Automated Brain Structures Segmentation using the Mean-Shift algorithm

A dissertation for the licentiate
of Electronic Engineer

Department of Electronic
and Computer Engineering
Technical University of Crete

by Mavrigiannakis Konstantinos

Supervisors: Prof. *M. Zervakis*[a]
Associate Researcher *V. Sakkalis*[b]
Assistant Prof. *K. Mania*[a]

[a] *Department of Electronic* and *Computer Engineering , Technical University of Crete*
[b] *Institute of Computer Science , Foundation for Research* and *Technology , Heraklion , Greece* and *the Department of Computer Science , University of Crete , Heraklion*

Chania 2010

Crete, Greece

ABSTRACT

Magnetic resonance imagine (MRI) is a widely used medical imaging modality that provides rich information about the human tissue anatomy and pathology. Being a non-invasive and safe technique, it offers several advantages over other imaging techniques enabling it to provide images with high contrast between the three basic brain tissues, Cerebro-Spinal Fluid (CSF), Gray Matter (GM) and White Matter (WM). Also, it provides a plethora of pathophysiological tissue information that assists the clinician in diagnosis, therapy design/monitoring and surgery. Manual delineation of brain tissues by a human expert is still considered as the reference and most acceptable method, but unfortunately it is too time consuming, especially in cases where large amounts of data need to be analyzed. In addition manual segmentations by the clinicians have been reported to be prone to large intra- and inter observer variability, fact that stresses out the need of objective and reproducible computer segmentation techniques for the brain in order to perform a number of computational medicine tasks including morphological measurements of brain structures, automatic detection of asymmetries and pathologies, and simulation of brain tissue growth.

This thesis proposes an automated brain structures segmentation algorithm based on the adaptive mean-shift theory. The MRI image space is used to compute a three-dimensional feature space that includes intensity features as well as spatial features, in particular pixel's coordinates. An adaptive mean-shift algorithm clusters the joint spatial-intensity feature space, thus extracting a representative set of high-density points within the feature space, otherwise known as modes. Because of its nonparametric nature, adaptive mean-shift can deal successfully with nonconvex clusters and produce convergence modes that are better candidates for intensity based classification than the initial pixels. All pixels are then classified to these modes according to the Euclidean distance from them. Then, an intensity only feature space, is used to merge the remaining modes, to further reduce the remaining number of clusters. Finally, pixels are assigned to the three desired clusters(CSF, GM and WM), according to the fuzzy k-means clustering algorithm. The effectiveness of this algorithm in the automatic detection of brain abnormalities in brain images is also investigated in the following way: Instead of three, four clusters are used and after identifying the cluster with the higher mean value, (since most likely it includes the tumor region), the tumor area is labelled as the largest connected component of that cluster. The output of the algorithm is not only a segmentation map for all the pixels of the dataset but also a membership matrix for each cluster which includes the probability of each pixel to belong to this category.

The proposed method is validated on both simulated and real MRI data. The algorithm was also assessed for various different parameters values in order to check their effect in algorithm's performance and ensure that the proposed segmentation method is reproducible and reliable. Finally, it was compared with a number of well-known brain tissues segmentation algorithms by using a presegmented atlas dataset as ground truth. The results indicate that the mean-shift algorithm outperforms these methods.

Key words: brain magnetic resonance imaging (MRI), segmentation, Adaptive mean-shift, mahalanobis distance, fuzzy k-means.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ABBREVIATIONS

MRI	Magnetic Resonance Imaging
ROIs	Region Of Interests
CSF	Cerebro-Spinal Fluid
GM	Gray Matter
WM	White Matter
CNS	Central Nervous System
NMRI	Nuclear Magnetic Resonance Imaging
CT	Computed Tomography
RF	Radio Frequency
MRT	Magnetic Resonance Tomography
GRE	GRadient Echo
TE	Echo Time
TR	Repetition Time
PD	Proton Density
MT	Magnetization Transfer
CAT	Computed Axial Tomography scans
WHO	World Health Organization
PDF	Probability Density Function
k-NN	k-Nearest Neighbor
MS	Multiple Sclerosis
GMM	Gaussian Mixture Model
EM	Expectation-Maximization
MRF	Markov Random Field
FCM	Fuzzy C-Means
KB	Knowledge Based
FC	Fuzzy Connectedness
T1E	T1 Enhanced
GTV	Growth Tumor Volume
AMS	Adaptive Mean-Shift
BET	Brain Extraction Tool
ICM	Iterated-Conditional Mode algorithm

1. INTRODUCTION

1.1 Motivation and objectives of the study

Magnetic resonance images (MRI) of the brain are acquired using different protocols, e.g. T1 weighted images, T1 weighted images with contrast enhancement of the active tumor region, T2 and Proton Density (Pd) weighted images. In order to be able to make use of the acquired images, different regions in the images have to be delineated. The region of interests (ROIs) usually correspond to the different tissue types, which are present in the brain. Clustering brain pixels into one of three main brain tissue types (Cerebrospinal fluid, Gray Matter, White Matter) proves to be of paramount importance in anticipation and treatment of various diseases, such as multiple sclerosis, Alzheimer's disease or epilepsy.

A great variety of segmentation methods have been proposed for this task, following either supervised or unsupervised approaches. Supervised classification requires input from the user, typically a set of pixel class samples. On the other hand, unsupervised approaches often rely on a Gaussian approximation of the pixel intensity distribution for each tissue type. Though, using intensity information alone has proven insufficient for a reliable automated segmentation of the brain tissues**.** For this reason, different algorithms have been proposed that model neighboring pixels interactions using a Markov-Random field (MRF) statistical model. An alternative to statistical parametric approaches is the use of unsupervised, nonparametric schemes. One such approach is the Mean-shift algorithm , which uses adaptive gradient ascent in order to detect local maximum of data density in feature space.

Manual segmentation by an expert of the actual tumor evolution, is still considered as the reference and most accurate method, but is a time consuming task with high inter and intra-observer variability. For this reason, the development of reliable algorithms that automatically or semi-automatically detects tumor pixels, is vital in quantifying tumor, in simulation of treatment effects and finally in optimization of therapeutic strategies. Challenges in the segmentation of gliomas from MRI data are related to the infiltration of cells into the tissue, inducing unsharp borders with irregularities and discontinuities, the great variability in their contrast uptake and their appearance on standard MRI protocols. The majority of MRI tumor segmentation methods depend on region-based approach (In region-based approach,the segmentation task of the given image can be seen as the partition of the image into homogeneous objects) while some recent methods also include data from edge-based approach (In edge-based approach, the segmentation task of the given image can be seen as the detection of object contours within the image.) .

Object of this work is to propose an automated,region-based brain structures segmentation algorithm, based on the mean-shift theory, which will produce high qualitative classification results and to propose a technique to detect and classify tumor and edema pixels from T1-enhanced and T2-flair modalities.

1.2 Outline of the Dissertation

Chapter 1 gives an introduction to the concept of brain structures segmentation and the objectives of the study. **Chapter 2** introduces us a brief background of human brain anatomy. **Chapter 3** introduces us to brain image processing and brain structures segmentation concepts. **Chapter 4** demonstrates the basic theoretical parts of the proposed algorithm, such as the mean-shift theory and k-means algorithm. **Chapter 5** demonstrates analytically the proposed, mean-shift algorithm**. Chapter 6** presents experimental results, discusses the selection of optimal parameters and performs a comparison with other common algorithms. **Chapter 7** summarizes and concludes the thesis, and proposes recommendations for future work. Further related information is detailed in the **Appendix**.

2. CLINICAL BACKGROUND

2.1 Introduction

In this chapter we are going to analyze the basic tissues of human brain, what we define as brain tumor, human brain data visualization and finally why segmentation of the human brain tissues is useful in various clinical applications.

2.2 Human Brain Anatomy

2.2.1 Introduction

As mentioned in [1], human brain is the center of the human nervous system and is a highly complex organ. Enclosed in the cranium, it has the same general structure as the brains of other mammals, but is over three times as large as the brain of a typical mammal with an equivalent body size. Most of the expansion comes from the cerebral cortex, a convoluted layer of neural tissue that covers the surface of the forebrain. Especially expanded are the frontal lobes, which are associated with executive functions such as self-control, planning, reasoning, and abstract thought. The portion of the brain devoted to vision is also greatly enlarged in human beings.

Brain evolution, from the earliest shrewlike mammals through primates to hominids, is marked by a steady increase in encephalization, or the ratio of brain to body size. The human brain has been estimated to contain 50–100 billion neurons, of which about 10 billion are cortical pyramidal cells. These cells pass signals to each other via as many as 1000 trillion synaptic connections.

The brain monitors and regulates the body's actions and reactions. It continuously receives sensory information, and rapidly analyzes this data and then responds, controlling bodily actions and functions. The brainstem controls breathing, heart rate, and other autonomic processes. The neocortex is the center of higher-order thinking, learning, and memory. The cerebellum is responsible for the body's balance, posture, and the coordination of movement.

In spite of the fact that it is protected by the thick bones of the skull, suspended in cerebrospinal fluid, and isolated from the bloodstream by the blood-brain barrier, the delicate nature of the human brain makes it susceptible to many types of damage and disease. The most common forms of physical damage are closed head injuries such as a blow to the head, a stroke, or poisoning by a wide variety of chemicals that can act as neurotoxins. Infection of the brain is rare because of the barriers that protect it, but is very serious when it occurs. The human brain is also susceptible to degenerative disorders, such as Parkinson's disease, multiple sclerosis, and Alzheimer's disease. A number of psychiatric conditions, such as schizophrenia and depression, are widely thought to be caused at least partially by brain dysfunctions, although the nature of such brain anomalies is not well understood.

2.2.2 Basic brain tissues

The basic brain human tissue types are three: Cerebro-Spinal Fluid (CSF), White Matter (WM) and Gray Matter (GM):

→ **Cerebrospinal fluid (CSF).** As referred in [2], liquor cerebrospinalis, is a clear bodily fluid that occupies the subarachnoid space and the ventricular system around and inside the brain and spinal cord. In essence, the brain "floats" in it. The CSF occupies the space between the arachnoid mater (the middle layer of the brain cover, meninges), and the pia mater (the layer of the meninges closest to the brain). It constitutes the content of all intra-cerebral (inside the brain, cerebrum) ventricles,

cisterns, and sulci (singular sulcus), as well as the central canal of the spinal cord.It acts as a "cushion" or buffer for the cortex, providing a basic mechanical and immunological protection to the brain inside the skull.

The CSF is produced at a rate of 500 ml/day. Since the brain can contain only 135 to 150 ml, large amounts are drained primarily into the blood through arachnoid granulations in the superior sagittal sinus. Thus the CSF turns over about 3.7 times a day. This continuous flow into the venous system dilutes the concentration of larger, lipoinsoluble molecules penetrating the brain and CSF.

The CSF contains approximately 0.3% plasma proteins, or approximately 15 to 40 mg/dL, depending on sampling site. CSF pressure ranges from 80 to 100 mmH2O (780–980 Pa or 4.4–7.3 mmHg) in newborns, and < 200 mmH20 (1.94 kPa) in normal children and adults, with most variations due to coughing or internal compression of jugular veins in the neck.

CSF serves four primary purposes:

i. Buoyancy: The actual mass of the human brain is about 1400 grams; however the net weight of the brain suspended in the CSF is equivalent to a mass of 25 grams.The brain therefore exists in neutral buoyancy, which allows the brain to maintain its density without being impaired by its own weight, which would cut off blood supply and kill neurons in the lower sections without CSF.

ii. Protection: CSF protects the brain tissue from injury when jolted or hit. In certain situations such as auto accidents or sports injuries, the CSF cannot protect the brain from forced contact with the skull case, causing hemorrhaging, brain damage, and sometimes death.

iii. Chemical stability: CSF flows throughout the inner ventricular system in the brain and is absorbed back into the bloodstream, rinsing the metabolic waste from the central nervous system through the blood-brain barrier. This allows for homeostatic regulation of the distribution of neuroendocrine factors, to which slight changes can cause problems or damage to the nervous system. For example, high glycine concentration disrupts temperature and blood pressure control, and high CSF pH causes dizziness and syncope.

iv. Prevention of brain ischemia: The prevention of brain ischemia is made by decreasing the amount of CSF in the limited space inside the skull. This decreases total intracranial pressure and facilitates blood perfusion.

An example of CSF tissue is shown in figure 2.1.

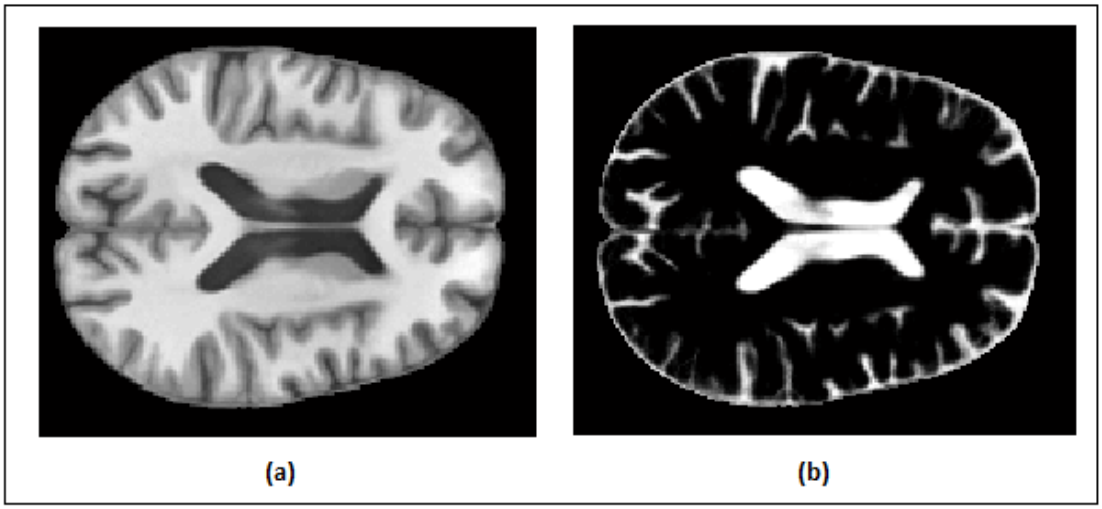


Fig. 2.1 : (a) MRI brain slice (b) CSF tissues, colored in white.

→ **Gray Matter (GM),** as referred in [3], is a major component of the central nervous system (CNS), consisting of neuronal cell bodies, neuropil (dendrites and both myelinated (myelin is a dielectric material that forms a layer, the myelin sheath, usually around only the axon of a neuron.) axons and unmyelinated axons), glial cells (astroglia and oligodendrocytes) and capillaries. Gray matter contains neural cell bodies, in contrast to white matter, which does not and mostly contains myelinated axon tracts. The color difference arises mainly from the whiteness of myelin. In living tissue, gray matter actually has a gray-brown color which comes from capillary blood vessels and neuronal cell bodies.

Gray matter is distributed at the surface of the cerebral hemispheres (cerebral cortex) and of the cerebellum (cerebellar cortex), as well as in the depths of the cerebrum (thalamus; hypothalamus; subthalamus, basal ganglia - putamen, globus pallidus, nucleus accumbens; septal nuclei), cerebellar (deep cerebellar nuclei - dentate nucleus, globose nucleus, emboliform nucleus, fastigial nucleus), brainstem (substantia nigra, red nucleus, olivary nuclei, cranial nerve nuclei) and spinal grey matter (anterior horn, lateral horn, posterior horn).

The function of grey matter is to route sensory or motor stimulus to interneurons of the CNS in order to create a response to the stimulus through chemical synapse activity. Grey matter structures (cortex, deep nuclei) process information originating in the sensory organs or in other grey matter regions. This information is conveyed via specialized nerve cell extensions (long axons), which form the bulk of the cerebral, cerebellar, and spinal white matter. An example of GM tissue is shown in figure 2.2.

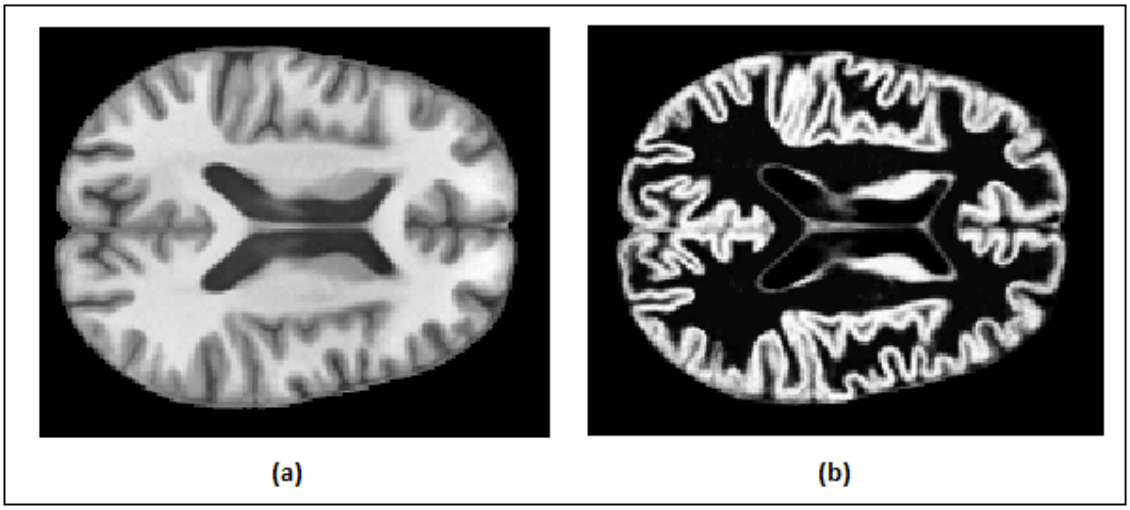


Fig. 2.2 : (a) MRI brain slice (b) GM tissues, colored in white.

→ **White Matter (WM)** as mentioned in [4] is one of the two components of the central nervous system and consists mostly of myelinated axons. White matter tissue of the freshly cut brain appears pinkish white to the naked eye because myelin is composed largely of lipid tissue veined with capillaries. Its white color is due to its usual preservation in formaldehyde. A 20 year-old male has around 176 km of myelinated axons in his brain.

White matter is composed of bundles of myelinated nerve cell processes (or axons), which connect various grey matter areas (the locations of nerve cell bodies) of the brain to each other, and carry nerve impulses between neurons.

In the cerebral hemispheres two types of myelinated axons are identified: short-distance (10 –30 mm) fibers below the gray matter that follow its contours, and long distance (30–170 mm) fibers that are bundled into fasciculi in the deep white matter. There are also shorter intracortical (1–3 mm) unmyelinated fibers within the grey matter.

The total number of long range fibers within a cerebral hemisphere is 2% of the total number of cortico-cortical fibers and is roughly the same number as those that communicate between the two hemispheres in Corpus callosum. Using a computer network as an analogy, the gray matter can be thought of as the actual computers themselves, whereas the white matter represents the network cables connecting the computers together. Myelin, which surrounds the nerve fibers, is found in almost all long nerve fibers, and acts as an electrical insulation. This is important because it allows the messages to pass quickly from place to place.

The brain in general (and especially a child's brain) can adapt to white-matter damage by finding alternative routes that bypass the damaged white-matter areas, and can therefore maintain good connections between the various areas of gray matter.

Unlike gray matter, which peaks in development in a person's twenties, the white matter continues to develop, and peaks in middle age (Sowell et al., 2003)

A 2009 paper by Jan Scholz and colleagues [5] used diffusion tensor imaging (DTI) to demonstrate changes in white matter volume as a result of learning a new motor task (juggling). The study is important as the first paper to correlate motor learning with white matter changes. Previously, many researchers had considered this type of learning to be exclusively mediated by dendrites, which are not present in white matter. The authors suggest that electrical activity in axons may regulate myelination in axons. Similarly, the cause may be gross changes in the diameter or packing density of the axon.

In figure 2.3 is demonstrated an example of WM tissue.

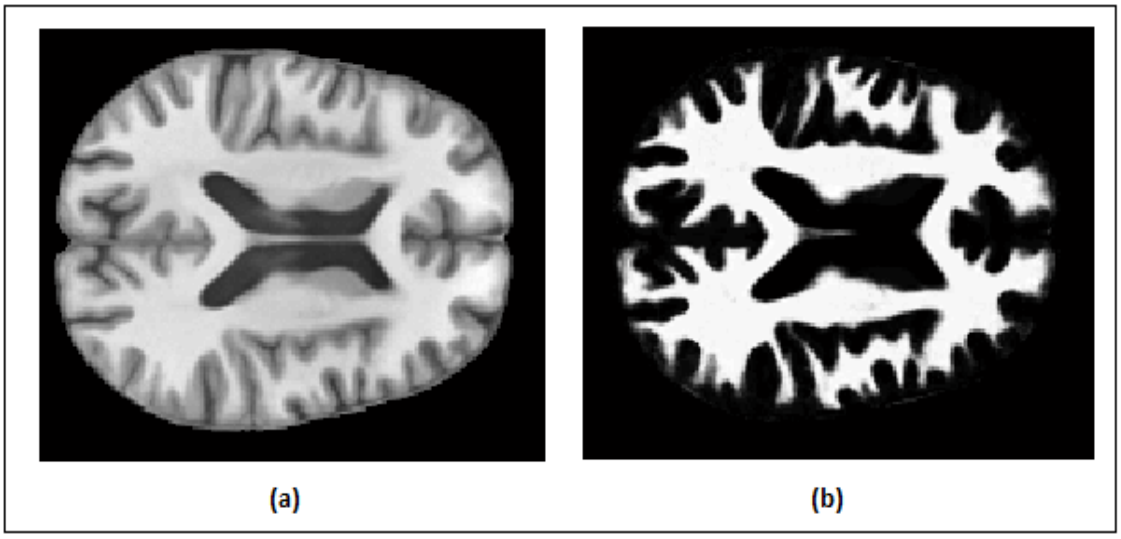


Fig 2.3: (a) MRI brain slice (b) WM tissues, colored in white.

2.3 Magnetic resonance imaging (MRI)

2.3.1 Introduction

Magnetic resonance imaging (MRI), or nuclear magnetic resonance imaging (NMRI), as mentioned in [6] is primarily a medical imaging technique used in radiology to visualize detailed internal structure and limited function of the body. MRI provides much greater contrast between the different soft tissues of the body than Computed Tomography (CT) ([7]) does, making it especially useful in neurological (brain), musculoskeletal, cardiovascular, and oncological (cancer) imaging. Unlike CT ([7]), it uses no ionizing radiation, but uses a powerful magnetic field to align the nuclear magnetization of (usually) hydrogen atoms in water in the body. Radio frequency (RF) fields are used to systematically alter the alignment of this magnetization. This causes the hydrogen nuclei to produce a rotating magnetic field detectable by the scanner. This signal can be manipulated by additional magnetic fields to build up enough information to construct an image of the body [8] .

Magnetic resonance imaging is a relatively new technology. The first MR image was published in 1973 and the first cross-sectional image of a living mouse was published in January 1974.The first studies performed on humans were published in 1977. By comparison, the first human X-ray image was taken in 1895.

