in results is. From Table 5.4, we can see that the segmentation results achieve encouraging percentage of correct match to the ground truth since the average value of MP is 79.7811%.

Figures 5.7, 5.8 and 5.9 show our segmentation results before and after post-processing versus their respective ground truth to give a visual comparison.

As we have mentioned above, our training data is composed of axial slices, so it is obvious that testing images have the axial orientation. The question here : *what is the performance of SVM if we use coronal and sagittal slices?* Thus, in what follows we propose to test some coronal and sagittal slices of Patient 1 and 2 as shown in Table 5.5, Figures 5.10, 5.11 and 5.12.

To summarize, the average value of PCC is 74.2547% and 79.9% for respectively axial and both coronal and sagittal slices, while the average of MP values is 79.7811% for axial views and 91.12% for coronal and sagittal views. From these percentages, we can notice that SVM is able to segment coronal and sagittal slices even it is trained by axial views. This can be explained by both the fact that SVM learns the characteristics of tumor regions (intensities values, texture values) and do not consider their localizations and the fact that SVM is robust in generalization properties. We can conclude that our segmentation results of real data are promising and encouraging by radiologists in this field.

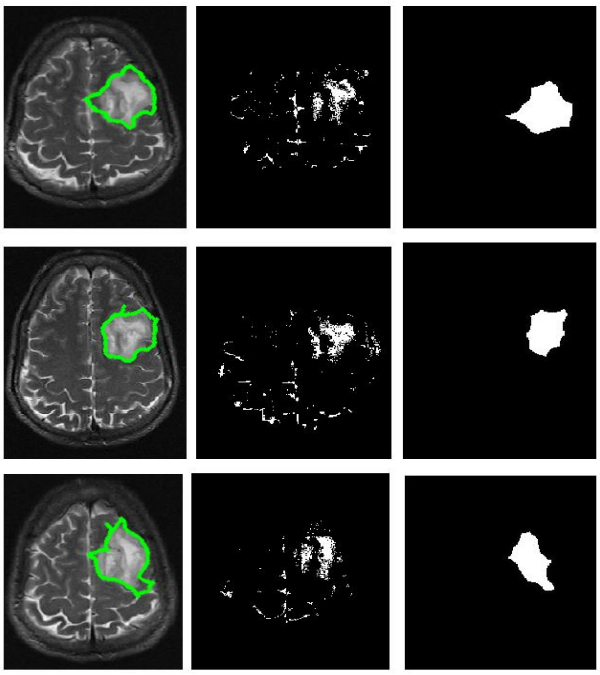


Figure 5.7: Segmentations of axial slices of Patient 1. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing.

5.4.2 Results on simulated data

To obtain the best values of C and γ of the RBF kernel, we have applied the grid search and cross-validation techniques and results are shown in Table 5.6 where the best PCC is 62.98% for C=1 and $\gamma = 2$. After the achievement of optimal parameters, we train SVM and we obtain 87 426 support vectors among 147 680 samples. These support vectors will be used in classifying new unknown data. Results are summarized in Table 5.7 and represented in Figures 5.13, 5.14, 5.15 and 5.16.

From Table 5.7, the average value of PCC is 66.4964% which is lower than those of real data (74.2547%). Hence, we can say that segmentation results of real data achieve higher percentage of correct classification than the segmentation results of simulated ones. This is due especially to several factors. The main factor is the quality of simulated images. In fact, they are not as clear as real data due to their darkness. The second factor is deduced from the first one. In fact, texture that is computed for each real image, can not offer a valuable information in simulated images because these latters provide dark texture images that do not

Orientation	**Patient number**	**PCC** (%)	**GT**	**TP**	**MP** (%)
Coronal	1	86.7076	1440	1369	95.069
Coronal	1	82.7991	1873	1858	99.19
Sagittal	1	89.4634	1166	1096	93.99
Sagittal	1	88.8658	1698	1586	93.40
Sagittal	2	80.5122	3913	2861	73.11
Sagittal	2	69.8705	5155	4742	91.98
Average	-	79.90	-	-	91.12

Table 5.5: Comparison of the ground truth and our segmentation outputs for coronal and sagittal images.