Magnetic resonance imaging was developed from knowledge gained in the study of nuclear magnetic resonance. In its early years the technique was referred to as nuclear magnetic resonance imaging (NMRI). However, because the word nuclear was associated in the public mind with ionizing radiation exposure it is generally now referred to simply as MRI. Scientists still use the term NMRI when discussing non-medical devices operating on the same principles. The term magnetic resonance tomography (MRT) is also sometimes used. An example of MRI brain slice image is shown in figure 2.4.

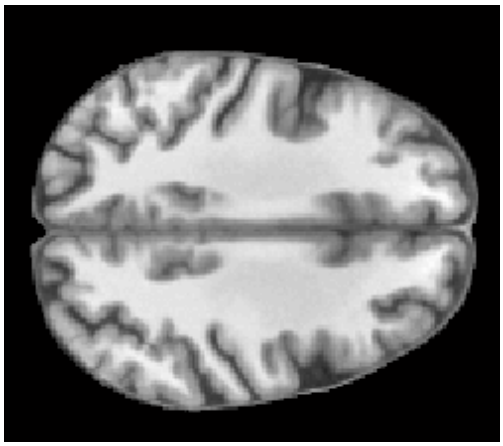


Fig. 2.4: An MRI brain slice

2.3.2 How MRI works

A magnetic resonance imaging instrument (MRI scanner), or "nuclear magnetic resonance (NMR) imaging" scanner as it was originally known, uses powerful magnets to polarise and excite hydrogen nuclei (single proton) in water molecules in human tissue, producing a detectable signal which is spatially encoded, resulting in images of the body. MRI uses three electromagnetic fields: a very strong (on the order of units of teslas) static magnetic field to polarize the hydrogen nuclei, called the static field; a weaker time-varying (on the order of 1 kHz) field(s) for spatial encoding, called the gradient field(s); and a weak radio-frequency (RF) field for manipulation of the hydrogen nuclei to produce measurable signals, collected through an RF antenna.

Like CT , MRI traditionally creates a two dimensional image of a thin "slice" of the body and is therefore considered a tomographic imaging technique. Modern MRI instruments are capable of producing images in the form of 3D blocks, which may be considered a generalisation of the single-slice, tomographic, concept. Unlike CT, MRI does not involve the use of ionizing radiation and is therefore not associated with the same health hazards. For example, because MRI has only been in use since the early 1980s, there are no known long-term effects of exposure to strong static fields (this is the subject of some debate; see 'Safety' in MRI) and therefore there is no limit to the number of scans to which an individual can be subjected, in contrast with X-ray and CT. However, there are well-identified health risks associated with tissue heating from exposure to the RF field and the presence of implanted devices in the body, such as pace makers. These risks are strictly controlled as part of the design of the instrument and the scanning protocol used.

Because CT and MRI are sensitive to different tissue properties, the appearance of the images obtained with the two techniques differ markedly. In CT, X-rays must be blocked by some form of dense tissue to create an image, so the image quality when looking at soft tissues will be poor. In MRI, any nucleus with a net nuclear spin can be used, but the proton of the hydrogen atom remains the most widely used, especially in the clinical setting, because it is so ubiquitous and returns a large signal. This nucleus, present in water molecules, allows the excellent soft-tissue contrast achievable with MRI.

2.3.3 Basic MRI scans

The basic MRI scans are the following :

T1-weighted MRI

T1-weighted scans use a gradient echo (GRE) sequence [9], with short echo time (TE) (The echo time represents the time in milliseconds between the application of the 90° pulse and the peak of the echo signal in spin echo and inversion recovery pulse sequences [10].) and short repetition time (TR) (The amount of time that exists between successive pulse sequences applied to the same slice [11].). This is one of the basic types of MR contrast and is a commonly run clinical scan. The T1 weighting can be increased (improving contrast) with the use of an inversion pulse as in an MP-RAGE sequence. Due to the short TR this scan can be run very fast allowing the collection of high resolution 3D datasets. A T1 reducing gadolinium contrast agent is also commonly used, with a T1 scan being collected before and after administration of contrast agent to compare the difference. In the brain T1-weighted scans provide good gray matter/white matter contrast. An example of T1-weighted MRI scan is shown in figure 2.4(a).

T2-weighted MRI

T2-weighted scans use a spin echo (SE) sequence [12], with long TE and long TR. The difference

of T1 and T2 modalities according to times TE and TR is shown in figure 2.5. They have long been the clinical workhorse as the spin echo sequence is less susceptible to inhomogeneities in the magnetic field. They are particularly well suited to edema as they are sensitive to water content (edema is characterized by increased water content). An example of T2-weighted MRI scan is shown in figure 2.6(b).

T2* -weighted MRI

T2* (pronounced "T2 star") weighted scans use a gradient echo (GRE) sequence, with long TE and long TR. The gradient echo sequence used does not have the extra refocusing pulse used in spin echo so it is subject to additional losses above the normal T2 decay (referred to as T2′), these taken together are called T2*. This also makes it more prone to susceptibility losses at air/tissue boundaries, but can increase contrast for certain types of tissue, such as venous blood. An MRI T2*-weighted slice is imagined in figure 2.6 (c).

T2 Flair MRI

T2 Flair modality is similar to T2 except the signal from cerebrospinal fluid (CSF) has been suppressed, so we no longer confuse lesions that are T2-bright from CSF which is also T2-bright. These are the best sequences for a quick look for pathology. For example, it can be used in brain imaging to suppress cerebrospinal fluid (CSF) so as to bring out the periventricular hyperintense lesions, such as multiple sclerosis (MS) plaques. By carefully choosing the inversion time TI (the time between the inversion and excitation pulses), the signal from any particular tissue can be suppressed. In figure 2.6 (d) it is shown an MRI T2 Flair brain slice.

Spin density weighted MRI

Spin density, also called proton density (Pd), weighted scans try to have no contrast from either T2 or T1 decay, the only signal change coming from differences in the amount of available spins (hydrogen nuclei in water). It uses a spin echo or sometimes a gradient echo sequence, with short TE and long TR. It should be mentioned that Pd-weighted modality is not used as much any more, since FLAIR and other sequences have eliminated the need. Occasionally it can be used in joints display. An example of Pd MRI scan is shown in figure 2.6 (e).

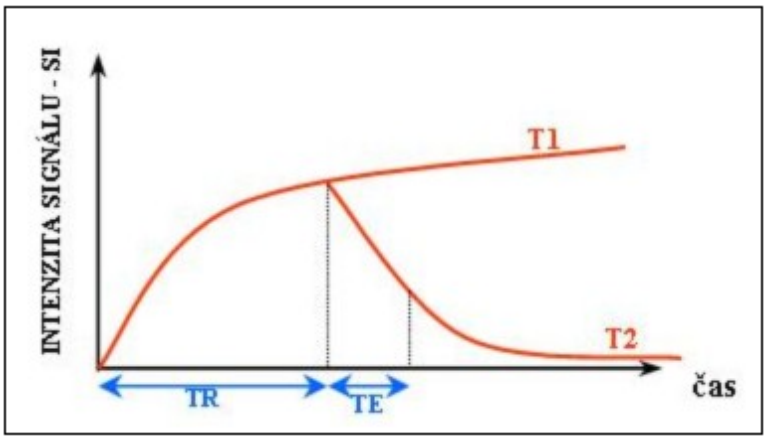


Fig. 2.5: Effects of TR, TE, on formation of T1 and T2 modalities [6].

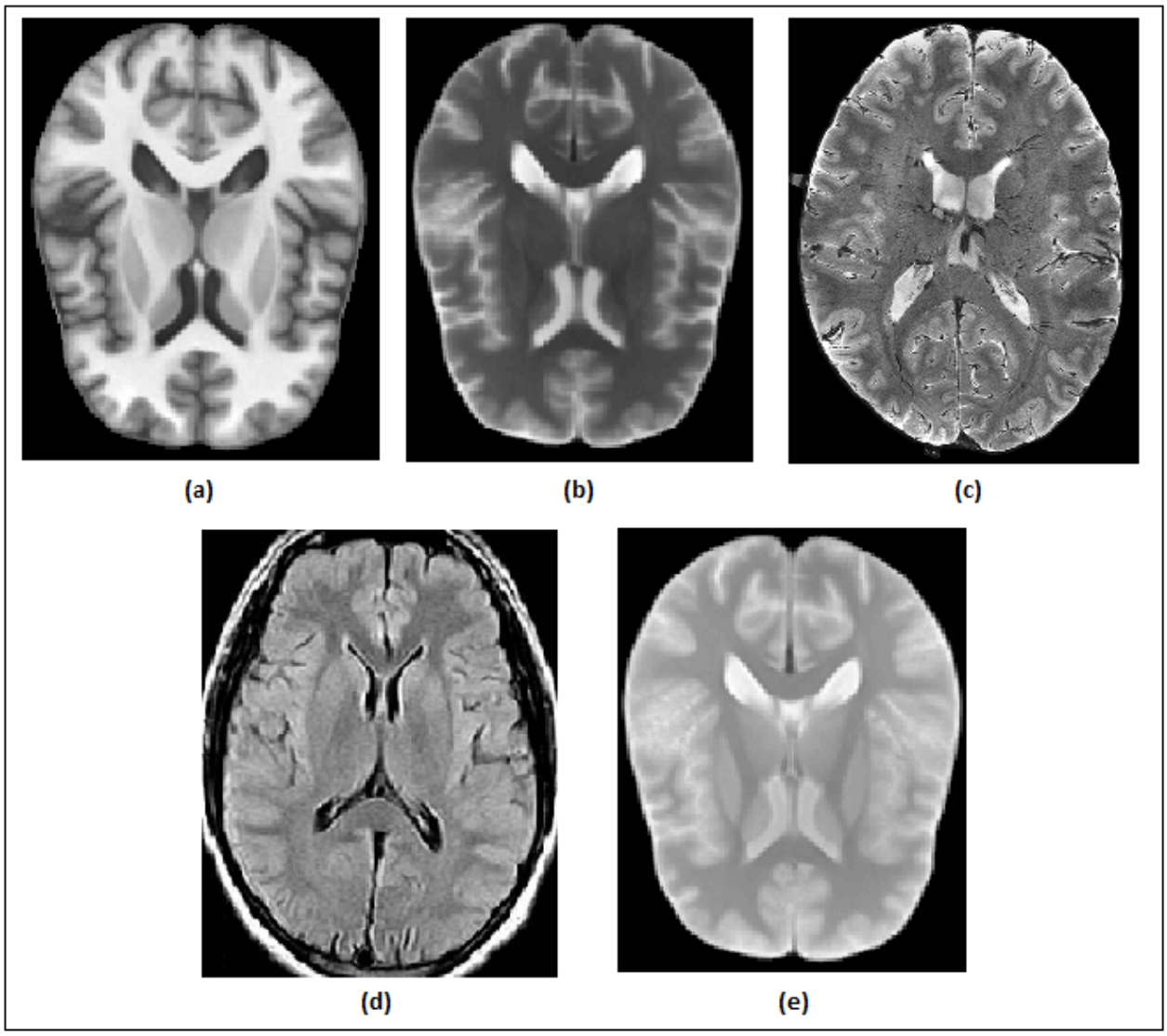


Fig. 2.6: Five basic MRI scans: (a) T1-weighted , (b) T2-weighted , (c) T2*-weighted , (d) T2 Flair and (e) Pd-weighted

Furthermore there are several Specialized MRI scans such as Diffusion MRI, which its principal use is in the imaging of white matter where the location, orientation, and anisotropy of the tracts can be measured, but has also plenty other usages such as applications in the characterization of skeletal and cardiac muscle, Magnetization Transfer (MT) MRI, which can be used to provide an alternative contrast method in addition to T1, T2, and PD, the extension of MT, the magnetization transfer ratio, has been used in neuroradiology to highlight abnormalities in brain structures, Fluid attenuated inversion recovery (FLAIR), which is a pulse sequence which use an inversion recovery technique that nulls fluids and can be used in brain imaging to suppress cerebrospinal fluid (CSF) effects on the image, so as to bring out the periventricular hyperintense lesions, such as multiple sclerosis (MS) plaques, etc but only the basic scans are used in this work [6].

2.3.4 MRI versus CT

A computed tomography (CT) scanner uses X-rays, a type of ionizing radiation, to acquire its images, making it a good tool for examining tissue composed of elements of a higher atomic number than the tissue surrounding them, such as bone and calcifications (calcium based) within the body (carbon based flesh), or of structures (vessels, bowel). MRI, on the other hand, uses non-ionizing radio frequency (RF) signals to acquire its images and is best suited for non-calcified tissue, though MR ima ges can also be acquired from bones and teeth as well as fossils.

CT may be enhanced by use of contrast agents containing elements of a higher atomic number than the surrounding flesh such as iodine or barium. Contrast agents for MRI are those which have paramagnetic properties, e.g. gadolinium and manganese.

Both CT and MRI scanners are able to generate multiple two-dimensional cross-sections (slices) of tissue and three-dimensional reconstructions. Unlike CT, which uses only X-ray attenuation to generate image contrast, MRI has a long list of properties that may be used to generate image contrast. By variation of scanning parameters, tissue contrast can be altered and enhanced in various ways to detect different features.

MRI can generate cross-sectional images in any plane (including oblique planes). In the past, CT was limited to acquiring images in the axial (or near axial) plane. The scans used to be called Computed Axial Tomography scans (CAT scans). However, the development of multi-detector CT scanners with near-isotropic resolution, allows the CT scanner to produce data that can be retrospectively reconstructed in any plane with minimal loss of image quality.

For purposes of tumor detection and identification in the brain, MRI is generally superior. However, in the case of solid tumors of the abdomen and chest, CT is often preferred due to less motion artifact. Furthermore, CT usually is more widely available, faster, less expensive, and may be less likely to require the person to be sedated or anesthetized.

MRI is also best suited for cases when a patient is to undergo the exam several times successively in the short term, because, unlike CT, it does not expose the patient to the hazards of ionizing radiation.

2.4 Brain Tumor

2.4.1 Brain Tumor Definition

A brain tumor, as referred in [13] is an intracranial solid neoplasm, a tumor (defined as an abnormal growth of cells) within the brain or the central spinal canal.

Brain tumors include all tumors inside the cranium or in the central spinal canal. They are created by an abnormal and uncontrolled cell division, normally either in the brain itself (neurons, glial cells (astrocytes, oligodendrocytes, ependymal cells, myelin-producing Schwann cells), lymphatic tissue, blood vessels), in the cranial nerves, in the brain envelopes (meninges), skull, pituitary and pineal gland, or spread from cancers primarily located in other organs (metastatic tumors).

Any brain tumor is inherently serious and life-threatening because of its invasive and infiltrative character in the limited space of the intracranial cavity. However, brain tumors (even malignant ones) do not automatically cause death. Brain tumors or intracranial neoplasms can be cancerous (malignant) or non-cancerous (benign); however, the definitions of malignant or benign neoplasms

differs from those commonly used in other types of cancerous or non-cancerous neoplasms in the body. Its threat level depends on the combination of factors like the type of tumor, its location, its size and its state of development. Because the brain is well protected by the skull, the early detection of a brain tumor only occurs when diagnostic tools are directed at the intracranial cavity. Usually detection occurs in advanced stages when the presence of the tumor has side effects that cause unexplained symptoms.

Primary (true) brain tumors are commonly located in the posterior cranial fossa in children and in the anterior two-thirds of the cerebral hemispheres in adults, although they can affect any part of the brain.

2.4.2 Taxonomy of brain tumors

→ By location and origin of the neoplasm

Primary brain tumors

Primary neoplasms of the brain are tumors that originate in the intracranial sphere or the central spinal canal, based on the organic tissues that make up the brain and the spinal cord. From the brain-lemma we can learn a lot of things about the composition of the brain from different types of organic tissues. For the purpose of this work we will discuss only some types.

- The brain itself is composed of neurons and glia (that function primarily as the physical support for neurons). The neuron itself is rarely the basis for a tumor, though tumors of the glial cells are glioma and often are of the cancerous type.
- The brain is surrounded by a system of connective tissue membranes called meninges that separate the skull from the brain. Tumors of the meninges are meningioma and are often benign neoplasms.
- Below the brain is pituitary and pineal gland which could be the basis for its own -albeit rare- kind of benign glandular neoplasms.

Secondary brain tumors

Secondary tumors of the brain are metastatic tumors that invaded the intracranial sphere from cancers primarily located in other organs. This means that a (malignant) cancerous neoplasm has developed in another organ elsewhere in the body and that cancer cells leak from that primary tumor. The leaked cells enter the lymphatic system and blood vessels, circulate through the bloodstream, and are deposited (strand in the small blood vessels in the brain) within normal tissue elsewhere in the body, in this case in the brain. There these cells continue growing & dividing and become another invasive neoplasm of the primary cancers tissue. Secondary tumors of the brain are very common in the terminal phases of patients with an incurable metastased cancer , most common types of cancers that bring about secondary tumors of the brain are lung cancer, breast cancer and malignant melanoma (skin cancer), kidney cancer and cancer of the colon (in decreasing order of frequency).

Unfortunately enough this is the most common cause of neoplasms in the intracranial cavity.

The skull bone structure can also be subject to a neoplasm that by it very nature reduces the volume of the intracranial cavity, and can damage the brain.

→ By behavior of the neoplasm

Brain tumors or intracanial neoplasms can be cancerous (malignant) or non-cancerous (benign). However, the definitions of malignant or benign neoplasms differs from those commonly used in other types of cancerous or non-cancerous neoplasms in the body. In ordinary cancers (elsewhere in the body) three malignant properties differentiate benign tumors from malignant forms of cancer: benign tumors are self-limited and do not invade or metastasize. The malignant characteristics of tumors are:

- Uncontrolled mitosis (growth by division beyond the normal limits)
- Anaplasia. Is a term to explain that the cells in the neoplasm have an obvious different form (in size and shape). Anaplastic cells display marked pleomorphism. The cell nuclei are characteristically extremely hyperchromatic (darkly stained) and enlarged; the nucleus might have the same size as the cytoplasm of the cell (nuclear-cytoplasmic ratio may approach 1:1, instead of the normal 1:4 or 1:6 ratio). Giant cells that are considerably larger than their neighbors may be formed and possess either one enormous nucleus or several nuclei (syncytia). Anaplastic nuclei are variable and bizarre in size and shape.
- Invasion or infiltration: in medical literature these terms are used as synonymous equivalents. However for clarity in the articles that follow we will adhere to a convention that they mean slightly different things (so readers should be aware that this convention is not kept outside these articles):
- Invasion or invasiveness is the spatial expansion of the tumor through the uncontrolled mitosis, in the sense that the neoplasm invades the space occupied by adjacent tissue, thereby pushing the other tissue aside and eventually compressing the tissue. Often these tumors are associated with clearly outlined tumors in imaging.
- Inflitration is the behavior of the tumor either to grow (microscopic) tentacles that push into the surrounding tissue (often making the outline of the tumor undefined or diffuse) or to have tumor cells "seeded" into the tissue beyond the circumference of the tumorous mass; this doesn't mean that an infiltrative tumor doesn't take up space or doesn't compress the surrounding tissue as it grows, but an infiltrating neoplasm makes it difficult to say where the tumor ends and the healthy tissue starts.
- Metastasis (spread to other locations in the body via lymph or blood).

Of the above malignant characteristics, some elements don't apply to primary neoplasms of the brain :

- Primary brain tumors rarely metastasize to other organs; some forms of primary brain tumors can metastasize but will not spread outside the intracranial cavity or the central spinal canal. Due to the blood-brain barrier cancerous cells of a primary neoplasm cannot enter the bloodstream and get carried to another location in the body. (Occasional isolated case reports suggest spread of certain brain tumors outside the central nervous system, e.g. bone metastasis of glioblastoma multiforme [14].)
- Primary brain tumors generally are invasive (i.e. they will expand spatially and intrude into the space occupied by other brain tissue and compress those brain tissues), however some of the more malignant primary brain tumors will infiltrate the surrounding tissue.

Of numerous grading systems in use for the classification of tumor of the central nervous system, the World Health Organization (WHO) grading system is commonly used for astrocytoma (neoplasms of the brain that originate in a particular kind of glial-cells: the star-shaped brain cells called astrocytes. This type of tumor doesn't usually spread outside the brain and spinal cord and it doesn't usually affect other organs. Astrocytomas are the most common glioma, and can occur in most parts of the brain and occasionally in the spinal cord [15].). Established in 1993 in an effort to

eliminate confusion regarding diagnoses, the WHO system established a four-tiered histologic grading guideline for astrocytomas that assigns a grade from 1 to 4, with 1 being the least aggressive and 4 being the most aggressive.

2.4.4 Characteristics of tumors

The characteristics of tumor allow pathologists to determine how dangerous a tumor is/was for the patient, how it will evolve and it will allow the medical team to determine the therapeutic plan for the patient.

Anaplasia: or dedifferentiation; loss of differentiation of cells and of their orientation to one another and blood vessels, a characteristic of anaplastic tumor tissue. Anaplastic cells have lost total control of their normal functions and many have deteriorated cell structures. Anaplastic cells often have abnormally high nuclear-to-cytoplasmic ratios, and many are multinucleated. Additionally, the nuclei of anaplastic cells are usually unnaturally shaped or oversized nuclei. Cells can become anaplastic in two ways: neoplastic tumor cells can dedifferentiate to become anaplasias (the dedifferentiation causes the cells to lose all of their normal structure/function), or cancer stem cells can increase in their capacity to multiply (i.e., uncontrollable growth due to failure of differentiation).

Atypia: is an indication of abnormality of a cell (which may be indicative for malignancy). Significance of the abnormality is highly dependent on context.

Neoplasia: is the (uncontrolled division) of cells; as such neoplasia is not problematic but its consequences are: the uncontrolled division of cells means that the mass of a neoplasm increases in size, in a confined space such as the intracranial cavity this quickly becomes problematic because the mass invades the space of the brain pushing it aside, leading to compression of the brain tissue and increased intracranial pressure and destruction of brain parenchyma. Increased Intracranial pressure (ICP) may be attributable to the direct mass effect of the tumor, increased blood volume, or increased cerebrospinal fluid (CSF) volume may in turn have secondary symptoms

Necrosis: is the (premature) death of cells, caused by external factors such as infection, toxin or trauma. Necrotic cells send the wrong chemical signals which prevents phagocytes from disposing of the dead cells, leading to a build up of dead tissue, cell debris an oxins at or near the site of the necrotic cells .

Arterial and venous hypoxia or the deprivation of adequate oxygen supply to certain areas of the brain, this is due to the fact that the tumor taps into nearby bloodvessels for its supply of blood, the neoplasm enters into competition for nutrients with the surrounding braintissue.

More generally a neoplasm may cause release of metabolic end products (e.g., free radicals, altered electrolytes, neurotransmitters), release and recruitment of cellular mediators (e.g., cytokines) that disrupt normal parenchymal function.

2.4.5 Diagnosis

Although there is no specific or singular clinical symptom or sign for any brain tumors, the presence of a combination of symptoms and the lack of corresponding clinical indications of infections might be an indicator to step up the diagnostic investigation to the direction of an intracranial neoplasm.

The diagnosis will often start with an interrogation of the patient to get a clear view of his medical

antecedents, and his current symptoms. Clinical and laboratory investigations will serve to exclude infections as cause of the symptoms. Examinations in this stage may include ophtamological, otolaryngological and/or Electrophysiological exams, other means such as electroencephalography play a role in the diagnosis of brain tumors.

Swelling, or obstructing the passage of cerebrospinal fluid may cause (early) signs of increased intracranial pressure which translates clinically into headaches, vomiting, or an altered state of consciousness, (and in children) changes to the diameter of the skull and bulging of the fontanelles. More complex symptoms such as endocrine dysfunctions should alarm doctors not to exclude brain tumors.

A bilateral temporal visual field defect (due to compression of the optic chiasm) or dilatation of the pupil, and the occurrence of either slowly evolving or the sudden onset of focal neurologic symptoms, such as cognitive and behavioral impairment (including impaired judgment, memory loss, lack of recognition, spatial orientation disorders), personality or emotional changes, hemiparesis, hypoesthesia, aphasia, ataxia, visual field impairment, impaired sense of smell, impaired hearing, facial paralysis, double vision, but also more severe symptoms might accur too such as: tremors, paralysis on one side of the body hemiplegia, but also (epileptic) seizures in a patient with a negative history for epilepsy, impairment to swallow should raise red flags.

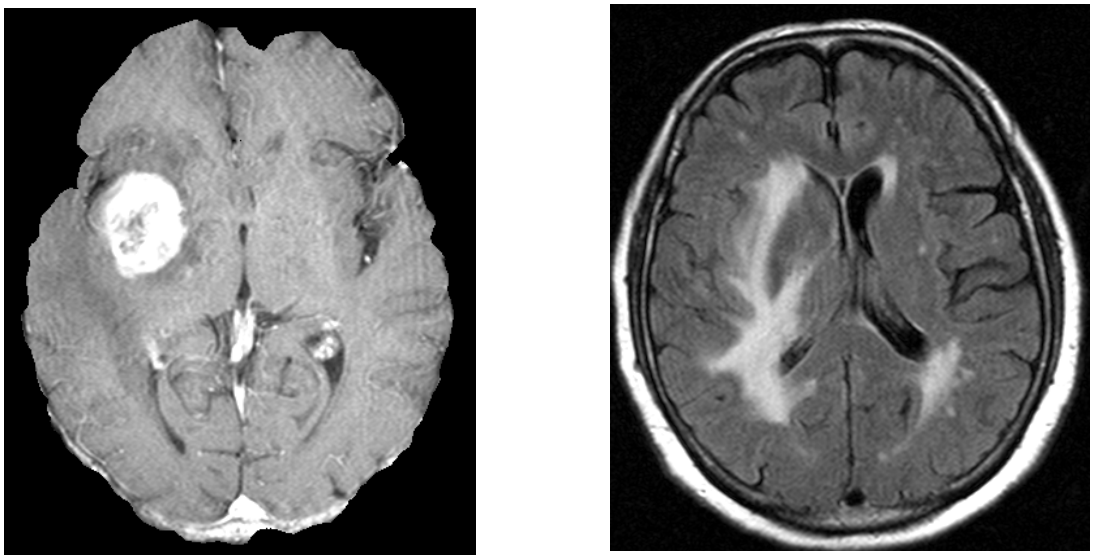
Imaging plays a central role in the diagnosis of brain tumors. Early imaging methods —invasive and sometimes dangerous— such as pneumoencephalography and cerebral angiography, have been abandoned in recent times in favor of non-invasive, high-resolution techniques, such as CT -scans and especially MRI.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

- Benign brain tumors often show up as hypodense (darker than brain tissue) mass lesions on cranial CT-scans. On MRI, they appear either hyperintense (brighter than brain tissue) or isointense (same intensity as brain tissue) on T1-weighted scans, or hyperintense on T2-weighted MRI, although the appearance is variable.An example of T-1 and T-2 brain tumor scan is presented in figure 2.7.
- Contrast agent uptake, sometimes in characteristic patterns, can be demonstrated on either CT or MRI-scans in most malignant primary and metastatic brain tumors.
- Perifocal edema , or pressure-areas, or where the brain tissue has been compressed by an invasive process also appears hyperintense on T2-weighted MRI, they might indicate the presence a diffuse neoplasm (unclear outline).

This is because these tumors disrupt the normal functioning of the blood-brain barrier and lead to an increase in its permeability. However it is not possible to diagnose high versus low grame gliomas based on enhancement pattern alone.

Another possible diagnostic indicator would be neurofibromatosis which can be in type one or type two.

The definitive diagnosis of brain tumor can only be confirmed by histological examination of tumor tissue samples obtained either by means of brain biopsy or open surgery. The histological examination is essential for determining the appropriate treatment and the correct prognosis. This examination, performed by a pathologist, typically has three stages: interoperative examination of fresh tissue, preliminary microscopic examination of prepared tissues, and followup examination of prepared tissues after immunohistochemical staining or genetic analysis.



(a) **(b)**

Fig. 2.7: Brain tumor representation (white area) in (a) T1-weighted MRI scan and (b) T2 Flair-weighted MRI scan

3. BASIC IMAGE PROCESSING AND BRAIN STUCTURES SEGMENTATION CONCEPTS

3.1Histogram

The histogram of a gray scale picture contains useful information about the picture and for this reason is considered to be one of the most valuable tools for the processing of digital images. Histogram is a representation of the distribution of colors (or in gray scale images,the percentage of black and white color). It is actually a graph in which the horizontal axis represents the intensity range (in our case 0 to 4095) and the vertical axis represents the number of photo-elements that contains each intensity respectively. An example of histogram of a gray scale image is shown in figure 3.1 .

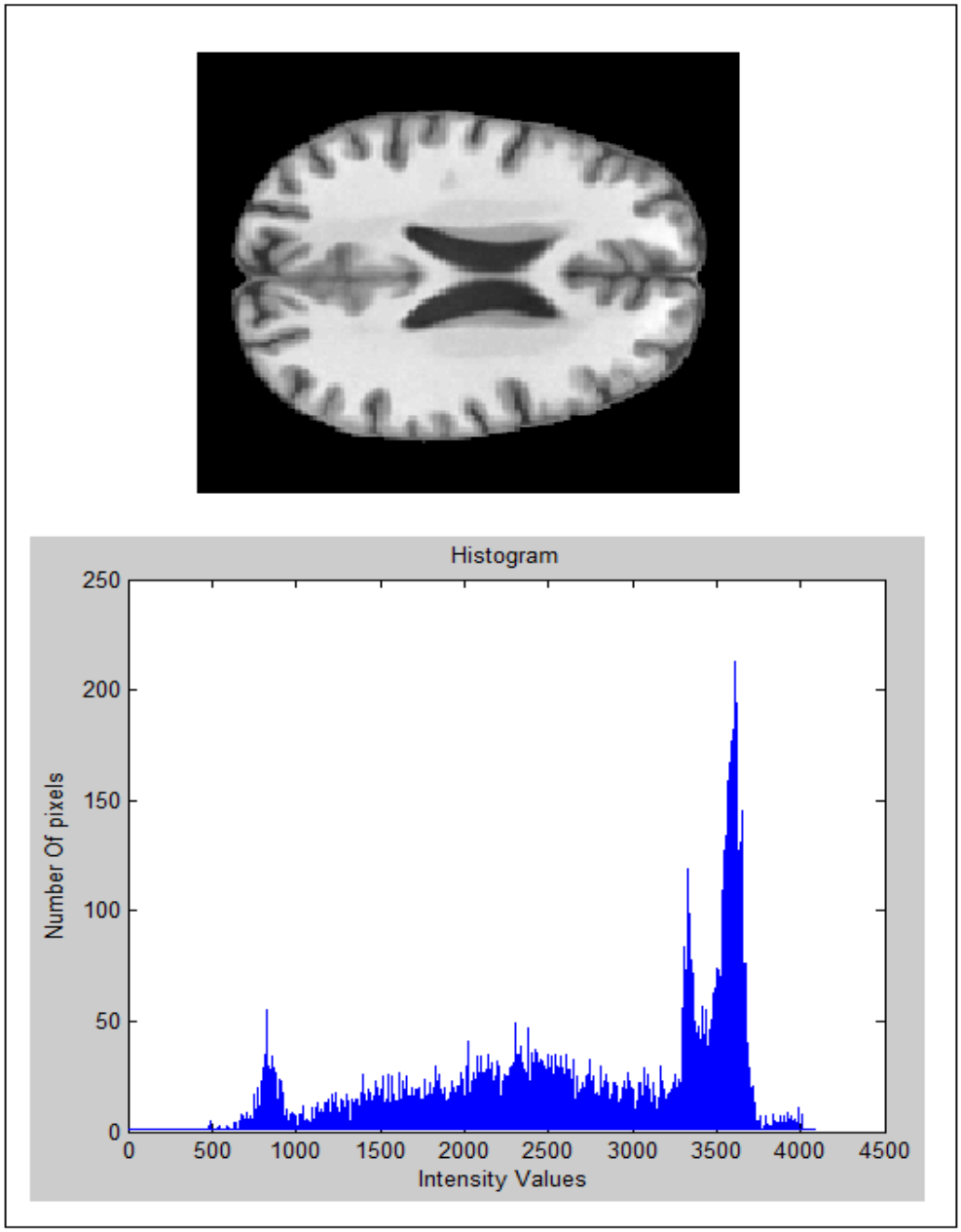


Fig. 3.1 :An MRI gray scale image of a brain slice and the histogram of its brain tissue pixels, normalized to values between 0 and 4095.

Furthermore an image histogram can be used for the optimization of the image, for the modification of it's characteristics, the transformation of the image to an image with shorter density range, the extraction of the image characteristics and numerous other applications [16] .

3.2 Probability Density Function

When the maximum value of histogram is normalized to one, then the produced normalized histogram corresponds to the distribution of the density-probability of the gray-scale levels of the image (in a gray scale image). In figure 3.2 we can observe the Probability Density Function (PDF) of the same image of figure 3.1 :

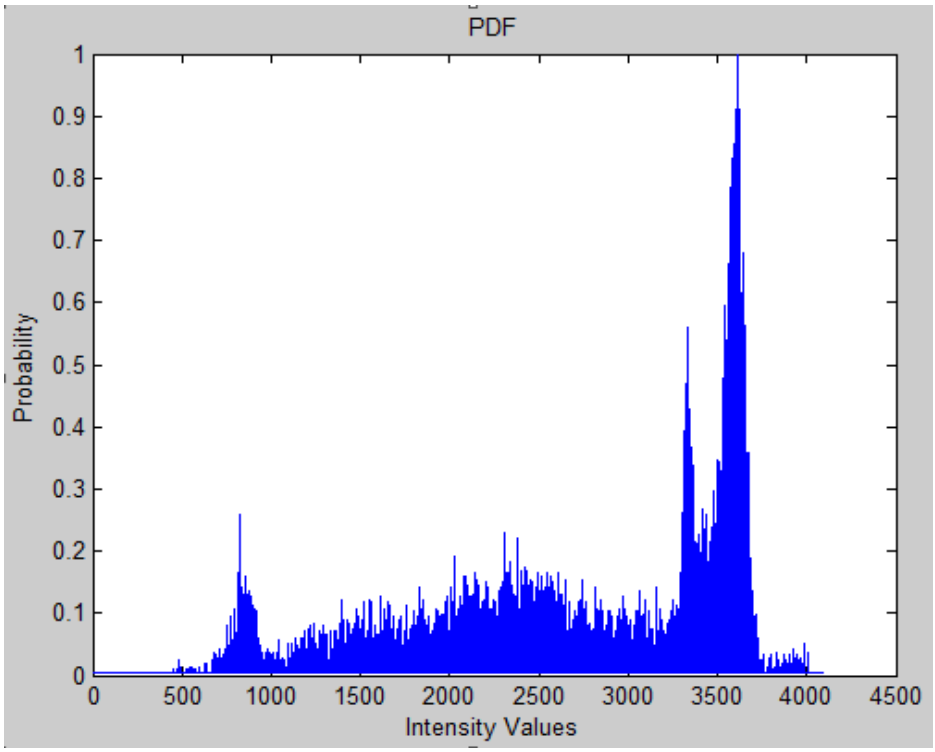


Fig. 3.2: The pdf of the image of figure 3.1

3.3 Density Estimation

In pattern recognition, the difference between parametric and non-parametric segmentation techniques, is that in parametric approaches the forms of the density functions which are formed according to the way that the data are being modeled, are known or at least assumed according to some common parametric forms (for example Gaussian pdf). Although, in most pattern recognition applications this assumption is suspect. These common parametric forms rarely fit the densities actually encountered in practice. Non-parametric applications do not assume any parametric form for the pdf, but instead they estimate them. As mentioned in [17], even though rigorous demonstrations that the estimates for the unknown pdf converge may require considerable care, the basic ideas behind many of the methods of estimating an unknown pdf are very simple. The most fundamental techniques rely on the fact that the probability P that a vector $\mathbf{x}$ will fall in a region R is given by :

$$P = \int_R p(x')\,dx' \qquad \textbf{(3.1)}$$

Thus, it can be claimed that P is a smoothed or in other words, an averaged version of the density function $p(\mathbf{x})$. This smoothed value of p can be estimated, by estimating the probability P. Let us suppose that n samples $x_1, \ldots, x_n$ are drawn independently and identically distributed according to the probability law $p(\mathbf{x})$ (basic probability theory is available in Appendix A). The probability that k of these n fall in region R is given by the binomial law (Appendix A.6):

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \qquad \textbf{(3.2)}$$

and the expected value (Appendix A.3) for k is:

$$E[k] = nP \qquad \textbf{(3.3)}$$

Furthermore, this binomial distribution (Appendix A.6) for k peaks very sharply about the mean, so that it can be expected that the ratio k/n will be a very good estimate for the probability P, and consequently for the smoothed density function. This estimate is especially accurate when n is very large (Fig. 3.3). Assuming that $p(\mathbf{x})$ is continuous and that the region R is so small that p does not vary appreciably within it, it can be written:

$$\int_R p(x')\,dx' \simeq p(x)\,V \qquad \textbf{(3.4)}$$

where x is a point within R and V is the volume enclosed by R. Combining Eqs. 3.1, 3.3 & 3.4, we get the following obvious estimate for $p(\mathbf{x})$:

$$p(x) \simeq \frac{k\,/\,n}{V} \qquad \textbf{(3.5)}$$

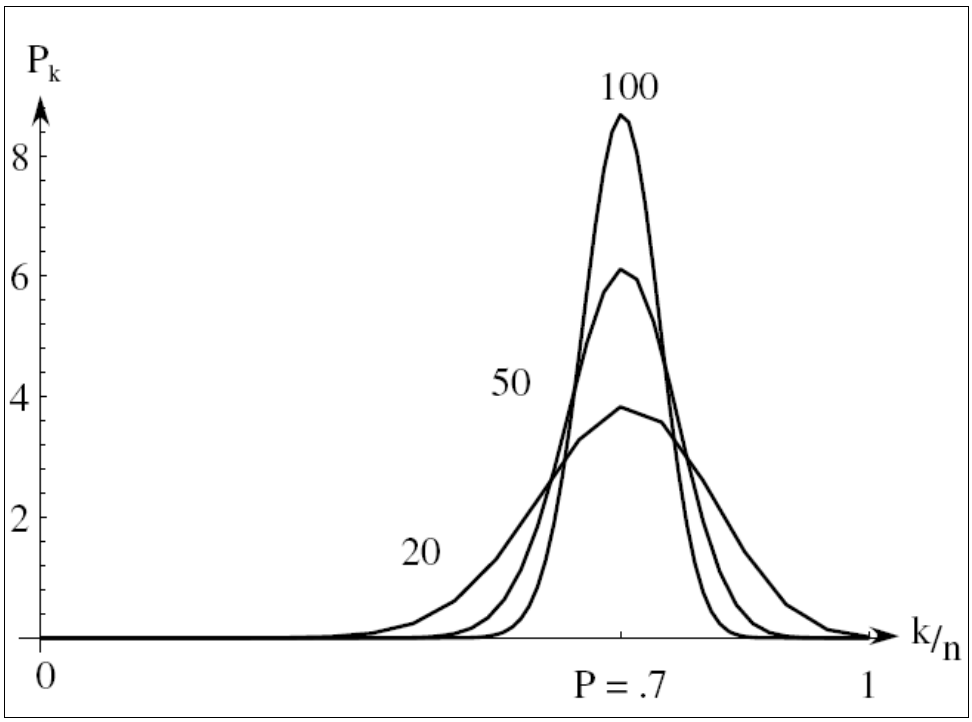


Fig. 3.3: The probability P_k of finding k patterns in a volume where the space averaged probability is P as a function of k/n. Each curve is labelled by the total number of patterns n. For large n, such binomial distributions peak strongly at $k/n = P$ (here chosen to be 0.7) [17].

Although, several problems still remain, some practical and some theoretical. Fixing the volume V and taking more and more training samples, the ratio *k/n* will converge as desired, but we have only obtained an estimate of the space-averaged value of *p(**x**):*

$$\frac{P}{V} = \frac{\int_R p(x')\,dx'}{\int_R dx'} \qquad \textbf{(3.6)}$$

If we want to obtain *p(**x**)* rather than just an averaged version of it, we must let V approach zero. However, by fixing the number n of samples and by letting V approach zero, the region will finally become so small that it will enclose no samples and as a result our estimate $p(x) \neq 0$ will be useless. Or even if by chance one or more of the training samples coincide at **x**, the estimate diverges to infinity, which is equally useless.

Practically, the number of samples available is always limited. Consequently, the volume V can not be allowed to become arbitrarily small. If this kind of estimate is to be used, one will have to accept a certain amount of variance in the ratio *k/n* and a certain amount of averaging of the density *p(**x**)*.

Theoretically, it is interesting to ask how these limitations can be circumvented if an unlimited number of samples is available. Suppose we use the following procedure: In order to estimate the density at ***x***, we form a sequence of regions $R_1, R_2, \ldots$, containing ***x*,** the first region to be used with one sample, the second with two, and so on. Let V_n be the volume of R_n , k_n be the number of samples falling in R_n , and $p_n(x)$ be the *n*-th estimate for *p(**x**)*:

$$p_n(x) = \frac{k_n / n}{V_n} \qquad \textbf{(3.7)}$$

If $p_n(x)$ is to converge to *p(**x**)*, three conditions appear to be required, as discussed in [17]:

- $\lim_{n \to \infty} V_n = 0$

This condition assures us that the space averaged P/V will converge to *p(**x**)*, provided that the regions shrink uniformly and that *p(.)* is continuous at ***x***.

- $\lim_{n \to \infty} k_n = \infty$

This second condition makes sense only if $p(x) \neq 0$ and assures us that the frequency ratio will converge to the probability *P*.

- $\lim_{n \to \infty} k_n / n = 0$

The third condition is clearly necessary if $p_n(x)$ given by Eq. 3.7 is to converge at all. It also says that although a huge number of samples will eventually fall within the small region R_n , they will form a negligibly small fraction of the total number of samples.

In pattern recognition, there are two common ways of obtaining sequences of regions that satisfy these conditions (Fig. 3.4). The first method is the **Parzen-window** approach, explained in Section

4.1. Here, the basic idea is to shrink an initial region by specifying the volume V_n as some function of n, such as for example $V_n = 1/\sqrt{n}$. It then must be shown that the random variables k_n and k_n/n behave properly, or more substantially, that $p_n(x)$ converges to $p(\boldsymbol{x})$. The second method is k-nearest neighbor (k-NN) where k_n is specified as some function of n, such as $k_n = \sqrt{n}$. Here the volume V_n is grown until it encloses k_n neighbors of $\boldsymbol{x}$. Although it is difficult to make meaningful statements about their finite-sample behavior, both of these methods do in fact converge.

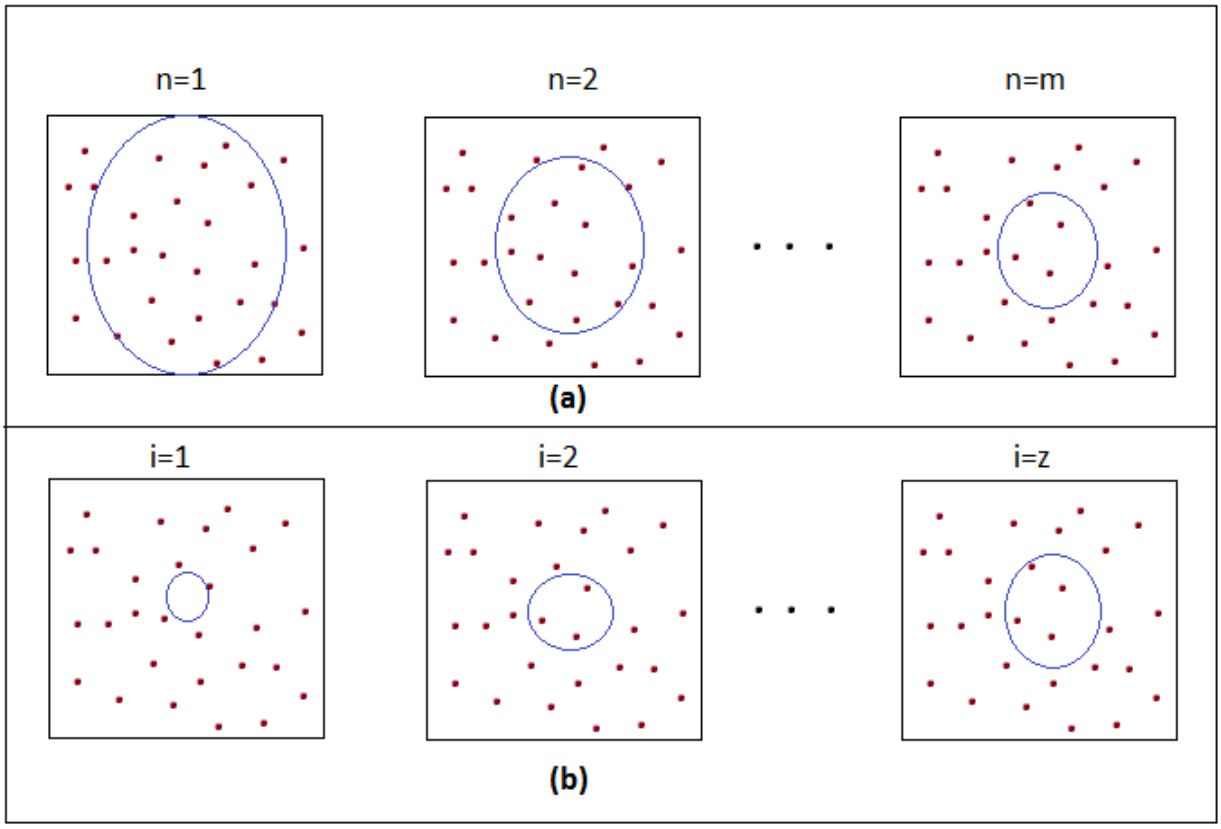


Fig. 3.4: Two methods for estimating the density at a point $\boldsymbol{x}$ (at the center of each square).In (a), it is shown the basic idea of the Parzen-window density estimation method, which converges in step n=m. In (b), it is shown the k-NN estimation method, which converges in step i=z. Of course, m step may be different from z.