C	γ	**Training accuracy** (%)
0.5	0.0078125	56
0.5	0.125	56.5
0.5	0.5	57
1	0.0078125	57.5
1	0.125	60
1	0.5	62.98

Table 5.6: Important variations on accuracy values for simulated data.

Volume number	**PCC** (%)	**GT**	**Tumor pixels**	**MP** (%)
Volume 1	67.8744	2718	2335	85.9
Volume 1	63.8411	3004	2447	81.45
Volume 1	62.7754	3288	2634	80.01
Volume 1	63.4579	3754	3087	82.22
Volume 2	66.8277	2114	1612	76.25
Volume 2	68.514	2043	1597	78.16
Volume 2	62.8218	1481	1132	76.43
Volume 2	69.7577	1025	734	71.6
Volume 3	70.6465	1224	841	68.7
Volume 3	67.3196	4929	4322	87.68
Volume 3	68.0487	4983	4223	84.74
Volume 3	66.7376	4728	3891	82.29
Volume 4	64.9444	947	553	58.39
Volume 4	65.2483	723	575	79.52
Volume 4	67.0039	765	748	97.77
Volume 4	68.1238	700	692	98.85
Average	66.4964	-	-	80.6225

Table 5.7: Comparison of the ground truth and our segmentation results.

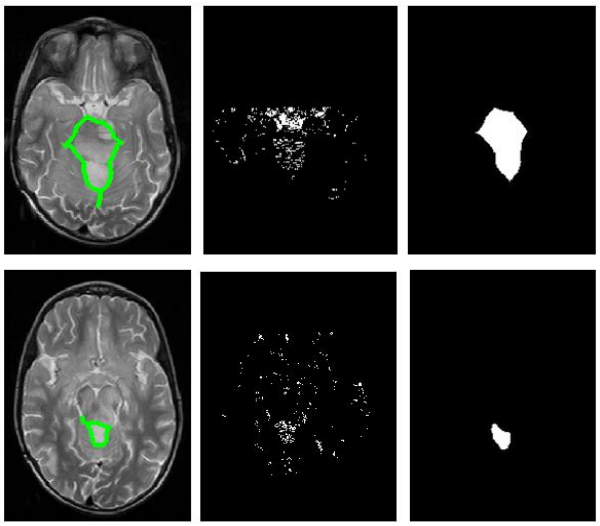


Figure 5.8: Segmentations of axial slices of Patient 2. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing.

allow the discrimination between tumor and normal regions. So in this case, only intensities features have been taken into consideration, which explains the reduction of PCC from 74% to 66%.

5.5 Conclusion

The experimentation study, provided in this Chapter, shows that our brain tumor segmentation process gives motivating results using both simulated and real data. The fact that segmentation results on real data are better than those relative to simulated data (around 75% for real data and 66% for simulated data) can be explained by the fact that the quality of real data is better than simulated ones. Moreover, the combination of both intensities and texture features is able to discriminate between tumor and normal regions, better than using only intensities features, as explained previously.

In our experiments, many technical constraints should be taken into consideration, namely the quality of images and the choice of features. Hence, our results confirm that our choices are right but can be improved. For instance, in the case of real images, especially the third patient it is clear that several normal regions have the same appearance as the tumor region,