3.4 Brain Tissues Segmentation

3.4.1 Introduction

Volumetric analysis of different parts of the brain is extremely useful in assessing the progress of remission of various diseases, such as Alzheimer's disease, epilepsy, multiple sclerosis and schizophrenia. The process of partitioning a biomedical brain image, into three segments, as the number of brain tissues, CSF, GM, WM is called brain tissues segmentation. Segmentation of brain tissues is a challenging problem due to the complexity of the images, as well as to the absence of models of the anatomy that fully capture the possible deformations in each brain. For the segmentation task, numerous methods have been proposed. In Sections 3.4.2-3.4.4 we demonstrate

some basic techniques that were used in the past, some of them even in nowadays, in order to provide a reliable segmentation algorithm.

3.4.2 Segmentation Using Intensity Decision Boundaries

Observing an MRI brain image, the first thing that we notice is that there are differences in intensity values between the three types of tissues. For instance in T1 modality , as shown in figure 3.5a, CSF is represented hypointense, GM in medium intensity values and WM hyperintense. If we take the histogram of this image without the background (figure 3.5b), we should notice that our previous observation was right, as there are three main lobes where all pixels are gathered around, the lobe with intensity value 1200, the lobe with intensity value 2400 and the lobe with intensity value 3500.

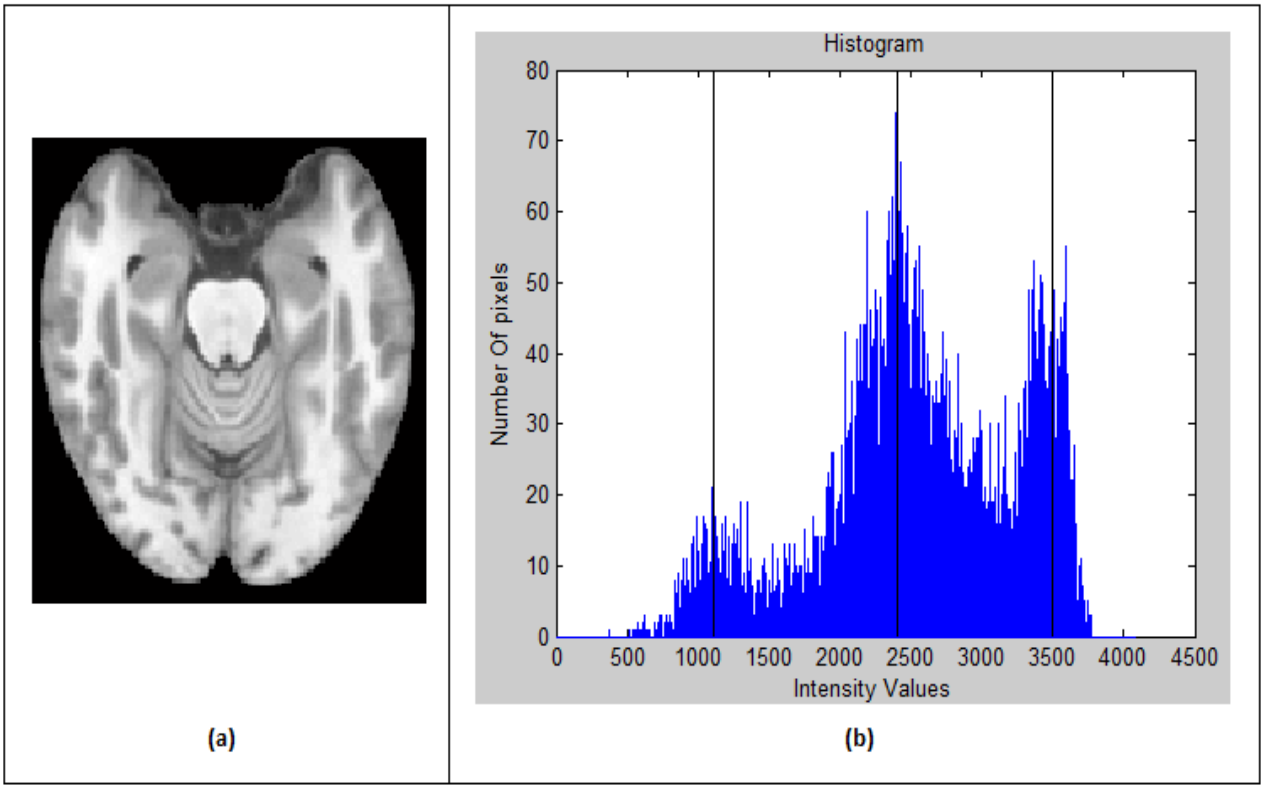


Fig. 3.5: (a) A T1 MRI brain slice (b) The histogram of image (a). This image is normalized to values between 0 and 4095.

According to these observations, can we make a classifier that would segment correctly pixels to the three desired categories? The answer is no and we explain later in this section why. But suppose that we can. Then, using pattern recognition expressions, our *feature vector* would be the intensity of the pixels and consequently our *feature space* is one-dimensional. Our classifier should make a decision how to classify the pixels to the three tissue types. This decision could be whether the pixel's intensity is lower than some critical value a* then the classifier decides that the pixel is CSF, and whether the pixel's intensity exceeds another critical value b* then the classifier decides that the pixel is WM. If none of these categories is decided, then the pixel is classified to the GM category. The values a* and b* are called *decision boundaries*. Back to our example, the decision boundaries can be selected as 1800 and 3000 as shown in figure 3.6.

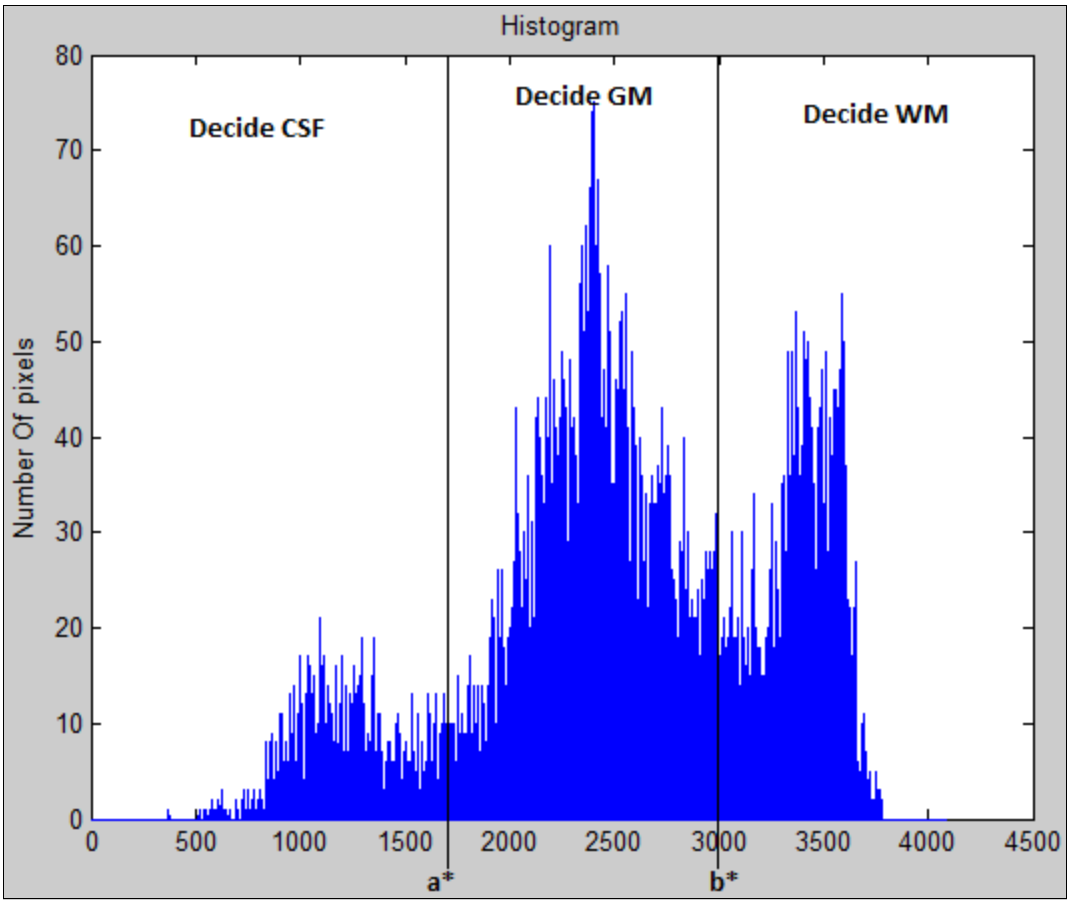


Fig. 3.6: Histogram of the image of figure 3.3 a with decision boundaries a* = 1800 and b* = 3000. In this way, our classifier segments each pixel in CSF, if its intensity is lower than 1800, in WM if its intensity exceeds the 3000 value and GM if its intensity is between these two values.

The decision boundaries, have been selected not in an accurate but in a doubtful way and misclassification of pixels is unavoidable. For example a pixel with intensity value 1700 is not necessarily CSF. Furthermore, these decision boundaries cannot be generalized in all occasions. Take for example the histogram of fig. 3.7 which corresponds to a real MRI T1 image. Obviously, taking decision boundaries 1700 and 3500 the classifier will lead to a completely wrong segmentation result. Brain tissue segmentation should be an accurate and reliable task. In other words, there is raised a question of how much is the overall single *cost* of the segmentation classifier that we use. The true task of every segmentation method is to minimize such a cost. This is the central task of *decision theory* of which pattern recognition is perhaps the most important subfield [18]. Back to our example, undoubtedly the cost of our decision is too high, making this segmentation method unreliable to use. The solution of moving the decision boundaries forward or backward will not solve the problem of making an accurate, reliable and generalized segmentation algorithm.

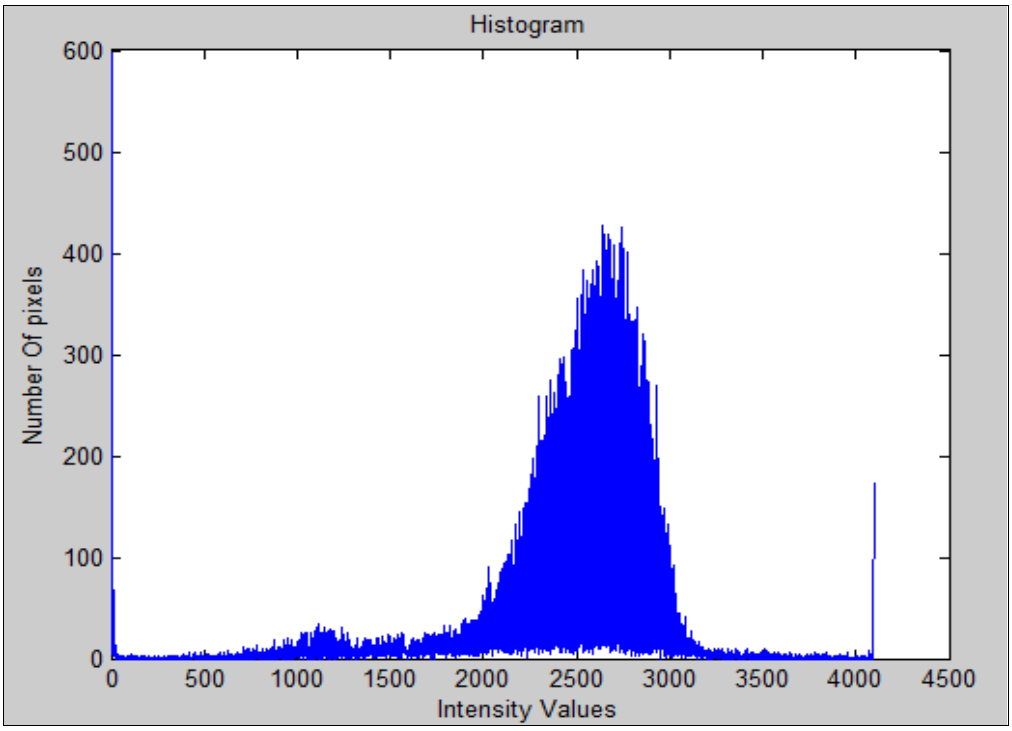


Fig. 3.7: Histogram of brain tissues of a real MRI T1 brain slice, normalized to values 0 to 4095.

3.4.3 Supervised Techniques

Supervised technique is called every brain tissue segmentation approach that uses some labeled sample pixels (or even voxels in some occasions) from each tissue (prototypes), provided by the user, in order to train the classifier properly, so to perform the segmentation task. In order to avoid re-training the classifier for each new scan, supervised methods have to normalize the intensity between the MRI scans, allowing the selection of prototypes and train of the classifier on a reference scan, following which pixels (or voxels) of any other scan to be classified using the same classifier without further human intervention. As a conclusion, because of the fact that human interaction is required, supervised based algorithms are semi-automatic.

In [19] the following approach is used, based on a three-step procedure:

- A conventional k-Nearest Neighbor (k-NN, [20]) classifier is applied to pre-classify the three brain tissue types and Multiple Sclerosis (MS) lesions from a set of prototypes by an expert.
- The classification of problematic patterns is resolved computing a fast distance transformation algorithm from the set of prototypes in the Euclidean space defined by the MRI dataset.
- Finally, a connected component filtering algorithm is used to remove lesion voxels not connected to the real lesions.

To sum up, this supervised, nonparametric technique (nonparametric because the k-NN algorithm does not require any knowledge or assumptions about statistical parameters of the data) can

segment different structures with the same intensity level range. The other principal feature is the high performance achieved due to the fast algorithms to compute distance transformations and Voronoi diagrams [21] on which this method is based on. Furthermore, it shows some advantages with respect to unsupervised methods, because it is fairly stable for the segmentation of abnormal anatomy, and because no image-atlas registration is needed, which is usually a performance bottleneck in other methods. On the other hand, the whole execution time is to be increased around one more minute in order take into account the user interaction to train the classifier. The algorithm shows a high accuracy, depending essentially on the training dataset selected by a medical expert, and it performs really well using one intensity channel compared to segmentations carried out with more than one channel, which is a clear advantage for clinical applications. It is useful for interactive segmentation due to its high performance and the facility to add or remove training prototypes to improve the results. The applications of this method go well beyond MS MRI segmentation since it can be used to segment almost every type of image modalities. Currently it is also started to be used for MRI segmentation of the knee cartilage.

In [22], multi-spectral MR images from various modalities are used, so to increase the feature space and benefit from the new information that is available. In the first step, a clustering algorithm such as K-means or ISODATA (ISODATA algorithm stands for Iterative Self-Organizing Data Analysis Techniques. This is a more sophisticated algorithm which allows the number of clusters to be automatically adjusted during the iteration by merging similar clusters and splitting clusters with large standard deviations [23].) is used. This tends to result in some over-segmentation of the MR image, where the use multi-spectal MR images such as the PD, T1 and T2 images obtain more differentiation of brain tissues. Over-segmentation practically means that at the initial clustering step, the data set is deliberately clustered into a greater number of classes than actually exist. This reduces both the chance and frequency that different objects are clustered into one class. This step is necessary because different objects may be very close in some features and ordinarily tend to be under-segmented. Combining clusters that belong together is much simpler than splitting up those that do not.Then a supervised classification algorithm such as a back-propagation network [24] is used so to provide the final segmentation results. As a conclusion, in [22] they claim that their proposed algorithm is applied successfully to various MR images acquired from MR scanners at different times with different slice thicknesses and fields of view and also that it successfully segments MR images of the brain containing ambiguous boundaries.

As a conclusion, several supervised methods like the mentioned in this section have been proposed, providing decent results but the main weakness of supervised segmentation algorithms is that as they require human interaction in order to provide the initial sample values so as to train the classifier, they are semi-automatic and not fully automatic, making their results not fully objective and reproducible.

3.4.4 Unsupervised Techniques

A common unsupervised approach is the Gaussian Mixture Models (GMM) (A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities [25]). MRI noise is known to follow a Rician PDF, which can be reasonably approximated by a Gaussian PDF. As a consequence, several segmentation methods are based on a GMM within the intensity feature space. In [26], a Gaussian Mixture Model (GMM) is fitted to the voxels intensity using the expectation-maximization (EM) algorithm [27], according to which every voxel is assigned to the tissue class for which it gives the highest probability. The GMM-EM intensity based framework has been refined in [28] and [29] to account for partial volume effects and blood vessel signals that may alter CSF segmentation.

Although, it is widely recognized that using intensity information alone **has proven insufficient for a reliable segmentation algorithm**. Additive noise and multiplicative bias-fields cause local signal perturbations which are responsible for cluster overlaps in the intensity feature space, resulting in poor tissue-class separability. Furthermore, intensity based segmentation methods may give unrealistic results, with tissue-class regions appearing granular, fragmented, or violating anatomical results. Incorporating spatial information provides a means for improving the segmentation results. By appending the spatial position coordinates in 2d dimension [x-y] as in our implemented algorithm or even in 3d dimension [x-y-z], to the intensity features, a higher dimensional feature space is obtained where clusters represents both pixel's (in 2d case) or voxel's (in 3d case) intensity and spatial position distribution. Clusters in the augmented feature space are more closely related to the brain anatomy. This observation may prove problematic for parametric modes such as GMMs that implicitly assume cluster convexity, as the brain anatomy cannot be decomposed into a small number of convex regions in the joint spatial-intensity feature space. A recently published solution includes using a large number of Gaussians per brain tissue, in order to capture the complicated spatial layout of the individual tissues [30].

One common way to incorporate spatial information is the use of statistical atlas which provides the prior probability for each pixel to originate from a particular tissue-class. In [31], each tissue's intensity is modeled by a Parzen density fitted to voxels selected from an affine-registered atlas. Co-registration of the input image and atlas is critical in this scenario [32]. Although, it must be emphasized that an appropriate atlas does not always exist for the data at hand. Two such cases are brain data with pathologies or brain data obtained from young infants.

Another common unsupervised technique that is widely used, in combination with statistical atlas in some occasions, is to model neighboring pixels (or voxels) interactions using a Markov Random Field (MRF) statistical spatial model. The MRF is a stochastic process that specifies the local characteristics of an image and is combined with the given data to reconstruct the true image. The MRF of prior contextual information is a powerful method for modeling spatial continuity and other features and even simple modeling of this type can provide useful information for the segmentation process and improve segmentation smoothness. The MRF itself is a conditional probability model, where the probability of a pixel (or voxel) depends on its neighborhood. It is equivalent to a Gibbs joint probability distribution [33] determined by an energy function. This energy function is a more convenient and natural mechanism for modeling contextual information than the local conditional probabilities of the MRF. The MRF on the other hand is the appropriate method to sample the probability distribution. In [34] the distribution of tissue intensities are described by Parzen-window statistics [35] and both neighborhood tissue correlations and signal inhomogeneities are modeled by a priori MRF, leading to an accurate and robust segmentation with respect to noise, inhomogeneities, and structure thickness. In [36] a combination of Hidden MRF and EM techniques are used so to provide a decent segmentation algorithm. On the other hand, the main disadvantage of MRF based segmentation algorithms is that they are computationally intensive, requiring critical parameter settings. It is possible to use MRF with predefined settings, which is faster but possibly less accurate.

An alternative to statistical parametric approaches is the use of unsupervised nonparametric schemes. One such approach is the mean-shift algorithm [37]. Here, adaptive gradient ascent is used to detect local maximum of data density in feature space. Data points are associated with local maximum, or modes, thereby defining the clusters. Key characteristics of the mean-shift algorithm include the fact that no initial cluster positions are required, as well as the fact that the final number of extracted clusters is a result of the algorithm. A detailed description of the mean-shift algorithm

is provided in Section 5. In recent years, mean-shift technique has been used for image segmentation, object tracking and medical image analysis applications [38-40]. In [41] a mean-shift based, segmentation algorithm for brain tissues is proposed, which is the basis of our mean-shift implemented algorithm, providing a simple, full automated, accurate and reliable segmentation method.

A comparison of the proposed unsupervised, nonparametric mean-shift algorithm with an MRF and GMM implementation is available in the Experimental Section.

3.5 Brain Tumor Modeling

3.5.1 Brain Tumor Modeling Challenges

The rapid advent of MRI scanning protocols gave the opportunity of accurate follow-up of tumor growth through volumetric measurements, as referred in [42]. Accurate brain tumor modeling appears to be of uttermost importance for therapeutic management, especially for low-grade glioma. During the low-grade phase, patients in most occasions are asymptomatic and the tumor evolution can only be monitored by MRI. Unfortunately, such information provided by MRI, is usually not fully integrated with the therapeutic strategy and as a result, assessment of tumor evolution is still limited to qualitative descriptions including recurrence, progression, regression and stability. Thus, bio-mathematical models are expected to help in tumor modeling, simulation of treatment effects and eventually in optimization of therapeutic strategies.

It must be emphasized that computational models of gliomas dynamics have been initiated more than ten years ago [43,44]. At first, studies were focused on modeling the effect of chemotherapy and surgical resection on the evolution of high-grade gliomas. The mathematical framework introduced at that time may be still in use, but undoubtedly there have been considerable advances in its numerical resolution. In particular, digital brain templates, provided by MRI, enable to implement the biophysics equations onto accurate virtual anatomy. This in turn allows to refine the model, by introducing for example different cell motility in white and gray matter [45], and inside white matter, along and orthogonally to axonal fasciculus [46]. However, published studies have never seriously matched observed radiological evolution with virtual in silico dynamics. Such a comparison would require three different steps: segmentation of actual growth, registration on a virtual brain atlas, and identification of model parameters corresponding to optimal matching between actual and simulated evolution.

Full 3D segmentation on digital MRI images is required, in order to obtain an accurate determination of the actual tumor evolution. Manual segmentation by an expert is still considered as the reference method, but is a time consuming task with high inter and intra-observer variability.

Many automated or semi-automated approaches were recently developed, showing great variability in results and performance in terms of reproducibility. Challenges in the segmentation of gliomas, from MRI data are related to:

- the infiltration of cells into the tissue, inducing unsharp borders with irregularities and discontinuities (a tumor is not necessary a single connected object),
- the great variability in their contrast uptake (depending on their vascularisation) and
- their appearance on standard MRI protocols.

MRI protocols used for brain imaging typically include Proton density (PD), T1-weighted (e.g. SPGR), T1-weighted enhanced (T1E) with contrast agent (usually Gadolinium), and T2-weighted (e.g. FLAIR) data. T1 data provides detailed anatomical views of the brain along with high signals on haemorrhages. T1E data shows strong signal on all vascularized structures (including tumors and haemorrhages), whereas usual FLAIR images (with a slice thickness around 5 mm) show less anatomical details, but high signal on tumors, infiltrations and edema.

Given an image, the segmentation task can be seen as the partition of the image into homogeneous objects, which correspond to a region-based segmentation approach, or as the detection of object contours within the image, corresponding to an edge-based segmentation approach. The majority of the MRI-based glioma segmentation methods that have been proposed in the literature are region-based. More recent methods, based on deformable models, also included edge-based information. In the case of MRI segmentation, several factors introduce a large amount of uncertainty in the segmentation process, including partial volume effects, integration of multi-protocol image data and observer variability.

3.5.2 Brain Tumor Segmentation Methods

The majority of brain tumor segmentation methods were designed in a statistical framework, providing a classification of the image data into different tissue types, while only few were designed with a deterministic approach. Some methods that were used in the past are briefly presented right below:

- **Deterministic Approaches**

In 1996, Gibbs in [47] introduced a morphological edge detection technique combined with simple region growing to segment enhancing tumors on T1 MRI data. Based on an initial sample of the enhanced tumor signal and the surrounding tissues, provided manually, an initial segmentation was performed combining pixel thresholding, fitting to an edge map of the image data and morphological opening and closing, inspired by the work proposed by Kennedy in [48]. The tumor area was defined based on pixel values in the range of 4 standard deviations around the mean value, constrained by the edge map.