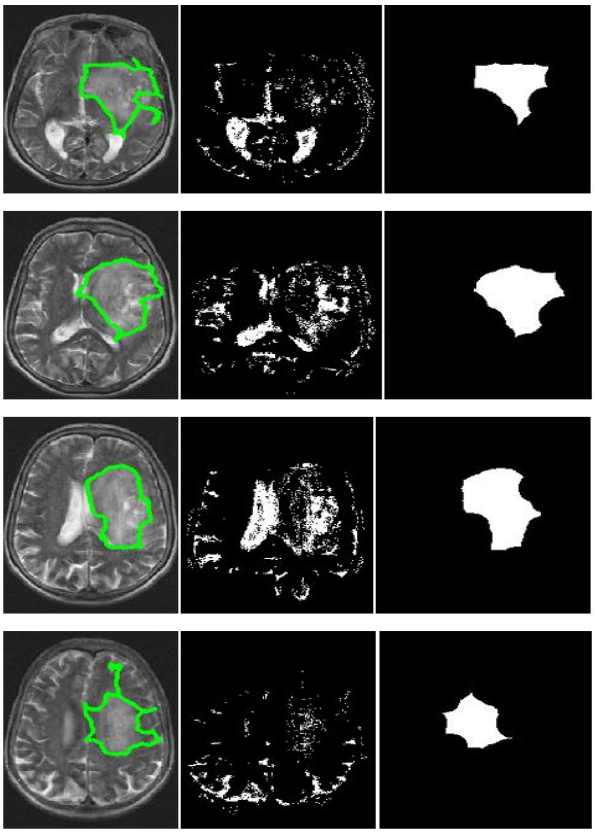


Figure 5.9: Segmentations of axial slices of Patient 3. From left to right: Ground truth of T2 axial slice, segmentation result before post-processing, and segmentation result after post-processing.

which makes the SVM classification erroneous. This point can be improved by taking into consideration tumor normalities. For example, an individual pixel can not be classified as tumor if its neighborhood pixels are not within tumor pixels. Furthermore, several scattered tumor regions (more than 2) can not be in the same brain. Thus if this assumption is taken into account, classification results may be ameliorated.

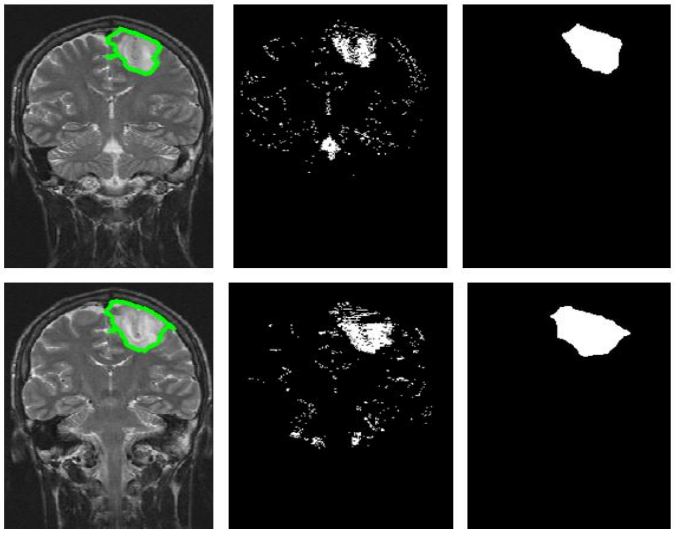


Figure 5.10: Segmentations of coronal slices of Patient 1. From left to right: Ground truth of T2 coronal slice, segmentation result before post-processing, and segmentation result after post-processing.

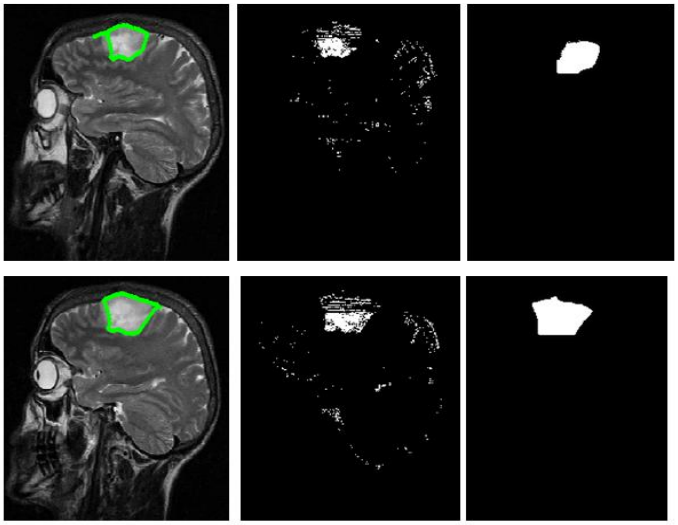


Figure 5.11: Segmentations of sagittal slices of Patient 1. From left to right: Ground truth of T2 sagittal slice, segmentation result before post-processing, and segmentation result after post-processing.

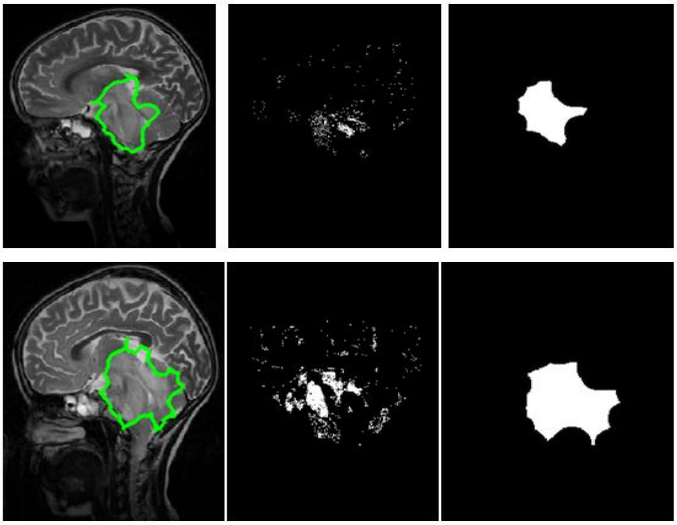


Figure 5.12: Segmentations of sagittal slices of Patient 2. From left to right: Ground truth of T2 sagittal slice, segmentation result before post-processing, and segmentation result after post-processing.

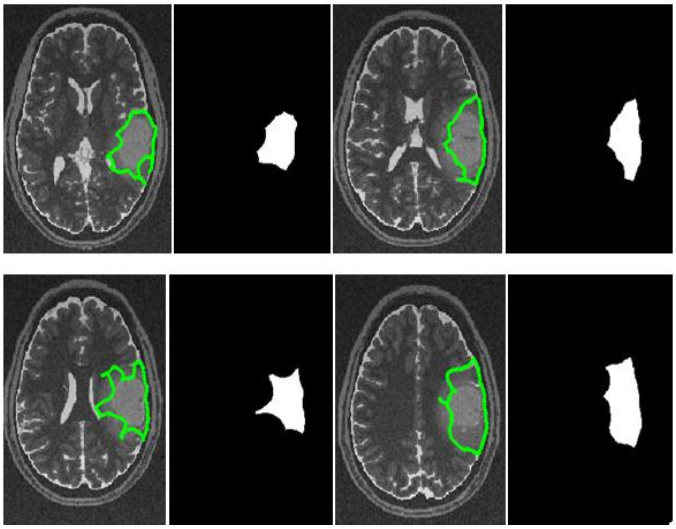


Figure 5.13: Segmentation results of Volume 1.

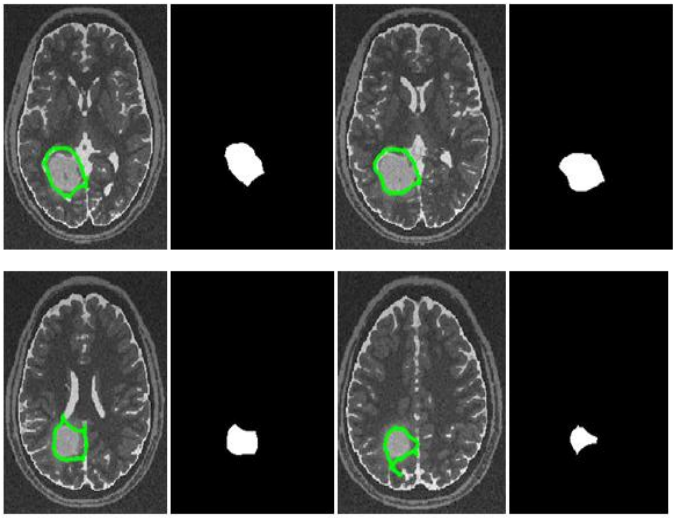


Figure 5.14: Segmentation results of Volume 2.

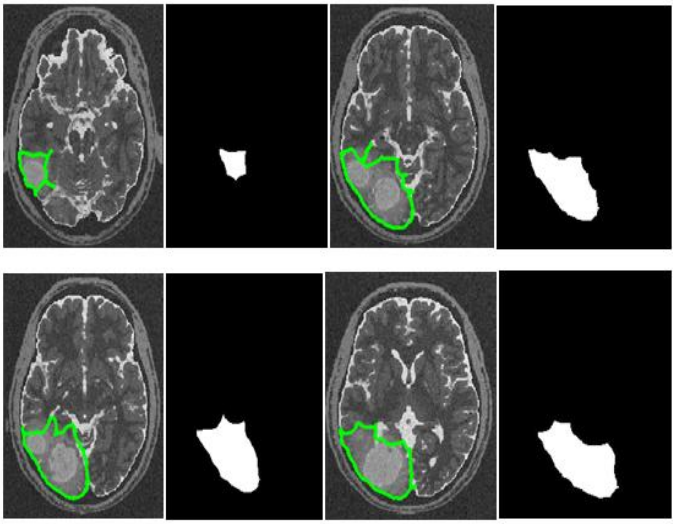


Figure 5.15: Segmentation results of Volume 3.