In 2005, Droske in [49], proposed to use a deformable model, implemented with a level set formulation, to partition the MRI data into regions with similar image properties, based on prior intensity-based pixel likelihoods for tumoral tissues. The deformable model optimization was performed on a spatially-adaptive grid, only refined in inhomogeneous regions. Homogeneity measures included gray value intervals, defined from a user input, and image gradient values. Some manual supervision of the deformable model was required, so that incremental segmented areas were proposed to the user who controlled the final segmentation results. More specifically, heterogeneous tumors, involving necrosis for example, required successive segmentations by addition or removal of intermediate results.

- **Statistical Approaches**

In 1995, Vaiddynathan in [50], compared two supervised multispectral classification methods: k nearest neighbour (k-NN) and spectral Fuzzy C-Means (FCM). For these two classification approaches, nine tissue classes were considered (background, CSF, WM, GM, fat, muscle, tumor, edema, necrosis). The authors also tested an interactive seedgrowing segmentation approach on T1 Enhanced (T1E) MRI data. The seed-growing algorithm only segmented tumor tissue based on a

sample pixel population manually selected by the user.
In 1998, Clark in [51] introduced a knowledge-based (KB) automated segmentation method for glioblastomas on multispectral data combining T1E, PD and T2 weighted data. A training phase was performed on 17 slices from seven patients, extracting tumor size and enhancement level characteristics. Slices were first characterized as normal or abnormal via a FCM classification and the analysis of the clustering result through an expert system. Two examples of knowledge used in the predecessor system were:

i) in a normal slice, CSF belongs to the cluster center with the highest value in the intracranial region
ii) in image space, all normal tissues are roughly symmetrical along the vertical axis.

After a brain mask was computed, initial tumor segmentation, generated from vectorial histogram thresholding in the T1, PD and T2 images, was post-processed with a Knowledge Based (KB) approach to eliminate non-tumor pixels.

Tumor heuristics used in the KB system were the following:

i) Gadolinium-enhanced tumor pixels occupy the higher-end of the T1 spectrum.
ii) Gadolinium-enhanced tumor pixels occupy the higher-end of the PD spectrum, though not with the degree of separation found in T1 space.
iii) Gadolinium-enhanced tumor pixels are generally found in the "middle" of the T2 spectrum, making segmentation based on T2 values difficult.
iv) Slices with greater enhancement have better separation between tumor and non-tumor pixels, while less enhancement results in more overlap between tissue types.

It is important to note that their notion of tumor pixel included edema and necrosis. A final processing stage was performed, based on histogram analysis of the tumor pixels and heuristics on the "density" of intensity features of non-tumor tissues. Indeed, based on the observation that tumors can show different levels of enhancement and very complex shapes, the final KB approach was focused on characterizing non-tumoral tissues.

In 2001, Kaus in [52] presented a complete validation of an automated segmentation method on T1E data from twenty patients with meningiomas and low-grade gliomas. The segmentation method, called an adaptive templatemoderated classification, and described in [53, 54] was based on an iterative process. It alternated between a k-NN classification of voxels into five hierarchical tissue types (background, skin-fat-bone, brain, ventricles, tumor) and a nonlinear registration of the data with an anatomical atlas (manually segmented MRI data of a single subject) to align the data with the template. The k-NN classification used features from data intensity values and anatomical priors on the tissue location from the atlas. This method performed extraction of the five tissues in a pre-determined hierarchical order. Tissue mean values were learned on the patient's data via manual selection of three or four points for each tissue. To handle the presence of the tumor in the registration process, voxels assigned to the tumor class were masked with brain labels prior to registration with the atlas. This method obviously relied on a strong homogeneity assumption of the tumor's appearance on MRI data, which was reinforced by the use of anisotropic diffusion filtering.

In 2001 Moonis in [55] proposed a segmentation framework based on fuzzy connectedness (FC) which optimally clustered voxels into classes of high connectivity (analogous to a similarity measure). The method was applied to T1, T1E and T2 data, and initialized with an MRI data standardisation of the gray levels based on non-linear transformation of the histograms [56].

In 2005, Liu in [57], from the same group, used a similar approach based on a volume of interest on

coregistered T1 and T2 data, to process only slices containing the tumor. A set of points inside the tumor were selected to initialize the statistics used in the FC. The threshold level applied to the FC maps to define the final segmentation result was determined empirically on five datasets and then fixed once for all. Segmentation was performed separately on the T2, T1E and subtracted (T1-T1E) data sets in 3D. Manual corrections of the segmentation results were performed by experts.

In 2001, Fletcher-Heath in [58], proposed a combination of unsupervised classification with FCM and knowledge-based (KB) image processing for segmentation of non-enhancing tumors. The FCM was run on spectral data (T1, T2, PD). As the authors pointed out, FCM tended to define clusters with similar sizes, which required an initial classification in ten classes. A KB system was then designed to re-cluster the segmentation results into seven classes based on a training phase. Difficulties principally arose in the separation of CSF and tumor signals.

In 2004, Mazzara in [59], compared the k-NN approach from [50] and the KG-based approach from [51] for Growth Tumor Volume (GTV) measurements on eleven patients with high and low-grade gliomas. As used in oncology radiation therapy, GTV corresponded to the area enclosing several contiguous clusters of enhancing pixels (i.e. including non-enhancing pixels within the area). The study showed severe limitations of the KG-system (which was not trained with the dataset to segment) in handling particular cases such as non-enhancing tumor margins or the presence of non-enhancing cystic necrotic tissues at the center of the tumor. On the other hand, the k-NN segmentation method, trained with sample data from MRI slices to segment, lead to robust segmentation results on all patients. In 2006, Beyer in [60], from the same group, presented a similar and more recent comparative study, extracting GTV with the same two segmentation methods and evaluating the results in terms of predictive dose measurement for therapy planning.

In 2004, Zou in [61], proposed a continuous probabilistic segmentation framework, based on mixture modeling for two classes: tumor and non-tumor tissues. After initialization of the segmentation with the semi-automated method from Kaus in [52], the segmentation process involved estimation of the distribution parameters and probability values thresholding. Three metrics were proposed and evaluated to optimize the threshold selection: Receiver operating curve, which weights the sensitivity versus the specificity of the segmentation result, a Dice similarity coefficient, which is also a function of sensitivity and specificity and mutual information that directly compares the segmentation result to a ground truth.

In 2004, Prastawa in [62] proposed a segmentation framework based on outlier detection on T2 data. The abnormal tumor region was detected via registration on a normal brain atlas. Statistical clustering of the abnormal voxels, followed by a deformable model, were then used to isolate the tumor and the edema.

A summary of the above papers and the specificities of the clinical aspects of the evaluation setup are provided in Table 3.1 .

	Data	LGG	Glioma HG	Year
Gibs	T1E		10	1996
Letteoboer		20		2004
Droske	T1E	?		2005
Liu	FLAIR,T1,T1E		10	2005
Vaidyanathan	T1,PD,T2		4	1995
Fletcher-Heath	T1,PD,T2		6	2001
Clark	T1,PD,T2 (all with Gd)		7	1998
Kaus	SPGR-Enh	14		2001
Moonis	FLAIR		19	2001
Mazzara	T1E,FLAIR (CT)	3	8	2004
Zou	T1E (SPGR)	3		2004
Prastawa	T2	1		2004

Table 3.1: Summary of Reviewed Papers and Clinical Setup [42]

4. BASIC THEORETICAL PARTS OF THE PROPOSED ALGORITHM

4.1 Parzen Windows

4.1.1 Window Function

In Section 3.3 the basic theory behind estimating an unknown density function was presented. One of the most common techniques, is Parzen windows. The Parzen-window approach for unknown densities estimation, as detailed discussed in [63], can be introduced by assuming that the region R_n is a *d*-dimensional hypercube. Letting h_n be the length of an edge of that hypercube, then its volume is given by:

$$V_n = h_n^d \qquad \textbf{(4.1)}$$

Let us define by k_n the number of samples falling in the hypercube. An analytic expression for k_n can be obtained by firstly defining the following window function:

$$\varphi(u) = \begin{cases} 1 & , |u_j| \leqslant 1/2 \quad , j = 1,\ldots,d \\ 0 & \textit{otherwise} \end{cases} \qquad \textbf{(4.2)}$$

As we can observe, *φ(u)* defines a unit hypercube centered at the origin. As a result, $\varphi(x - x_i)/h_n$ is equal to unity if x_i falls within the hypercube of volume V_n centered at ***x***, and is zero otherwise. The number of samples, k_n, in this hypercube is therefore given by:

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{x - x_i}{h_n}\right) \qquad \textbf{(4.3)}$$

At this point we must remind that from Eq. 3.7, the *n-th* estimate for ***p(x)*** is calculated by the equation:

$$p_n(x) = \frac{k_n/n}{V_n} \qquad \textbf{(4.4)}$$

and when we substitute Eq. 4.3 into Eq. 4.4 we obtain the estimate:

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{x - x_i}{h_n}\right) \qquad \textbf{(4.5)}$$

Eq. 4.5 suggests a more general approach to estimating unknown density functions. Instead of limiting ourselves to the hypercube window function that we defined in Eq. 4.2, suppose that we allow a more general class of window functions. In such a case, Eq.4.5 expresses our estimate for ***p(x)*** as an average of functions of *x* and the samples x_i. In essence, the window function is being used for interpolation (**each sample contributes to the estimate in accordance with its distance from *x***). As the estimate $p_n(x)$ is a density function, it must be guaranteed that this estimate for ***p(x)*** is nonnegative and integrated to one. This fact can be simply assured by requiring the window

function itself to be a density function. In order to be more precise, if we require that:

$$\varphi(x) \geq 0 \qquad \textbf{(4.6)}$$

and

$$\int \varphi(u)\,du = 1 \qquad \textbf{(4.7)}$$

and by maintaining the relation $V_n = h_n^d$, then continually we can observe that $p_n(x)$ also satisfies these conditions. Now, let us examine the effect that the window width h_n has on $p_n(x)$. If we denote the function $\delta_n(x)$ by:

$$\delta_n(x) = \frac{1}{V_n}\varphi\left(\frac{x}{h_n}\right) \qquad \textbf{(4.8)}$$

then $p_n(x)$ can be written as the average:

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(x - x_i) \qquad \textbf{(4.9)}$$

As a consequence, h_n affects significantly both the amplitude and the width of $\delta_n(x)$, since $V_n = h_n^d$ (Fig. 4.1). If h_n is very small, then the peak value of $\delta_n(x - x_i)$ is too large and occurs near $x = x_i$. In this case *p(**x**)* is the superposition of *n* sharp pulses centered at the samples, ending up to an erratic, "noisy" estimate (Fig. 4.2). On the other hand, if h_n is very large, then the amplitude of δ_n is too small, and ***x*** must be far from x_i before $\delta_n(x - x_i)$ changes much from $\delta_n(0)$. In this case, $p_n(x)$ is the superposition of *n* broad, slowly changing functions and is a very smooth "out-of-focus" estimate of *p(**x**)*. The distribution is normalized, for any value of h_n :

$$\int \delta_n(x - x_i)\,dx = \int \frac{1}{V_n}\varphi\left(\frac{x - x_i}{h_n}\right)dx = \int \varphi(u)\,du = 1 \qquad \textbf{(4.10)}$$

Clearly, we come to the conclusion that the choice of h_n (or V_n) has an important effect on $p_n(x)$. If V_n is too small, the estimate will suffer from too much statistical variability whereas if V_n is too large, the estimate will suffer from too little resolution. Having a limited number of samples, the best that can be done is to seek some acceptable compromise. Hypothetically, with an unlimited number of samples, it is possible to let V_n slowly approach zero as *n* increases and have $p_n(x)$ converge to the unknown density *p(**x**)*. At this point we should remember that as for any fixed ***x*** the value of $p_n(x)$ depends on the random samples $x_1, \ldots, x_n$, $p_n(x)$ is a random variable, with some mean $\bar{p}_n(x)$ and variance $\sigma_n^2(x)$ and we are talking about the convergence of a sequence of random variables. Taking this into consideration, the estimate $p_n(x)$ converges to *p(**x**)* if:

$$\lim_{n\to\infty} \bar{p}_n(x) = p(x) \qquad \textbf{(4.11)}$$

and

$$\lim_{n\to\infty} \sigma_n^2(x) = 0 \qquad \textbf{(4.12)}$$

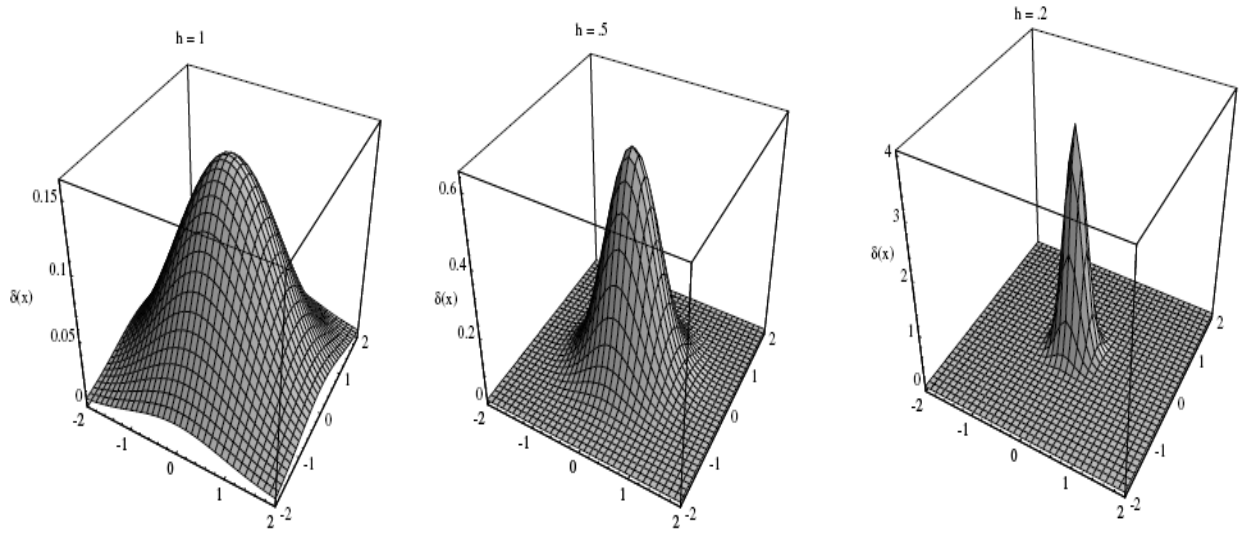


Fig. 4.1: Examples of two-dimensional circularly symmetric normal Parzen windows *φ(x/h)* for three different values of *h*. Note that because the $\delta_k(.)$ are normalized, different vertical scales must be used to show their structure [63].

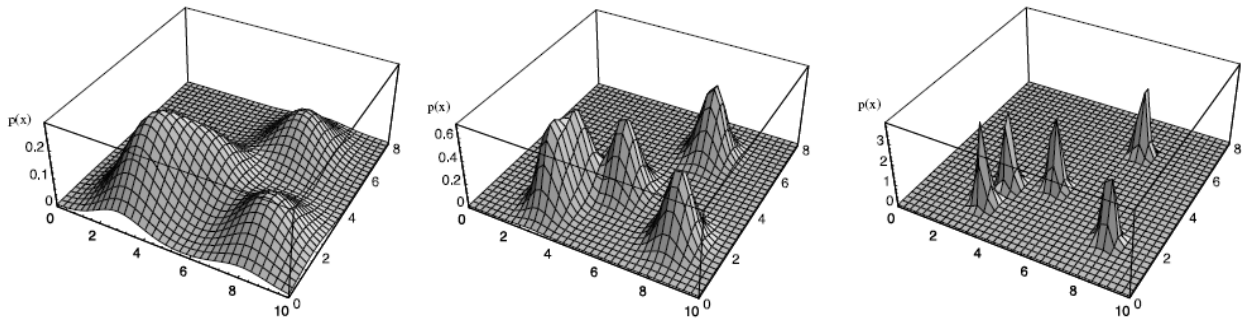


Fig. 4.2: Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.1. As before, the vertical axes have been scaled to show the structure of each function [63].

In order to prove convergence, conditions must be placed on the unknown density *p(**x**)*, on the window function *φ(u)*, and on the window width h_n. In general, continuity of *p(.)* at ***x*** is required, and the conditions imposed by Eqs. 4.6 & 4.7 are customarily invoked. It can be shown that convergence can be assured by the following additional conditions:

$$\sup_u \varphi(u) < \infty \qquad \textbf{(4.13)}$$

$$\lim_{\|u\|\to\infty} \varphi(u) \prod_{i=1}^{d} u_i = 0 \qquad \textbf{(4.14)}$$

$$\lim_{n\to\infty} V_n = 0 \qquad \textbf{(4.15)}$$

and:

$$\lim_{n\to\infty} nV_n = \infty \qquad \textbf{(4.16)}$$

Equations 4.13 & 4.14 manage to keep *φ(.)* well behaved, while equations 4.15 & 4.16 state that the volume V_n must approach zero at a rate slower than *1/n*.

4.1.2 Classifiers based on Parzen Windows

In classifiers based on Parzen-window estimation, we estimate the densities for each category and classify a test point by the label corresponding to the maximum posterior. The decision regions for a Parzen-window classifier depend upon the choice of window function. In general, the training error, the empirical error on the training points themselves, can be made arbitrarily low by making the window width sufficiently small. However, the goal of creating a classifier is to classify novel patterns, and alas a low training error does not guarantee a small test error. Although a generic Gaussian window shape can be justified by considerations of noise, statistical independence and uncertainty, in the absense of other information about the underlying distributions there is little theoretical justification of one window width over another.

The advantages of classifiers based on Parzen windows resides in their generality. We do not need to make any assumptions about the distributions ahead of time. With enough samples, we are essentially assured of convergence to an arbitrarily complicated target density. On the other hand, the number of samples needed may be very large indeed, much greater than would be required if we knew the form of the unknown density. Moreover, the demand for a large number of samples grows exponentially with the dimensionality of the feature space. This limitation is known as the "curse of dimensionality" and severely restricts the practical application of such nonparametric procedures, including brain tissues segmentation.

4.2 The Mean-Shift Procedure

4.2.1 Constant-Adaptive Mean-Shift

Parzen window density estimation (or kernel density) is considered to be the most popular density estimation method. Given *n* data points, x_i , $i=1,\ldots,n$ in the *d*-dimensional feature space R^d , the multivariate kernel density estimator with kernel *K(x)* and a symmetric positive definite *d*d* bandwidth matrix *H*, computed in the point ***x***, as detailed discussed in [37] is given by:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - x_i) \qquad \textbf{(4.17)}$$

where:

$$K_H(x) = |H|^{-1/2}\, K(H^{-1/2}x) \qquad \textbf{(4.18)}$$

where *K(x)* is a d-variate, bounded function with compact support, which satisfies the following:

$$\int_{R^d} K(x)\, dx = 1 \qquad \lim_{\|x\|\to\infty} \|x\|^d K(x) = 0$$

$$\int_{R^d} xK(x)\, dx = 0 \qquad \int_{R^d} xx^T K(x) = c_K I \qquad \textbf{(4.19)}$$

where c_k is a constant. The multivariate kernel can be generated from a symmetric univariate $K_1(x)$ in two different ways:

$$K^S(x) = a_{k,d} K_1(\|x\|) \qquad K^P(x) = \prod_{i=1}^{d} K_1(x_i) \qquad \textbf{(4.20)}$$

where $K^S(x)$ is obtained from rotating $K_1(x)$ in R^d, i.e. , $K^S(x)$ is radially symmetric. The constant $a^{-1}_{(k,d)} = \int_{R^{(d)}} K_1(\|x\|)\, dx$ assures that $K^S(x)$ integrates to one. On the other hand, $K^P(x)$ is obtained from the product of univariate kernels. Either type of multivariate kernel obeys Eq. 4.19, but, for our purposes, the radially symmetric kernels satisfies:

$$K(x) = c_{k,d}\, k\left(\|x\|^2\right) \qquad \textbf{(4.21)}$$

In this case, it suffices to define the function *k(x),* called the *profile of the kernel*, only for *x >=0*. The normalization constant $c_{k,d}$ which makes *K(x)* integrate to one, is assumed strictly positive.

Using a fully parameterized *H* increases the complexity of the estimation and, in practice, the bandwidth matrix *H* is chosen either as diagonal $H = diag[h_1^2, \ldots, h_d^2]$, or proportional to the identity matrix $H = h^2 I$. The clear advantage of the latter case is that only one bandwidth parameter $h > 0$ must be provided. However , as can be seen from Eq. 4.18, then the validity of an Euclidean metric for the feature space should be confirmed first. Employing only one constant bandwidth parameter, the kernel density estimator (Eq. 4.17) becomes the well-known expression:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad \textbf{(4.22)}$$

Intuitively, as referred in [41], a single *h* value may be inappropriate for a locally variable set of feature points, especially in brain structures segmentation. In areas where the density is low, as is often the case with high dimensional spaces, a single *h* value could be too small to reach neighbors and the iterative procedure would not start, that is feature points would be associated to themselves thus producing cluster over-splitting. The same fixed value could be too large for high density areas, leading to an undesirable merging of clusters in regions representing fine detail.

Moreover, in order to avoid under-segmentation, caused by a single value that is locally too large, it is necessary to choose a bandwidth value that is small enough to avoid over-smoothing anywhere in the dataset. This results in a strong over-splitting in the mean-shift output and requires further merging in the form of an iterative repetition of transitive closure on the region adjacency graph

followed by a union-find algorithm, before the final tissue classification is obtained [64]. In a subsequent conference paper [65], the same group recognized the superiority of the adaptive mean-shift for the segmentation of data that exhibits multiscale patterns as is the case with MRI of the brain [66].

The mean-shift implementation which uses an adaptive *h* value instead of a constant *h* is called adaptive mean-shift (AMS) algorithm and is the basis of the proposed algorithm. In high dimensional feature spaces, adaptive mean-shift clustering has been shown to produce better results than the fixed bandwidth algorithm as mentioned in [67]. Several methods have been proposed to determine an adaptive window size for the AMS algorithm [68], [69]. Although, the window size can be also simply defined as the distance h_i between x_i and it's *k*-nearest neighbor:

$$h_i = \| x_i - x_{i,k} \| \qquad \textbf{(4.23)}$$

The neighbors of x_i are sorted by order of increasing distance to x_i. Following the ordering process, $x_{i,k}$ is the k-th distant neighbor from x_i, and h_i is its distance to x_i. The number of neighbors considered for, should be chosen large enough to ensure that there is an increase in density within the support of most kernels. In this work, this method for h_i selection was preferred.

Using an adaptive *h* value, yields Eq. 4.22:

$$\hat{f}_K(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i^d} k\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \qquad \textbf{(4.24)}$$

4.2.2 Mean-Shift Kernels

The quality of a kernel density estimator, as mentioned in [37], can be measured by the mean of the square error between the density and its estimate, integrated over the domain of definition. In practice, however, only an asymptotic approximation of this measure (denoted as AMISE) can be computed. Under the asymptotics, the number of data points $n \to \infty$, while the bandwidth $h \to 0$ at a rate slower than n^{-1}. For both types of multivariate kernels, the AMISE measure is minimized by the *Epanechnikov Kernel* having the profile:

$$k_E(x) = \begin{cases} 1 - x & 0 \leqslant x \leqslant 1 \\ 0 & x > 1 \end{cases} \qquad \textbf{(4.25)}$$

which yields the radially symmetric kernel:

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1 - \|x\|^2) & \|x\| \leqslant 1 \\ 0 & \textit{otherwise} \end{cases} \qquad \textbf{(4.26)}$$

where c_d is the volume of the unit *d*-dimensional sphere. We can notice though, that the Epanechnikov profile is not differentiable at the boundary. The profile:

$$k_N(x) = \exp(-\frac{1}{2}x) \quad , x \geqslant 0 \qquad \textbf{(4.27)}$$

yields the multivariate *normal kernel*:

$$K_N(x) = (2\pi)^{-d/2}\exp\left(-\frac{1}{2}\|x\|^2\right) \qquad \textbf{(4.28)}$$

for both types of composition (Eq. 4.20). The normal kernel is often symmetrically truncated to have a kernel with finite support. Employing the profile notation, the density estimator (Eq. 4.24) can be rewritten as:

$$\hat{f}_{h,K}(x) = \frac{c_{k,d}}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right) \qquad \textbf{(4.29)}$$

Finding the modes of the underlying density ***f(x)*** is the first step in the analysis of a feature space. The modes are located among the zeros of the gradient $\nabla f(x)=0$ and the mean shift procedure is an elegant way to locate these zeros **without estimating the density**.

4.2.3 Density Gradient Estimation

By taking the gradient of the density estimator and by exploiting the linearity of Eq. 4.29, the following density gradient estimator can be obtained:

$$\hat{\nabla} f_{h,K}(x) \equiv \nabla \hat{f}_{h,k}(x) = \frac{2c_{k,d}}{n}\sum_{i=1}^{n}\frac{1}{h_i^{d+2}}(x-x_i)k'\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right) \qquad \textbf{(4.30)}$$

Assuming that the derivative of the kernel profile exists for all $x \exists [0,\infty)$ except for a finite set of points, we thus define the function:

$$g(x) = -k'(x) \qquad \textbf{(4.31)}$$

Now, using *g(x)* for profile, the kernel *G(x)* is defined as:

$$G(x) = c_{g,d}\,g\left(\|x\|^2\right) \qquad \textbf{(4.32)}$$

where $c_{g,d}$ is the corresponding normalization constant. The kernel *K(x)* was called the *shadow* of *G(x)* in [70] in a slightly different context. We can observe that the Epanechnikov kernel is the *shadow* of the uniform kernel, while the normal kernel and its shadow have the same expression. Introducing *g(x)* into Eq. 4.30 yields:

$$\hat{\nabla} f_{h,k}(x) =$$

$$= \frac{2c_{k,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} (x_i - x)\, g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)$$

$$= \frac{2c_{k,d}}{n} \left[\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i\, g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)} - x \right] \quad \textbf{(4.33)}$$

where $\sum_{i=1}^{n} g(\|(x - x_i)/h\|^2)$ is assumed to be a positive number. This condition is easy to satisfy for all the profiles met in practice. Both terms of the product in Eq. 4.33 have special significance. From Eq. 4.29, the first term is proportional to the density estimate at $\boldsymbol{x}$ computed with the kernel G:

$$\hat{f}_{h,G}(x) = \frac{c_{g,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d}} g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \quad \textbf{(4.34)}$$

The second term is the mean shift:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i\, g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \quad \textbf{(4.35)}$$

We can observe that mean-shift is actually the difference between the weighted mean, using the kernel G and the adaptive h value for weights, and $\boldsymbol{x}$, the center of the kernel (window). From Eqs 4.34 and 4.35, Eq. 4.33 becomes:

$$\hat{\nabla} f_{h,K}(x) = \hat{f}_{h,G}(x) \frac{2c_{k,d}}{c_{g,d}} m_{h,G}(x) \quad \textbf{(4.36)}$$

yielding:

$$m_{h,G}(x) = \frac{1}{2} c \frac{\hat{\nabla} f_{h,K}(x)}{\hat{f}_{h,G}(x)} \quad \textbf{(4.37)}$$

This expression (Eq. 4.37) shows that, at location $\boldsymbol{x}$, the mean shift vector computed with kernel G, **is proportional to the normalized density gradient estimate obtained with kernel *K***. The normalization is by the density estimate in $\boldsymbol{x}$ computed with the kernel G. **The mean shift vector thus always points toward the direction of maximum increase in density.** This is a more general formulation of the property first remarked by Fukunaga and Hosteler [71,72] and discussed in [70].

Though, it must be mentioned that the relation captured in Eq. 4.37 is intuitive, as the local mean is shifted toward the region in which the majority of the points reside. Although, a path which leads to a stationary point of the estimated density can be defined, since the mean shift vector is aligned with

the local gradient estimate. The modes of the density are such stationary points. As a conclusion, the mean-shift procedure, obtained by successive:

- computation of the mean shift vector $m_{h,G}(x)$
- translation of the kernel (window) *G(x)* by $m_{h,G}(x)$

is guaranteed to converge at a nearby point. At this point, the estimate (Eq. 4.29) has zero gradient, as will be shown in the next section. The presence of the normalization by the density estimate is a desirable feature. Finally, it can be claimed that the mean shift procedure is an adaptive gradient ascend method, as close to local maximum points, in high-density values, the mean-shift steps are small and the analysis more refined. The regions of low-density values are of no interest for the feature space analysis and in such regions the mean shift steps are large.

4.2.4 Convergence's Sufficient Condition

Let us define by $\{y_j\}$, $j=1,2,...$ the sequence of successive locations of the kernel *G*, where, from Eq. 4.35 is calculated in the following way:

$$y_{j+1} = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i g\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)} \qquad j=1,2,... \qquad \textbf{(4.38)}$$

and it is actually the weighted mean at y_j computed with kernel *G* and y_1 is the center of the initial position of the kernel. Respectively, the corresponding sequence of density estimates computed with kernel *K*, $\{\hat{f}_{h,K}(j)\}$, $j=1,2,..$ is given by:

$$\hat{f}_{h,K}(j) = \hat{f}_{h,K}(y_j) \quad j=1,2,... \qquad \textbf{(4.39)}$$

The following theorem states that when a kernel *K* obeys some mild conditions, it suffices for the convergence of the sequences $\{y_j\}$, $j=1,2,...$ and $\{\hat{f}_{h,K}(j)\}$, $j=1,2,...$.

Theorem 4.1 (Capture Theorem)

If the kernel K has a convex and monotonically decreasing profile, the sequences $\{y_j\}$, $j=1,2,...$ *and* $\{\hat{f}_{h,K}(j)\}$, $j=1,2,...$ *converge and* $\{\hat{f}_{h,K}(j)\}$, $j=1,2,...$ *is monotonically increasing.*

We can observe that this theorem generalizes the result derived in a different way from Eq.4.31, where *G* was the uniform kernel and *K* was the Epanechnikov kernel. A non-negative weight w_i must be associated in each data point x_i, in order to be this theorem valid. The proof of this theorem is given in the Appendix B.1.

In [70, Section iv] the convergence property of the mean shift was also discussed (However, it must be emphasized that almost all the discussion there was concerned about the blurring process in which the input, after each mean shift step, is recursively modified). In our case, the convergence of

the mean-shift procedure was attributed in [70] to the gradient ascent of nature of Eq. 4.37. However, moving in the direction of the local gradient, as shown in [73, Section 1.2], guarantees convergence **only for infinitesimal steps**. As a consequence, the step size of a gradient-based algorithm is crucial for the overall performance. If the step size is too small, the rate of convergence may be very slow while if the step size is too large, the algorithm will diverge. In order to select appropriately the step size, a number of costly procedures have been developed, as discussed in [73, p. 24]. Due to the adaptive magnitude of the mean shift vector, the convergence of the mean-shift procedure is guaranteed, as shown by theorem 4.1, **eliminating the need for additional procedures to chose the adequate step sizes**. This is a major advantage over the traditional gradient-based methods.

The number of steps to convergence in the mean-shift procedure, for discrete data, depends on the employed kernel. When the uniform kernel is applied, convergence is achieved in a finite number of steps, since the number of locations generating distinct mean values is also finite. However, when the kernel *G* imposes a weighting on the data points, according to the distance from its center, the mean shift procedure is infinitely convergent. Setting a threshold for the magnitude of the mean shift vector, is the most practical way to stop the mean-shift iterations.

4.2.5 Mode Detection

Suppose that y_c and $\hat{f}^{c}_{h,k} = \hat{f}_{h,k}(y_c)$ are the convergence points of the sequences $\{y_j\}$, $j=1,2,...$ and $\{\hat{f}_{h,K}(j)\}$, $j=1,2,...$ respectively. The two implications of Capture Theorem are the following:

Firstly, taking Eqs 4.35 and 4.38, the *j*-th mean shift vector is :

$$m_{h,G}(y_j) = y_{j+1} - y_j \qquad \textbf{(4.40)}$$

After some iterations, $m_{h,G}(y_j)$ becomes at the limit $m_{h,G}(y_j) = y_c - y_c = 0$, that is to say the magnitude of the mean shift vector converges to zero. In other words, the gradient of the density estimate (Eq.4.29) computed at y_c is zero:

$$\nabla \hat{f}_{h,K}(y_c) = 0 \qquad \textbf{(4.41)}$$

due to Eq. 4.37. Hence, $\boldsymbol{y_c}$ **is a stationary point of** $\boldsymbol{\hat{f}_{h,K}}$**.**

Secondly, we remind that the Capture Theorem states that the trajectories of gradient methods are attracted by local maximum if they are unique stationary points, within a small neighborhood. Because of the fact that $\{\hat{f}_{h,K}(j)\}$, $j=1,2,...$ is monotonically increasing, the mean shift iterations satisfy the conditions required by the Capture Theorem [Appendix B.1]. Hence, once y_j gets sufficiently close to a mode of $\hat{f}_{h,K}$, it converges to it. The set of all locations that converge to the same mode defines the basin of attraction of that mode.

To sum up, these two theoretical observations from above suggest a practical algorithm for mode detection:

- Run the mean shift procedure to find the stationary points of $\hat{f}_{h,K}$
- Prune these points by retaining only the local maximum within a small neighborhood

The local maximum points are defined, according to the Capture Theorem, as unique stationary points within some small open sphere. This property can be tested by perturbing each stationary point by a random vector of small norm and letting the mean-shift procedure converge again. Should the point of convergence be unchanged (up to a tolerance), the point is a local maximum, as discussed in [37].

4.2.6 Smooth Trajectory Property

When the normal kernel is used, the path of the mean shift procedure toward the mode has the following smooth trajectory property: The angle between two consecutive mean shift vectors is always less than 90 degrees.

Using the normal (Gaussian) kernel (Eq. 4.28), the *j*-th mean shift vector (Eq. 4.38) is now given by:

$$m_{h,N}(y_j) = y_{j+1} - y_j = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i \exp\left(\left\| \frac{x - x_i}{h_i} \right\|^2\right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} \exp\left(\left\| \frac{x - x_i}{h_i} \right\|^2\right)} - y_j \qquad \textbf{(4.42)}$$

Theorem 4.2

The cosine of the angle between two consecutive mean shift vectors is strictly positive when a normal kernel, as used in our work, is employed:

$$\frac{m_{h,N}(y_j)^T m_{h,N}(y_{j+1})}{\|m_{h,N}(y_j)\| \, \|m_{h,N}(y_{j+1})\|} > 0 \qquad \textbf{(4.43)}$$

The above theorem holds true for all $j = 1,2,\ldots$ according to the proof given in Appendix B.2. An implication of Theorem 4.2 is that **the Normal (Gaussian) kernel appears to be the optimal one for the mean shift procedure.** The smooth trajectory of the mean shift procedure is in contrast with the standard steepest ascent method whose convergence rate on surfaces with deep narrow valleys is slow due to its zigzagging trajectory. In practice, the convergence of the mean shift procedure based on the normal kernel requires large number of steps, as was discussed at the end of section 4.2.4 in contrast with the uniform kernel which needs much fewer steps for the mean shift procedure to converge. However, in this work the normal kernel is used, as it will be discussed in later section, as it produces better results than the uniform kernel.

4.3 Distances calculation

4.3.1 Euclidean distance

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula, as defined in [74]. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space.

The Euclidean distance between points p and q is defined as the length of the line segment $\bar{pq}$. In Cartesian coordinates, if $p=(p1, p2, \ldots, p_n)$ and $q=(q1, q2, \ldots, q_n)$ are two points in Euclidean *n-space*, then the distance from *p* to *q* is given by:

$$d(p,q) = \sqrt{(p_1-q_1)^2+(p_2-q_2)+\ldots+(p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2} \qquad \textbf{(4.44)}$$

The Euclidean norm measures the distance of a point to the origin of Euclidean space:

$$\|p\| = \sqrt{p_1^2+p_2^2+\ldots+p_n^2} = \sqrt{p*p} \qquad \textbf{(4.45)}$$

where the last equation involves the dot product. This is the length of *p*, when regarded as a Euclidean vector from the origin. The distance itself is given by:

$$\|p-q\| = \sqrt{(p-q)(p-q)} = \sqrt{\|p\|^2+\|q\|^2-2\text{pq}} \qquad \textbf{(4.46)}$$

4.3.2 Mahalanobis Distance

In statistics, as referred in [75], Mahalanobis distance is a distance measure introduced by P. C. Mahalanobis in 1936. It is based on correlations between variables by which different patterns can be identified and analyzed. It is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e. not dependent on the scale of measurements.

Formally, the Mahalanobis distance of a multivariate vector $x=(x_1, x_2, x_3, \ldots, x_n)^T$ from a group of values with mean $\mu=(\mu_1, \mu_2, \mu_3, \ldots, \mu_n)^T$ and covariance matrix *S* is defined as:

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \qquad \textbf{(4.47)}$$

Mahalanobis distance (or "generalized squared interpoint distance" for its squared value) can also be defined as a dissimilarity measure between two random vectors and of the same distribution with the covariance matrix S:

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1}(\vec{x}-\vec{y})} \qquad \textbf{(4.48)}$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called the

normalized Euclidean distance:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}} \qquad \textbf{(4.49)}$$

where σ_i is the standard deviation of the x_i over the sample set.

An intuitive explanation of mahalanobis distance can be given if we consider the problem of estimating the probability of a test point in N-dimensional Euclidean space to belong to a set, where we are given sample points that definitely belong to that set. Our first step would be to find the average or center of mass of the sample points. Intuitively, the closer the point in question is to this center of mass, the more likely it is to belong to the set.

However, we also need to know if the set is spread out over a large range or a small range, so that we can decide whether a given distance from the center is noteworthy or not. The simplistic approach is to estimate the standard deviation of the distances of the sample points from the center of mass. If the distance between the test point and the center of mass is less than one standard deviation, then we might conclude that it is highly probable that the test point belongs to the set. The further away it is, the more likely that the test point should not be classified as belonging to the set.

This intuitive approach can be made quantitative by defining the normalized distance between the test point and the set to be $(x - \mu)/\sigma$. By plugging this into the normal distribution we can derive the probability of the test point belonging to the set.

The drawback of the above approach was that we assumed that the sample points are distributed about the center of mass in a spherical manner. Were the distribution to be decidedly non-spherical, for instance ellipsoidal, then we would expect the probability of the test point belonging to the set to depend not only on the distance from the center of mass, but also on the direction. In those directions where the ellipsoid has a short axis the test point must be closer, while in those where the axis is long the test point can be further away from the center.

Putting this on a mathematical basis, the ellipsoid that best represents the set's probability distribution can be estimated by building the covariance matrix of the samples. The Mahalanobis distance is simply the distance of the test point from the center of mass divided by the width of the ellipsoid in the direction of the test point.

4.4 K-means algorithm

4.4.1 Introduction

Suppose that we want to classify *n* samples in *k* clusters (obviously $k < n$). The K-means algorithm depends on minimizing the square sum of distances between the n samples and the center of each cluster, as mentioned in [76]. To be more specific we want:

$$\sum_{x \in S_j(k)} \left(|x - z_j| \right)^2 \qquad \textbf{(4.50)}$$

where $S_j(k)$ is a cluster in *k* iteration, z_j is the center of the cluster and the expression inside the | . | is usually the Euclidean distance. As a conclusion, the function that has to be minimized for all

the clusters is:

$$J(Z)=\sum_{j}\sum_{x\in S_j(k)}\left(\left\|x-z_j\right\|\right)^2 \qquad \textbf{(4.51)}$$

If we take the gradient of function J we have:

$$\theta J(Z)/\theta z_j=\sum_{x\in S_j(k)}(x-z_j)=0,\forall\, j \qquad \textbf{(4.52)}$$

As a result, the optimal solution for the centers of clusters is:

$$z_j=(1/p_j)*\sum_{x\in S_j(k)}x \qquad \textbf{(4.53)}$$

where p_j is the number of members of class $S_j(k)$, that is to say z_j is the mean value of the members of each class.

4.4.2 K-means Steps

The K-means algorithm includes the following steps:

1. We define the number of clusters, *K*
2. We choose,randomly or by approach, K elements that they will be the initial centers of the *K* clusters
3. For the rest elements, we calculate their distance from the center of the *K* clusters, and we place them to the appropriate cluster, according to the minimum distance from the *K* clusters.
4. We re-calculate the centers of the clusters, by calculating the mean of the members of each cluster
5. We re-calculate the distances from the center of the *K* clusters, and once again we place them to the appropriate cluster.

We repeat steps 4 and 5 until there is no change in the mean values of the clusters, and as a result, the algorithm converges.

4.5 Fuzzy K-means

4.5.1 Introduction

The basic difference between the fuzzy classifiers and the binary ones, is that in the fuzzy classifiers, the elements have the possibility/opportunity to be classified in more than one clusters.

To be more specific, suppose that we have *n* elements, $x_{1,} x_{2,...}, x_n$ that belong to the set *S* and we are looking for the *K* clusters $S_{1,} S_{2...,} S_k$ in a way that for each $x_i, i=1,2,...,n$ to be classified in clusters so that $S_1 \cup S_2 \cup ... \cup S_k = S$ without the relation $S_i \cup S_j = \emptyset$, $\forall i \neq j$ to be accepted.

This idea is very practical and extremely useful, if we consider facts of the real world. Suppose that we have to classify men according to their height, in the clusters { Short, Medium, High }.The classification of the sample with height for instance 185 cm, it is hard to be classified only in one of the categories Medium or High. Differently , it is preferable to characterize this man as " he belongs 0.6 in class Medium and 0.4 in class High (we observe that 0.6 + 0.4 = 1)"

4.5.2 Fuzzy K-means Definition and Steps

The Fuzzy K-means algorithm, as defined in [77], depends on minimizing of the function:

$$J(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} (|x_k - v_i|)^2 \qquad \textbf{(4.54)}$$

where:

1. $x_{1,} x_{2,...}, x_n$ are the number of elements that we want to classify
2. $V = \{v_{1,} v_{2,...}, v_c\}$ are the center of the clusters
3. $U = [u_{ik}]$ is a *c* * *n* matrix where u_{ik} is the percentage of participation in the *i* cluster of *k* from the *n* elements of the sample.

   Each u_{ik} element satisfies the following:

   - $0 \leq u_{ik} \leq 1, \quad i = 1,..., c, \quad k = 1,..., n$
   - $\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1,..., n$
   - $0 \leq \sum_{k=1}^{n} u_{ik} < n, \quad i = 1,..., c$

4. *m>1* is a predefined exponential fuzzifier factor.

The fuzzy K-means algorithm includes the following steps:

1. Supposing that we have *n* elements, $X = \{x_{1,} x_{2,\dots}, x_n\}$, we define the number of clusters $2 \le c \le N$, the maximum number of iterations T, the value of $m > 1$ and the value of the constant $\varepsilon > 0$.
2. We define (randomly or with another way) the initial values of the participation matrix.
3. For all the *n* elements, we calculate their distance from the center of the *c* clusters, and we place them to the appropriate cluster, according to the minimum distance from the *c* clusters.
4. For $t = 1,2,\dots,T$ we calculate the *c* centers of the clusters according to the following equation:

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_{ik}}{\sum_{k=1}^{n} u_{ik}^m} \quad , i = 1,\dots,c \qquad \textbf{(4.55)}$$

(mean value of x_{ik} with weights u_{ik}^m)

5. We calculate the membership matrix:

$$u_{ik} = \frac{[1/(|x_k - v_i|)^2]}{\sum_{j=1}^{c} [1/(|x_k - v_j|)^2]^{1/(m-1)}} \quad , i = 1,\dots,c \text{ and } k = 1,\dots,n \qquad \textbf{(4.56)}$$

6. We stop the iterative procedure when the number of iterations reaches the limit that we have define or there is no change in the mean values of the clusters or when:

$$A_i = \sqrt[p]{det(S_i) S_i^{-1}} \qquad \textbf{(4.57)}$$

In any other case we return to step 3 and we repeat the procedure.

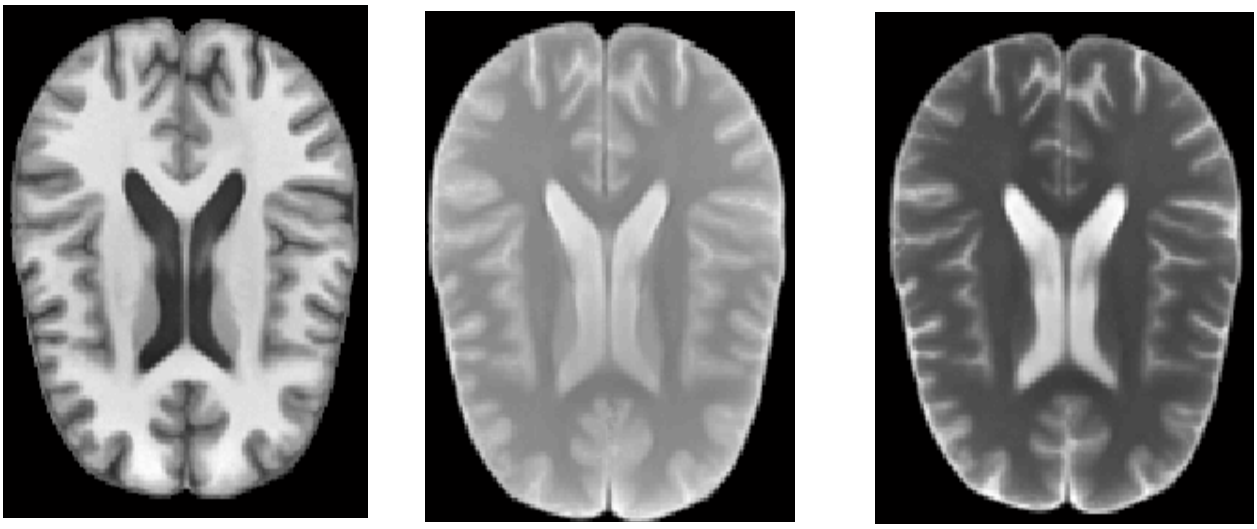
5. THE PROPOSED MEAN-SHIFT ALGORITHM

5.1 Introduction

The proposed brain structures segmentation algorithm is based on the adaptive mean-shift theory. It is utilized to analyze 2-D MRI data and provides segmentation maps of the three main brain tissue types (Gray matter (GM), White matter (WM), Cerebro-spinal fluid (CSF)) and a membership matrix for each tissue which contains the possibility for each pixel to be this tissue. One MRI modality is available per segmentation task. An example of such data is shown in Fig. 5.1. The initial MRI image needs to be preprocessed. The preprocessing step includes the following steps:

1. Brain parenchyma extraction using the brain extraction tool (BET) so to segment only the brain tissue pixels.
2. Apply of Median Filter
3. Intensity normalization across based on the darkest and brightest percentage points. The normalization sets the darkest percent of pixels to zero and rescales the brightest percent to 4095. The purpose is to obtain similar dynamic ranges for all the three tissues.
4. Background extraction

Our feature space consists of pixel's intensity and spatial information (pixel coordinates) for an overall dimensionality of three. Following the initial data processing, feature-vectors are extracted per input pixel. The set of feature-vectors is input to the adaptive mean-shift clustering stage, which is explained in Section 5.3. The output of the clustering step is a set of modes which provides a compact representation of the data. If the output of the mean-shift clustering stage includes several hundreds of modes (>400), a follow-up merging stage is proposed to further prune the initial set of modes. The iterative mode-pruning stage is described in Section 5.4. Finally, the categorization of the resultant modes into three categories, as defined in the brain segmentation task, is achieved via an intensity-based clustering stage, described in Section 5.5, using the fuzzy k-means algorithm. The output of the proposed algorithm is a segmentation map of all the pixels of the datasets and a membership map, with possibilities for each pixel to be CSF, GM and WM. Fig. 5.2 shows a summarizing block diagram for the proposed algorithm. Furthermore, the value of this algorithm in automatic detection of abnormalities in brain images is also investigated in Section 5.6.