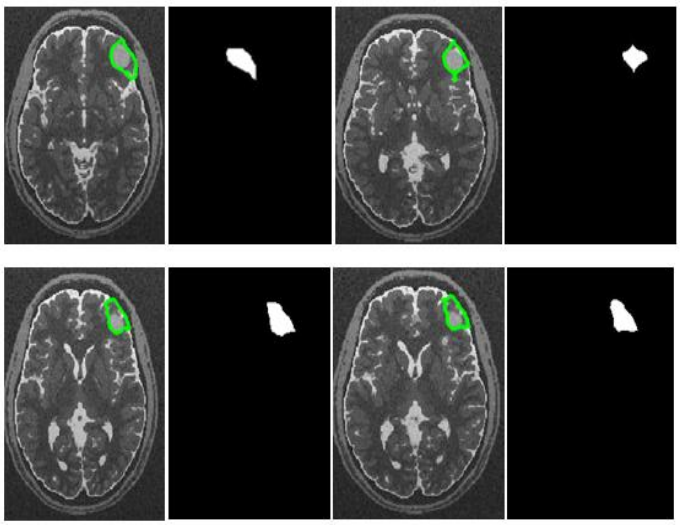


Figure 5.16: Segmentation results of Volume 4.

Conclusion and perspectives

Magnetic resonance imaging, *tomography* and other techniques allow the visualization of human anatomy in a safe and non-invasive way. These techniques have largely contributed to increase the knowledge of anatomy and its pathology.

Processing and analyzing the growing number of images by computers is becoming necessary. In particular, image segmentation facilitates the delineation of anatomical structures or other regions of interest. It plays a vital role in many medical applications such as measuring the volume of tissues, establishing of diagnosis, localizing pathologies, studying anatomical structures.

In this work we have proposed a segmentation method of MR images for the detection of brain tumors in the context of computer-aided-diagnosis. In fact, we have considered the segmentation as a classification problem where the aim is to distinguish between tumor and normal pixels. Our contribution in this work is applying the whole segmentation process from the pre-processing phase to the post-processing phase. The core of our system is based on a supervised classification approach, namely the Support Vector Machines (SVM).

We have first addressed the pre and post-processing steps where images have been co-registered with each other in order to spatially aligning them and they have been denoised in order to reduce the effect of eventual noise, introduced by image acquisition. While in the post-processing step, we have applied morphological filters to the binary segmented images due to the fact that these latters may have some cavities and some isolated regions which should be removed. Also, we have presented the classification method SVM used to segment the MR images where the input is a set of tumor and normal pixels of several training images and the output is the model that should discriminate between pixels in the new images. Besides we have presented the image-based-features used in our approach, namely intensities in different MR modalities, first-order texture and Haralick texture features.

In order to evaluate our approach we have proposed an experimental study based on both real and simulated data since real data are not available and they are usually confidential.

Segmentation results of real data are more promising than those on simulated ones. This is mainly due to the quality of images since as shown in Chapter 5 simulated images are dark so their texture images are considered meaningless. Indeed, we have obtained a PCC of around 75% for a set of real axial testing images, 80% for real coronal and sagittal testing images and 66% for simulated axial images. We have also computed the MP values for each segmented image in order to assess our results by taking into consideration only the tumor region i.e. the target of segmentation. We have obtained an average MP of around 80% and 91% for respectively axial, coronal and sagittal real images and 81% for axial simulated images.

Main results of this work are accepted for publication in [6]. The proposed system can be considered as a first attempt in the segmentation of medical images and it opens several future directions.