(a) **(b)** **(c)**

Fig. 5.1: Three MRI modalities: (a) **T1** , (b) **Proton density (Pd)** , (c) **T2**

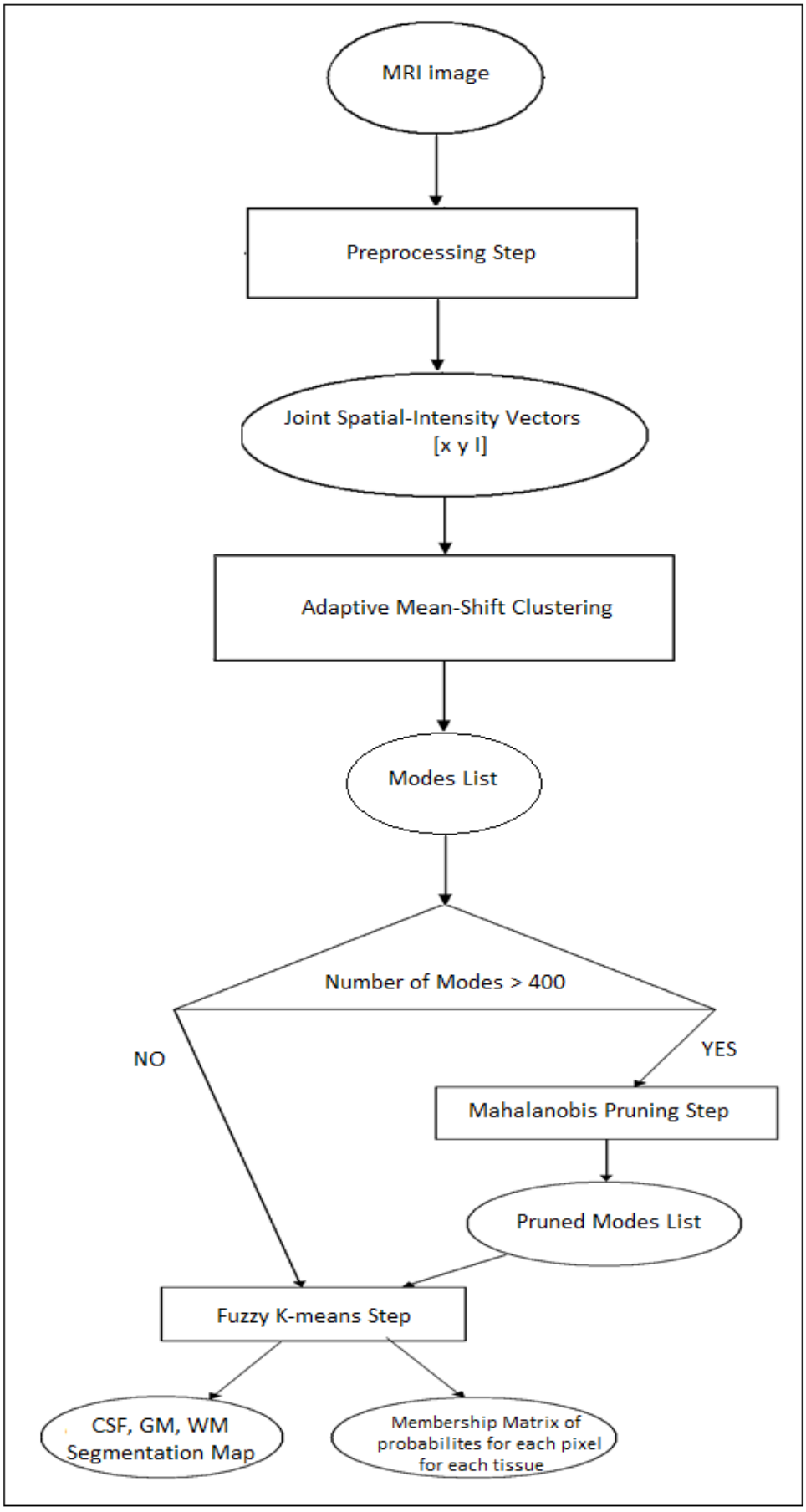


Fig 5.2: Block diagram for the proposed algorithm

5.2 Preprocessing Step

5.2.1 Brain Extraction

First of all, for all the images of the dataset, we must remove the scalp and scull pixels, in order to guarantee that the rest of the procedure will not classify pixels that do not belong to any tissue category. There is a great variety of appropriate brain extraction tools (BET) available, although neither of which remove completely all the scalp and scull pixels. However, in this work we used the BET tool of MRIcron program suite [78], in combination with a manual scalp and scull pixels removal. The only parameter that we have to choose in BET is a "crop edges" threshold, that regulates how many pixels should the program crop. It takes values from 0 to 1. Close to zero values means that the program is going to crop few pixels from the MRI image, and as a result plenty scull pixels will still remain, leading to misclassification (figure 5.3). On the other hand, close to 1 values means that the program is going to crop plenty pixels from the MRI image, and as a result pixels that belong to brain tissues will be mistakenly cropped. An instance of that case is shown in figure 5.4. As a conclusion much caution is needed with the use of this tool and in every occasion, we must check the resulting image. In this work, we observed that the best way to remove scalp and scull pixels is to apply BET one to two times with threshold value 0.4 to 0.6 max. Afterwards, the remaining scalp and scull pixels are removed manually. The basic idea is to apply BET carefully with threshold values 0.4 to 0.6, even though it may needs to be applied more than one time, so as to guarantee that scalp and scull pixels are completely removed and brain tissue pixels have not been cropped. BET procedure is of paramount importance, especially in locating cancer pixels (tumor or edema), that appear hyperintence (in T1 and T2 MRI modality respectively), an intensity value very close to brain scull pixels intensity, and if we don't remove them appropriately, we will end up to misclassification of brain cancer pixels. In figure 5.5 we can observe a good brain extraction example.

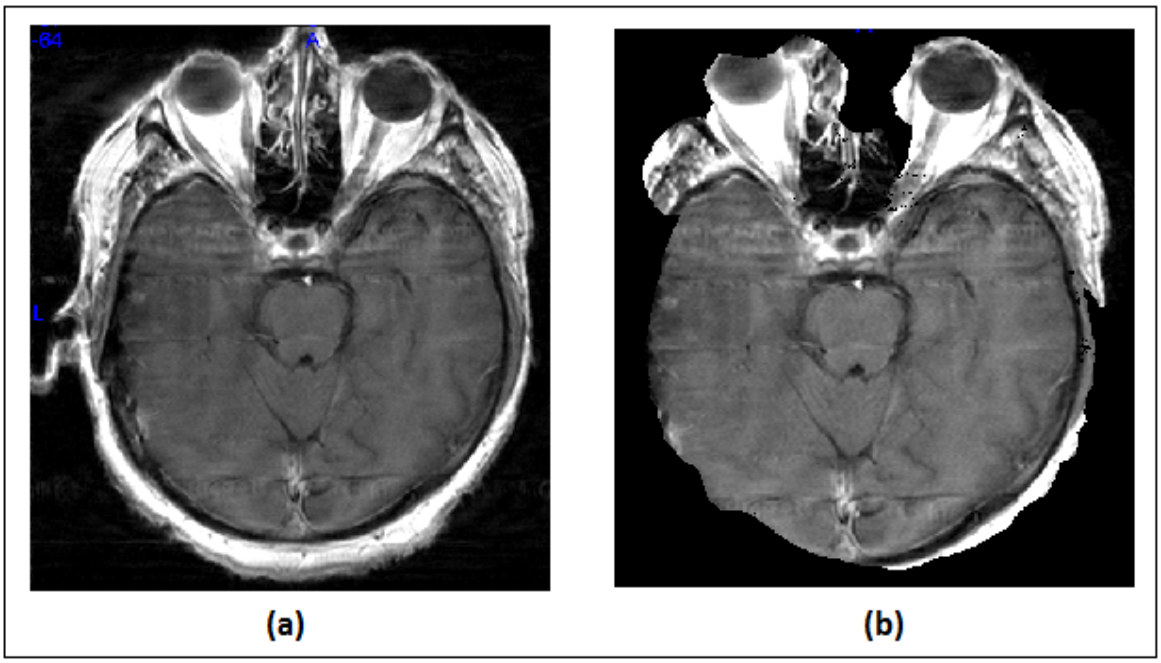


Fig. 5.3: (a) initial MRI image. The circumferential bright pixels are scalp and scull pixels. On top of the image we can observe the eyes of the patient. Between the two eyes , is off course the nose and all these pixels should be cropped. **(b)** The same image after we have applied the brain extraction tool with threshold 0.3. We observe that plenty scull and scalp pixels have not been cropped.

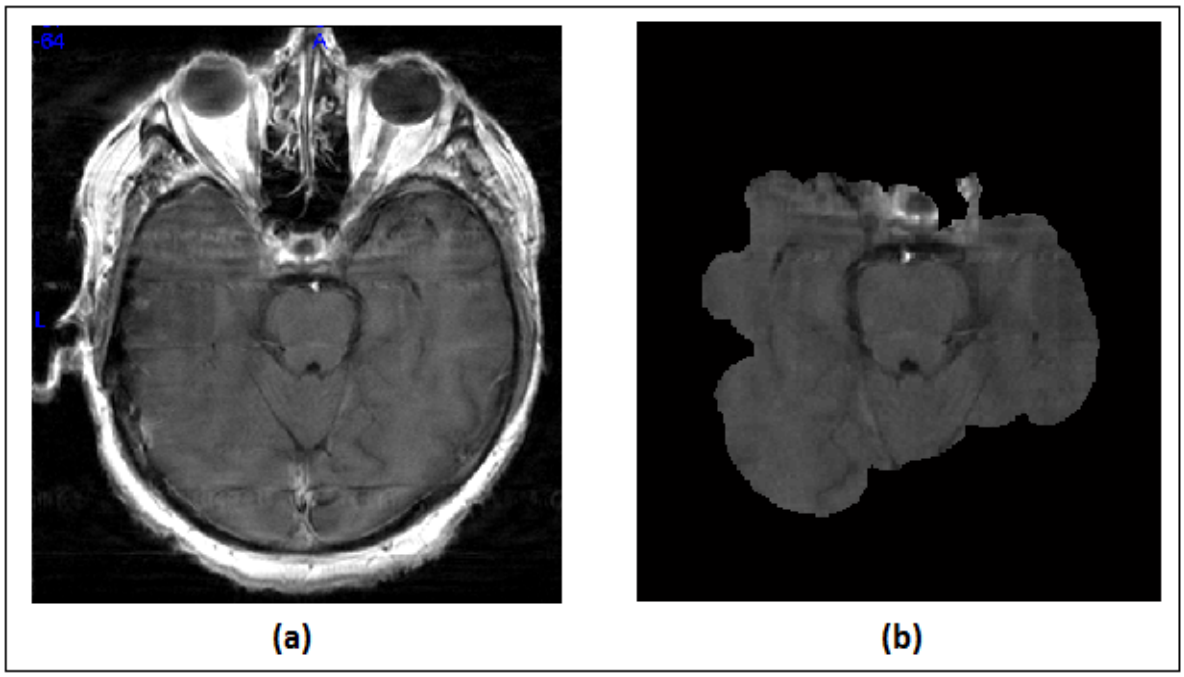


Fig. 5.4: (a) initial MRI image. The circumferential bright pixels are scalp and scull pixels. (b) The same image after we have applied the brain extraction tool with threshold 0.8. We observe that plenty brain tissue pixels have been wrongly cropped.

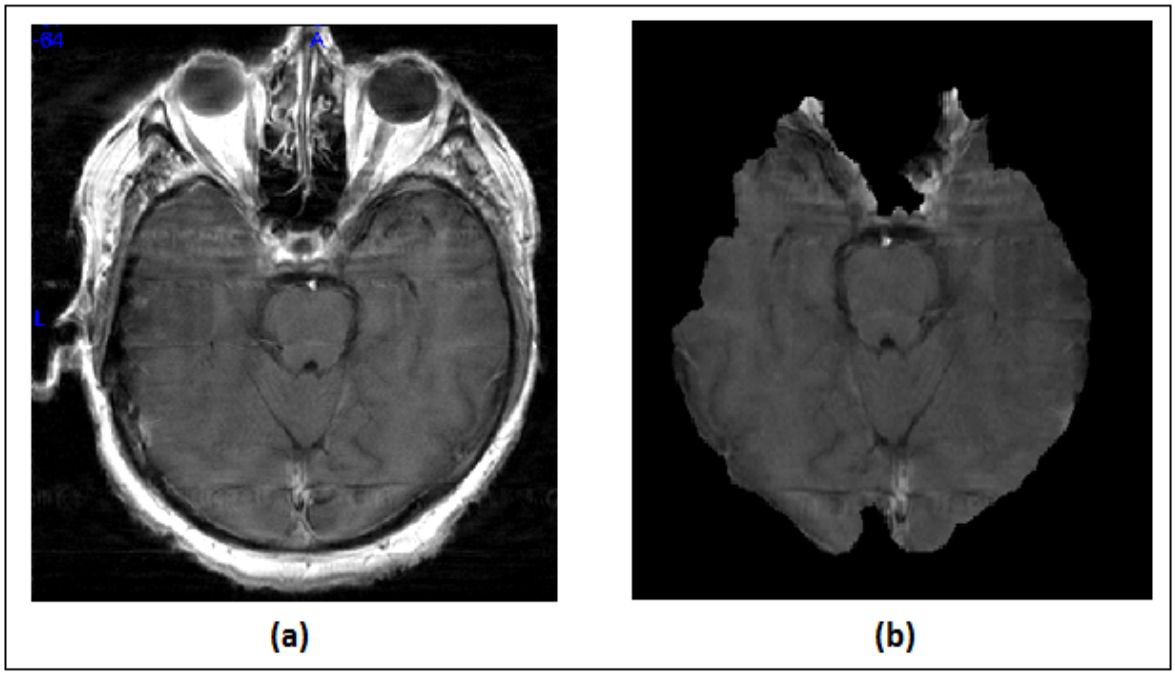


Fig. 5.5: (a) initial MRI image. The circumferential bright pixels are scalp and scull pixels. (b) The same image after we have applied the brain extraction tool firstly with threshold 0.4, secondly with threshold 0.5 and finally with threshold 0.55. We observe that the majority of scalp and scull pixels have been removed and simultaneously brain tissue pixels have not been cropped.

5.2.2 Median filter, Intensity Normalization, Background Extraction

After having extracted the scull and scalp pixels, now we apply a median filter in order to remove the additive noise. The median filter was preferred instead of Gaussian as it is discussed in the Experimental Results Section. The next step is the intensity normalization via linear histogram stretching based on the darkest and brightest percentage points. The normalization sets the darkest percent of pixels to zero and rescales the brightest percent to 4095. This step is necessary in order to guarantee that all the images of any MRI scan will have the same treatment and also to obtain similar dynamic ranges for all the three brain tissues. The last step of the preprocessing step is to estimate the background of every image, by performing a morphological opening operation [79]. After the estimation of the background, we remove it from the images and store **only** the brain tissue pixels in a matrix. This is done in order to perform the mean-shift algorithm using only the brain tissue pixels (if we were using the background pixels in the mean-shift procedure this would lead to a completely wrong processing of data in mean-shift clustering step and finally misclassification of the brain tissue pixels).

5.3 Mean-Shift Clustering Step

Our data are modeled in the following way:

We suppose that each brain tissue pixel of the MRI slice is a unique category. The feature space of our problem is both **spatial (x and y pixel's coordinates)** and **intensity**, for an overall dimensionality of **three**. So, we have to estimate the unknown probability density function (PDF) for each category, according to its neighborhood of pixels that act as samples. As already mentioned in Section 4.1, the most common way to estimate an unknown pdf, is the Parzen Windows method. According to Parzen Windows, an unkwown pdf can be estimated by the following equation:

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right) \qquad \textbf{(5.1)}$$

where d is the dimension of the feature space, in our case 3 as already mentioned, $\boldsymbol{x}$ is the feature vector of the pixel-category that we want to estimate its pdf, n is the number of neighborhood pixels-samples considered for the estimation of the pdf, x_i is the feature vector of a pixel-sample that belongs to the neighborhood of $\boldsymbol{x}$, h_i is the windows radius that we are taking into consideration (not all pixels of the neighborhood participate in the estimation of the pdf) and function k is named Kernel Profile and is actually an equation where the similarity of the characteristics between each pixel-sample of the neighborhood and the pixel that we want to estimate its pdf is taking into account. Pixels-samples that have similarly characteristics with the pixel that we want to estimate its pdf participate more than the pixels-samples that aren't so relative with the current pixel-category. In order to guarantee this discrimination, Kernel profile function must be a non-negative , non-increasing and normalized to one function. In this work, the Normal (Gaussian) Kernel Profile was preferred instead of the common Epanechnikov Kernel (Eq. 4.26), as it produces better results (comparison of these two kernels is presented in the Experimental Section) transforming Eq. 5.1 into:

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^3}\exp\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right) \qquad \textbf{(5.2)}$$

Although, we don't have to estimate the pdf of the pixels. As explained in Section 4.2.3, taking the gradient of equation 5.1 leads to the following result:

$$\hat{\nabla} f_{h,K}(x) \equiv \nabla \hat{f}_{h,k}(x) = \frac{2c_{k,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} (x - x_i) k' \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \quad \textbf{(5.3)}$$

By replacing $g(x) = -k'(x)$, Eq. 5.3 becomes:

$$\nabla \hat{f}_{h,k}(x) = \frac{2c_{k,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} (x_i - x) g \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \quad \textbf{(5.4)}$$

and finally:

$$\nabla \hat{f}_{h,k}(x) = \frac{2c_{k,d}}{n} \left[\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i g \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)} - x \right] \quad \textbf{(5.5)}$$

From Eq. 5.5 we are only interested in the following part, which is known as the **mean-shift vector**:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i g \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \quad \textbf{(5.6)}$$

Observing Eqs 5.5 and 5.6 we can come to the conclusion that the mean-shift vector is proportional to the normalized gradient of the density estimate. Therefore, as proved by Fukunaga and Hosteler [66], **the mean-shift vector points toward the direction of maximum density increase.** Back to our case, taking the gradient of Eq. 5.1, leads to the following mean-shift vector:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i \exp \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} \exp \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)} - x \quad \textbf{(5.7)}$$

and by replacing the dimensional of our feature space *d* with 3, as our feature space consists of pixel's spatial (x and y coordinates) and intensity information, Eq. 5.7 becomes:

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} \frac{1}{h_i^{5}} x_i \exp \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{5}} \exp \left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)} - x \quad \textbf{(5.8)}$$

We can observe that the left part of the mean-shift vector is a weighted-mean of the features vector of the pixels-samples of the neighborhood, of the pixel-category.

Starting from point $x^{(j)}$ in feature space, we move with the mean-shift vector to a point $x^{(j+1)}$,that lies in a higher density region than $x^{(j)}$, by repeating iteratively for $j=1,2,\ldots,$, computed as follows:

$$m_{h,G}(x_j)=x^{(j+1)}-x^{(j)}=\frac{\sum_{i=1}^{n}\frac{1}{h_i^5}x_i\exp\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)}{\sum_{i=1}^{n}\frac{1}{h_i^5}\exp\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)}-x^{(j)} \qquad \textbf{(5.9)}$$

until the difference $x^{(j+1)}-x^{(j)}$ is at the limit zero. Then, we will have reached the nearest stationary point, which is usually also one of the **local maximum**, of the pdf of the pixel-category.

We haven't defined yet the value of *n* parameter, the size of the neighborhood that we are taking into consideration for the calculation of the mean-shift vector for each pixel. Someone may think that the larger the neighborhood is, the better the results of the mean-shift clustering step. This assumption is wrong. It is not the size of the neighborhood that matters, but an appropriate neighborhood selection. In this work, a neighborhood size of [12 * 12] is utilized. In other words, n=144 neighbor pixels (12*12). Explanation of neighborhood queries is given in the Experimental Section. Neighborhood queries computed in Eq. 5.9 constitute a real bottleneck for the mean-shift algorithm. This is especially true for large datasets in high dimensional spaces, where a naïve computation of all the neighborhood queries at each mean-shift iteration is prohibitive. In order to solve this difficult problem, we have saved in a large matrix all the brain tissue pixels, and according to each pixel's coordinates, we are searching in a neighbor around of it, to find the appropriate neighborhood.

As for the selection of h_i , a simple method is to define the window size as the distance h_i between x_i and it's *k*-nearest neighbor:

$$h_i=\|x_i-x_{i,k}\| \qquad \textbf{(5.10)}$$

The neighbors of x_i are sorted by order of increasing distance to x_i . Following the ordering process, $x_{i,k}$ is the *k*-th distant neighbor from x_i , and h_i is its distance to x_i . The number of neighbors considered should be chosen large enough to ensure that there is an increase in density. In this work we have chosen the value $k=120$. We will show in the experimental section that in practice k can be chosen in a large interval of values without affecting significantly the quality of the results.

After locating the local maximum points with the the mean-shift vector for each pixel-category, we are now searching in small neighborhoods for the maximum of these local maximum points, defining in this way the **modes**. How small is a neighborhood that we are going to search, is subjectively. This factor has also major affect in the final results, as it affects the producing modes of the mean-shift procedure. Small neighborhood will lead to numerous modes (even > 2000) while large neighborhood will lead to few tens of modes (<100). Both facts lead to misclassification, because in the first case the mean-shift's procedure resulting modes are too many and as a result, the mean-shift procedure doesn't have significant effect in the procedure of segmentation. In the second

case, too few modes lead to merging pixels that may belong in the same brain tissue category. In this work, we chose a neighborhood of [5*5]. In other words, for each pixel we check if its local maximum is the maximum point in his neighborhood of [5*5] pixels. This choice leads to an amount from a few, to several hundreds of modes (it depends on the dataset, the amount of brain tissue pixels of the image).