Our first line of research consists on performing a spatial modeling in pixels data set since existing classification methods still do not take into account the spatial relationships between pixels. Therefore, neighbourhood information can be incorporated into the classifier approach as an additional feature. For instance a pixel having the same behaviour as a normal pixel, and represents a neighbour to a tumor pixel has a higher probability to be within tumor pixels than background pixels and using the spatial information, it will be classified as tumor. Moreover, supervised methods require a manual interaction to obtain training data. This can be time consuming and laborious. Thus the use of semi-supervised methods [73] [78] may be of great interest. These methods use both labeled and unlabeled samples in the same data set.

In the current work, we have only explored the tumor region and we do not consider its localization with regard to brain anatomical structures. This requires both brain and tumor segmentations. In fact, the knowledge of the localization, size and shape of anatomical structures is of great importance. They help doctors to understand and interpret brain images. For instance, planning radiological treatments and surgeries and studies about the growth of tumor regions over time need the localization of the pathology relative to healthy structures. Also, characteristics of neurological disorders can be obtained by analyzing the localization

and the shape of tumor regions and anatomical brain structures. So an interesting future work is to extend our system to a segmentation process that takes into account anatomical structures while segmenting tumor regions using multi-classification SVM as detailed in Section 2.4.

Another line of research will be to handle the uncertain, complementary and the abundant aspect of MR modalities by using *the evidence theory* [63] which deals with data uncertainty and the multi-sources aspect of the handled information. Indeed, most of the proposed segmentation methods treat MR modalities in an independent way. However the fusion of information provided from different sources may improve segmentation results as proven in [13] where an evidential segmentation scheme of multi-echo MR images for the detection of brain tumors has been proposed.

Bibliography

[1] S. Aksoy. Textural features for content-based image database retrieval. Master's thesis, University of Washington, Seattle, WA, June 1998.

[2] J. Ashburner and K.J. Friston. Rigid body registration. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.

[3] J. Ashburner and K.J. Friston. Spatial normalization using basis functions. In R.S.J. Frackowiak, K.J. Friston, C. Frith, R. Dolan, K.J. Friston, C.J. Price, S. Zeki, J. Ashburner, and W.D. Penny, editors, *Human Brain Function*. Academic Press, 2nd edition, 2003.

[4] J. Ashburner, P. Neelin, D. L. Collins, A. Evans, and K. Friston. Incorporating prior knowledge into image registration. *NeuroImage*, 6(4):344–352, November 1997.

[5] N. Ayache. L'analyse automatique des images médicales: Etat de l'art et perspectives. Technical Report 3364, INRIA, February 1998.

[6] R. Ayachi and N. Ben Amor. Brain tumor segmentation using support vector machines. *To appear in Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU' 2009)*, 2009.

[7] I. Bloch. *Mathematical Morphology*. ISTE Wiley, London, UK, 2008.

[8] T. Blu and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19:739–758, 2000.

[9] M. Brown and R. Semeka. *MRI: Basic Principles and Applications*. John Wiley and Sons, Inc., 3rd edition, 2003.

[10] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[11] C. Campbell. Algorithmic approaches to training support vector machines: A survey. In *In Proceedings of ESANN2000*, pages 27–36. D-Facto Publications, 2000.

[12] A. S. Capelle. *Segmentation d'images IRM multi-échos tridimensionnelles pour la détection des tumeurs cérébrales par la théorie de l'évidence.* PhD thesis, Université de Poitiers, December 2003.

[13] A. S. Capelle and O. Colot et C. F. Maloigne. Evidential segmentation scheme of multi-echo mr images for the detection of brain tumors using neighborhood information. *Information Fusion*, 5(4):203–216, 2004.

[14] F. Catte, P. L. Lions, J. M. Morel, and T. Coll. Image selective smoothing and edge detection by non linear diffusion. *SIAM J. Numer. Anal.*, 29(1):182–193, 1992.

[15] X. Chenyang, S. Member, J. L. Prince, and S. Member. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7:359–369, 1998.

[16] X. Chenyang and J. L. Prince. Gradient vector flow: A new external force for snakes. In *IEEE Proc. Conf. On*, pages 66–71, 1997.

[17] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multimodality image registration based on information theory. In *XIVth international conference on information processing in medical imaging - IPMI95*, volume 3, pages 263–274. Kluwer Academic Publishers, 1995.

[18] C. Cortes and V. Vapnik. Support vector networks. In *Machine Learning*, pages 273–297, 1995.

[19] P. A. Van Den Elsen, M. A. Pol, and Viergever. Medical image matching – A review with classification. *IEEE Eng. Med. Biol.*, 12:26–38, March 1993.

[20] A. Evans, D. Collins, and B. Milner. An mri-based stereotactic atlas from 250 young normal subjects. In *Society for Neuroscience Abstracts*, volume 18, page 408. 1992.

[21] O. D. Faugeras and W. K. Pratt. Decorrelation methods of texture feature extraction. *IEEE Trans. Pattern Anal. Machine Intell*, 2:323–332, july.

[22] L. M. Fletcher-Heath, L. O. Hall, D. B. Goldgof, and R. Murtagh. Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. In *Artificial Intelligence in Medicine*, pages 43–63, 2001.

[23] K. J. Friston, J. Ashburner, C. D. Frith, J. B. Poline, J. D. Heather, and R. S. J. Frackowiak. Spatial registration and normalization of images. *Human Brain Mapping*, 2:165–189, 1995.

[24] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, January 1989.

[25] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[26] S. R. Gunn. Support vector machine for classification and regression. Technical report, University of Southampton, May 1998.

[27] E. M. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan. *Magnetic Resonance Imaging: Physical Principles and Sequence Design.* Wiley, USA, 1999.

[28] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973.

[29] T. Hastie and R. Tibshiraniand J. Friedman. *Elements of Statistical Learning: data mining, inference and prediction.* Springer-Verlag, 2001.

[30] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, Taipei, 2003.

[31] W. M. Wells III, P. Viola, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. In *Proceedings of the 2nd International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 55–62, 1995.

[32] P. A. Sample M. Goldbaum K. Chan, T. W. Lee and R. N. Weinreb. Comparison of machine learning and traditional classifier in glaucoma diagnosis. *IEEE Transactions on Biomedical Engineering*, 49:963–974, 2002.

[33] 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*, 218(2):586–591, February 2001.

[34] P. Kleihues, P. C. Burger, and B. W. Scheithauer. Histological typing of tumors of the central nervous system. *World Health Organisation*, 1993.

[35] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia. Available from: "`http://www.csse.uwa.edu.au/$\sim$pk/research/matlabfns/`".

[36] Z. Lao, D. Shen, Z. Xue, B. Karacali, S. Resnick, and C. Davatzikos. Morphological classification of brains via high-dimensional shape transformations and machine learning methods. *NeuroImage*, 2:46–57, 2004.

[37] J. Lecoeur and C. Barillot. Segmentation d'images cérébrales : Etat de l'art. Technical Report 6306, INRIA, September 2007.

[38] H. Li, K. Zhang, and T. Jiang. Minimum entropy clustering and applications to gene expression analysis. In *CSB04: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, pages 142–151, Washington, DC, USA, 2004. IEEE Computer Society.

[39] I. Maglogiannis, E. Zafiropoulos, and I. Anagnostopoulos. An intelligent system for automated breast cancer diagnosis and prognosis using svm based classifiers. *Applied Intelligence*, 30(1):24–36, 2007.

[40] J. M. Camino Martinez and R. P. Ané. Recalage d'images en médecine nucléaire : concepts de base. *Med Nucl*, 30:701–713, 2006.

[41] J. H. Mathews. *Numerical methods for mathematics, science and engineering*. Prentice Hall College Div, 2nd edition.

[42] J. Mazziotta, A. Toga, A. Evans, P. Fox, J. Lancaster, K. Zilles, R. Woods, T. Paus, G. Simpson, B. Pike, C. Holmes, L. Collins, P. Thompson, D. MacDonald, M. Iacoboni, T. Schormann, K. Amunts, N. Palomero-Gallagher, S. Geyer, L. Parsons, K. Narr, N. Kabani, G. Le Goualher, D. Boomsma, T. Cannon, R. Kawashima, and B. Mazoyer. A probabilistic atlas and reference system for the human brain: International consortium for brain mapping (icbm). *Philos Trans R Soc Lond B Biol Sci*, 356(1412):1293–1322, August 2001.

[43] D.B. Goldgof R. Velthuizen F.R. Murtagh M.C. Clark, L.O. Hall and M.S. Silbiger. Automatic tumor segmentation using knowledge based techniques. *IEEE Trans. on Medical Imaging*, 17(2):238–251, 1998.

[44] E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. In *Proceedings of the IEEE*, pages 319–342, 2002.

[45] J. Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications.* Springer-Verlag, 3rd edition, 2003.

[46] E. Nolf. An open-source medical image conversion toolkit. *European Journal of Nuclear Medicine*, 30:246, 2003.

[47] E. E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical report, Massachusetts Institute of Technology, 1997.

[48] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[49] M. Petrou and P. Bosdogianni. *Image Processing: The fundamentals.* Wiley, 1999.

[50] D. L. Pham, X. Chenyiang, and L. P. Jerry. A survey of current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000.

[51] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[52] M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derviatives. *The Computer Journal*, 7(2):155–162, 1964.

[53] M. Prastawa. An mri segmentation framework for brains with anatomical deviations. Master's thesis, University of North Carolina, Department of Computing Science, 2007.

[54] M. Prastawa, E. Bullitt, and G. Gerig. Simulation of brain tumors in MR images for evaluation of segmentation efficacy. *Medical Image Analysis (MedIA)*, 2008.

[55] M. Prastawa, E. Bullitt, and N. Moon. Automatic brain tumor segmentation by subject specific modification of atlas priors. *Academic Radiology*, 10(12):1341–1348, 2003.

[56] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes in C.* Cambridge University Press, 2nd edition.

[57] R. J. Quinlan. *Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning).* Morgan Kaufmann, January 1993.

[58] G. Ratsch, T. Onoda, and K. R. Muller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, March 2001.

[59] E. M. Rikxoort. Texture analysis. Master's thesis, NICI Institute, April 2004.

[60] G. X. Ritter and J. N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC, 2 edition, September 2000.

[61] B. Schlkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, December 2001.

[62] M. Schmidt. Automatic brain tumor segmentation. Master's thesis, University of Alberta, Department of Computing Science, 2005.

[63] J. Serra. *Mathematical theory of evidence*. Princston University Press, 1976.

[64] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[65] A. Shmilovici. Support vector machine. In *The Data Mining and Knowledge Discovery Handbook*, pages 257–276. Springer, 2005.

[66] C. Studholme, D. L. G. Hill, and D. J. Hawkes. Multiresolution voxel similarity measures for mr-pet registration. In *Information Processing in Medical Imaging: Proc. 14th International Conference (IPMI '95)*, pages 287–298, 1995.

[67] M. A. Tahir, A. Bouridane, F. Kurugollu, and A. Amira. Accelerating the computation of glcm and haralick texture features on reconfigurable hardware. In *ICIP*, pages 2857–2860, 2004.

[68] J. Talairach and P. Tourneaux. *Co-Planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System : An Approach to Cerebral Imaging*. Thieme Medical Publishers, January 1988.

[69] P. Tofts. *Quantitative MRI of the Brain: Measuring Changes Caused by Disease*. Wiley, 2003.

[70] M. Tuceryan and A. K. Jain. Texture analysis. In P. S. P. Wang C. H. Chen, L. F. Pau, editor, *The Handbook of Pattern Recognition and Computer Vision*. World Scientific Publishing Co.

[71] A. Ukil. Support vector machine. In *Intelligent Systems and Signal Processing in Power Engineering*, pages 161–226. Springer, Berlin, Heidelberg, 2007.

[72] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[73] V. Vapnik. Transductive inference and semisupervised learning. *Semi-Supervised Learning*, pages 454–472, 2006.

[74] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. In *Fifth International Conference on Computer Vision*, pages 16–23, 1995.

[75] W. M. Wells III, W. E. L. Grimson, R. Kikinis, and F. A. Jolesz. Adaptive segmentation of MRI data. In Nicholas Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine*. Springer-Verlag, 1995.

[76] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing pet images. *Journal of Computer Assisted Tomography*, 16:620–633, 1992.

[77] R. P. Woods, J. C. Mazziotta, and S. R. Cherry. Mri-pet registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, 1993.

[78] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *In ICML*, pages 912–919, 2003.

[79] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.