The next step after locating the modes, is to categorize all brain tissue pixels of the MRI image to these modes. For the classification of the pixels, the Euclidean distance is used (Section 4.3.1), according to the three features (spatial and intensity) we have used in order to calculate the mean-shift vector.

In the end of Mean-Shift clustering step, from thousands of the initial pixels, now we have a few to several hundreds of modes. **Pixels that belong to the same mode can't change category** during the next steps of the proposed segmentation algorithm. In figure 5.6 it is shown a block diagram of the mean-shift clustering step and in figures 5.7 and 5.8 it is demonstrated a variety of Mean-Shift results, in order to get the picture of what was achieved.

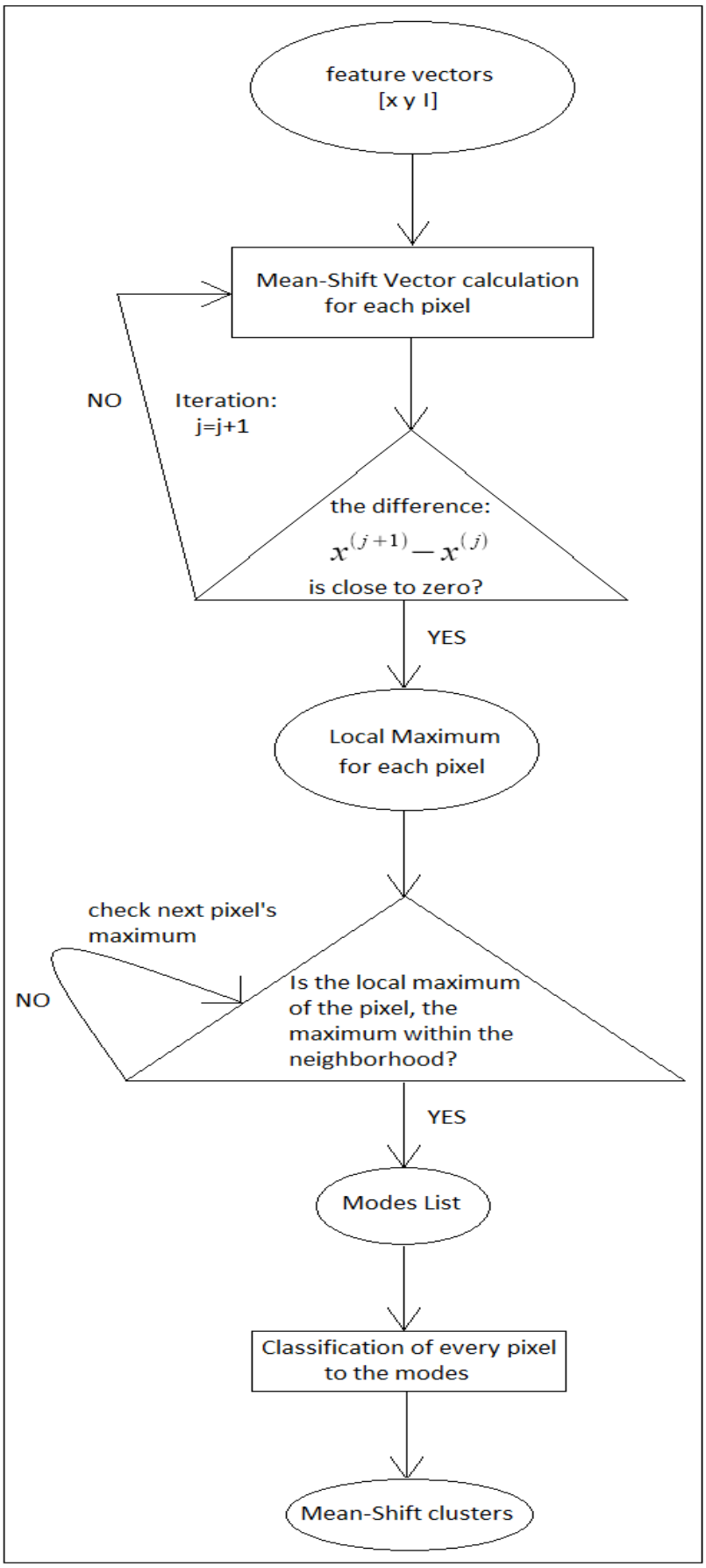


Fig. 5.6: Block diagram of the proposed mean-shift clustering procedure.

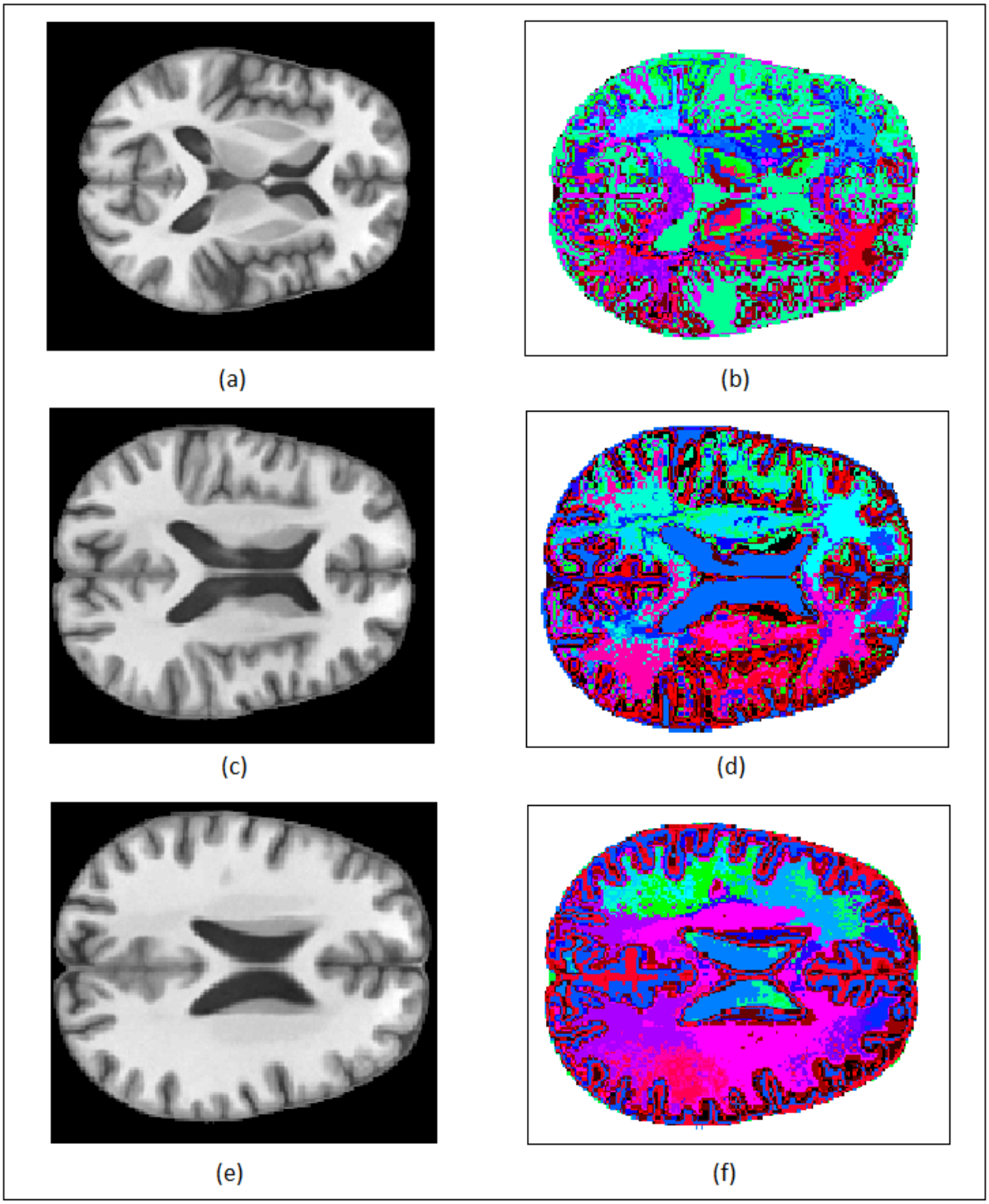


Fig. 5.7: In images (a), (c), (e) it is shown 3 MRI T1 brain slices and in images (b), (d), (f) the respective images after the mean-shift clustering step. In each image separately, same colored pixels correspond to the same cluster. There is a large compression of the initial data. In picture (b), from 17629 brain tissue pixels of image (a), after the mean-shift procedure there have been left 274 modes to still classify. In image (d), from 17348 pixels, after the mean-shift procedure we have 223 modes left and finally in image (f), from 17005 pixels, now we have 226. All MRI slices in this figure are simulated images, downloaded from [80].

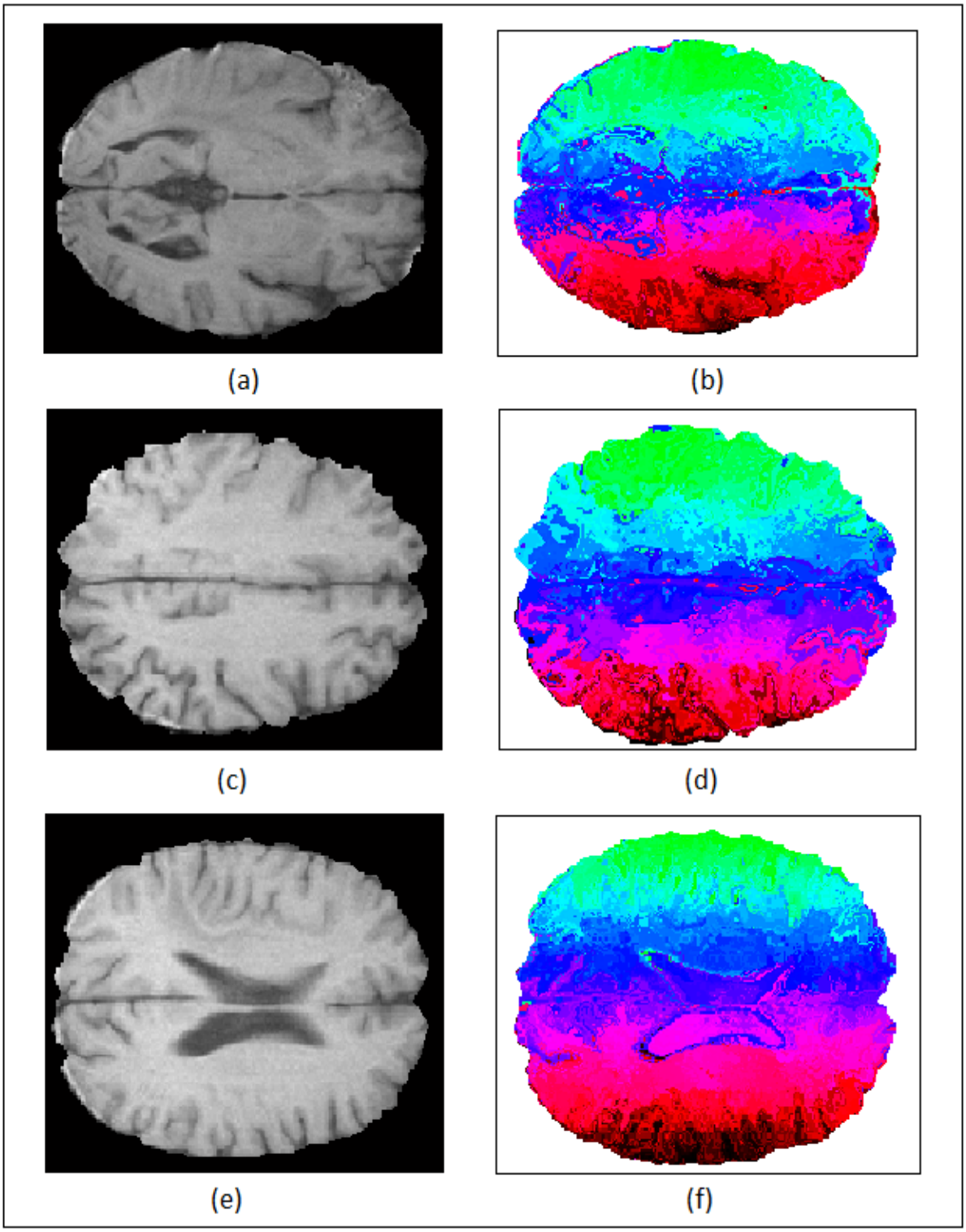


Fig. 5.8: Images (a), (c), (e) show 3 MRI T1 brain slices and images (b), (d), (f) the respective images after the mean-shift clustering step. In each image separately, same colored pixels correspond to the same cluster. In picture (b), from 77263 brain tissue pixels of image (a), after the mean-shift procedure there have been left 803 modes to still classify. In image (d), from 62949 pixels, after the mean-shift procedure we have 570 modes left and finally in image (f), from 74127 pixels, now we have 609 modes. In this figure, the data compression isn't as large as in fig. 5.7. due to the huge number of brain tissue pixels of the MRI images. All MRI slices in this figure are from real dataset.

5.4 Mahalanobis Pruning Modes Step

The number of remaining modes after the mean-shift procedure is a large compression of the initial data but it is still much larger than the targeted number of classes. Empirically, if the number of remaining modes are more than 400, a mode pruning step is therefore required. In fact, we have used the nonparametric adaptive mean-shift for clustering in the joint spatial-intensity feature space as the clusters are inherently nonconvex. For the pruning of the modes, however, we switch to an intensity-only feature space. For this purpose, a pruning mechanism is added as follows. A fixed-radius window is shifted across the intensity feature space (ignoring spatial features), centered on each mode. Modes that co-exist within the window are merged. Mahalanobis distance is utilized for the distance computation. For the computation of the Mahalanobis distance, a covariance matrix is computed per mode from the intensity values of its corresponding pixels. Therefore, for two intensity vectors, I_m and I_n , representing the intensity components for two convergence modes *m* and n respectively, *m* and *n* are merged if:

$$min\left(MahalDist\left(I_m, I_n\right), MahalDist\left(I_n, I_m\right)\right) < R \qquad \textbf{(5.11)}$$

where *MahalDist* is the Mahalanobis distance between vectors I_m and I_n, and *R* is the window radius.Every time there is a mode pruning, the covariance matrices (for the merged modes) are updated immediately. In practice, the initial window size, is set to 1. The increment between iterations, is set to 0.5 .

The process is repeated iteratively with an increasing window radius, until the remaining modes are less than 400. Further merging with the Mahalanobis pruning modes step is not suggested, as discussed in the Experimental Section, because of the fact that it doesn't improve significantly the results. Thus, if we over-merge the modes with this procedure, that is to say the number of remaining modes become less than 150, we would increase the possibility of misclassification, as we may merge modes that should have stayed separately.

In figure 5.9 we demonstrate the results from this second step. We can observe that we are getting closer to the final segmented result.

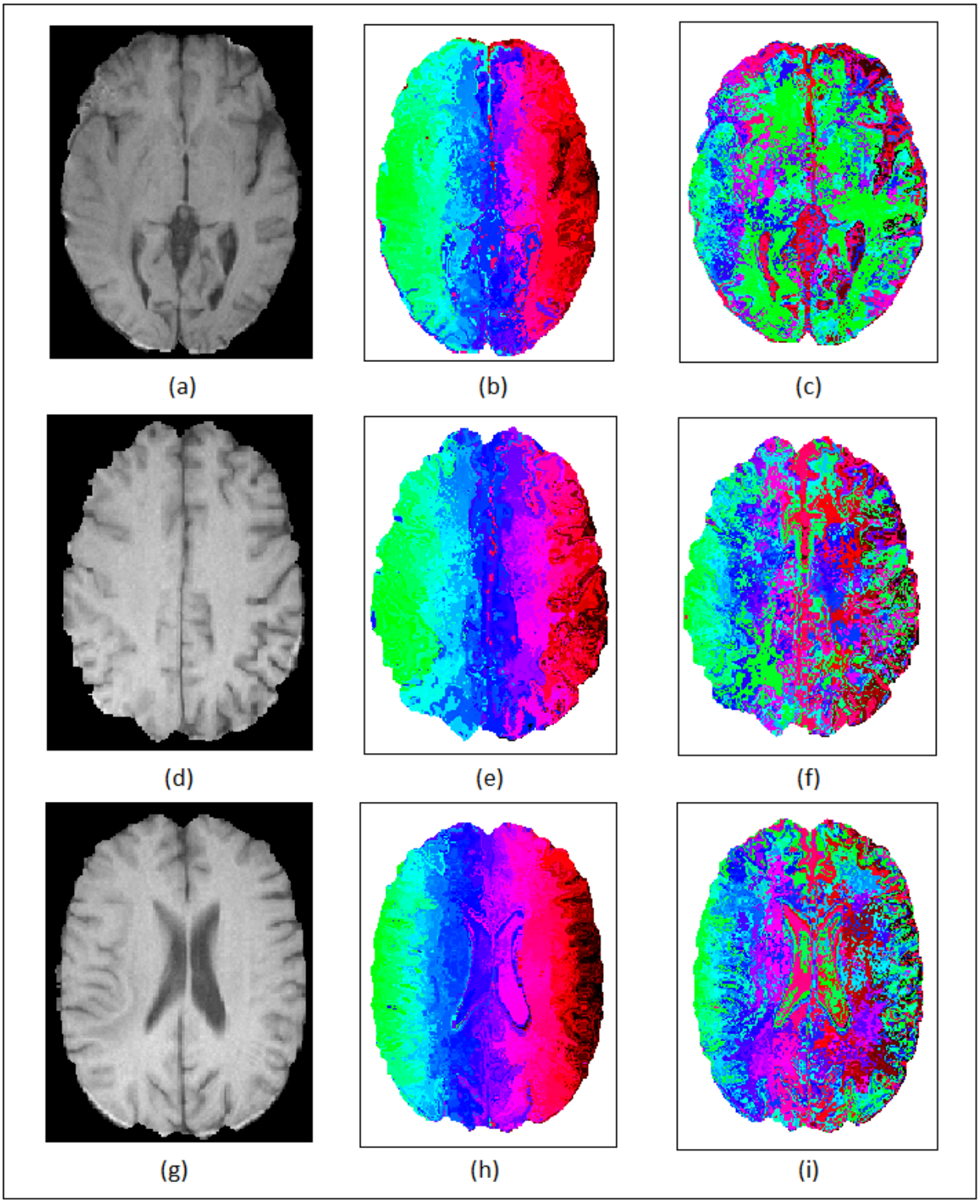


Fig. 5.9: In images (a), (d), (g) it is shown the initial MRI slices, in images (b), (e), (h) the respective images, after the mean-shift clustering step and finally in images (c), (f), (i) the respective images after the mahalanobis pruning modes step. In image (c), from 803 modes left after the mean-shift clustering step, now, after the mahalanobis pruning modes step there have been left 337 modes to still classify. In image (f), from 570 modes now we have 373 and finally in image (i), from 609 modes now there have been left 400 modes. In each image separately, same colored pixels correspond to the same cluster and cannot change during the next, final step. All images in this figure are from a real dataset.

5.5 Fuzzy C-means Step

The remaining modes are assigned to the desired tissue classes by clustering their intensity values using the fuzzy K-means clustering algorithm. In section 5.6, we propose a way to approach brain tumor and edema pixels, with the mean-shift algorithm. In this final step, the following procedure, according to the fuzzy k-means theory presented in Section 4.5, is executed:

i. Definition of the number of clusters ***c***. In our case, the number of clusters depends on the existence of tumor or edema pixels in the image. If it doesn't contain tumor or edema pixels, then the appropriate number of clusters is 3, as the number of brain tissues. If it does, then we need 4 clusters, three for the tissues and one for the tumor or edema pixels. In section 5.6, we propose a way to approach brain tumor and edema pixels. The initialization of the centers of the fuzzy k-means algorithm, in both cases, is set according to the histogram of the MRI image. We remind that the histogram of the image shows the distribution of the pixels of the image. So we locate in the histogram ***c*** points, with high distribution, from all the intensity range, and these values are our initial cluster centers. In figure 5.10 it is shown an example of histogram and cluster center initialization.
ii. Definition of the initial values of the participation matrix. In this work, the participation (or membership) matrix is initialized in the following way: All pixels of the remaining modes are classified, only in the first iteration, according to the simple k-means algorithm (Section 4.4). Then we calculate the membership matrix for each mode according to the equation:

$$u_{ik} = \frac{\frac{1}{(|x_k - v_i|)^2}}{\sum_{j=1}^{c} [\frac{1}{(|x_k - v_j|)^2}]^{1/(m-1)}} \quad , i = 1,\dots, c \text{ and } k = 1,\dots, n \qquad \textbf{(5.12)}$$

where ***c*** is the number of clusters, $x_{1,} x_{2,\dots}, x_n$ are the elements of the mode that we want to classify, $V = \{v_{1,} v_{2,\dots}, v_c\}$ are the center of the clusters, $U = [u_{ik}]$ is a $c * n$ matrix where u_{ik} is the percentage of participation of *k* pixel in the *i* cluster and finally *m* is a predefined exponential fuzzifier factor, here is set to 2. This factor though, doesn't affect significantly the final segmentation results.
iii. For all the *n* elements of each mode, we calculate their distance from the center of the ***c*** clusters, and we place them to the appropriate cluster, according to the minimum distance from the ***c*** clusters.
iv. We calculate the ***c*** centers of the clusters according to the following equation:

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_{ik}}{\sum_{k=1}^{n} u_{ik}^m} \quad , i = 1,\dots, c \qquad \textbf{(5.13)}$$

(mean value of x_{ik} with weights u_{ik}^m)

where ***c*** is the number of clusters, $x_{1,} x_{2,\dots}, x_n$ are the pixels of the image, u_{ik} is the percentage of participation of *k* pixel in the *i* cluster and finally *m* is the fuzzifier factor, here is set to 2.
v. We calculate the membership matrix for each mode according to Eq. 5.12:

$$u_{ik} = \frac{\frac{1}{(|x_k - v_i|)^2}}{\sum_{j=1}^{c} [\frac{1}{(|x_k - v_j|)^2}]^{1/(m-1)}} \quad , i = 1,\ldots, c \text{ and } k = 1,\ldots, n$$

where ***c*** is the number of clusters, $x_{1,} x_{2,\ldots}, x_n$ are the elements of the mode that we want to classify, $V = \{v_{1,} v_{2,\ldots}, v_c\}$ are the center of the clusters, $U = [u_{ik}]$ is a $c * n$ matrix where u_{ik} is the percentage of participation of k pixel in the i cluster and finally m is a predefined exponential fuzzifier factor, here is set to 2.

vi. We repeat steps iii – v until there is no change in the centers of the clusters.

The output of the algorithm is a segmentation map of the three brain tissue types, and a membership matrix which represents the percentage of participation (probability) for each pixel to belong in each tissue.

In figures 5.11-5.12 the final segmented results of various images are illustrated, after the use of the mean-shift, mahalanobis pruning and fuzzy k-means procedures.

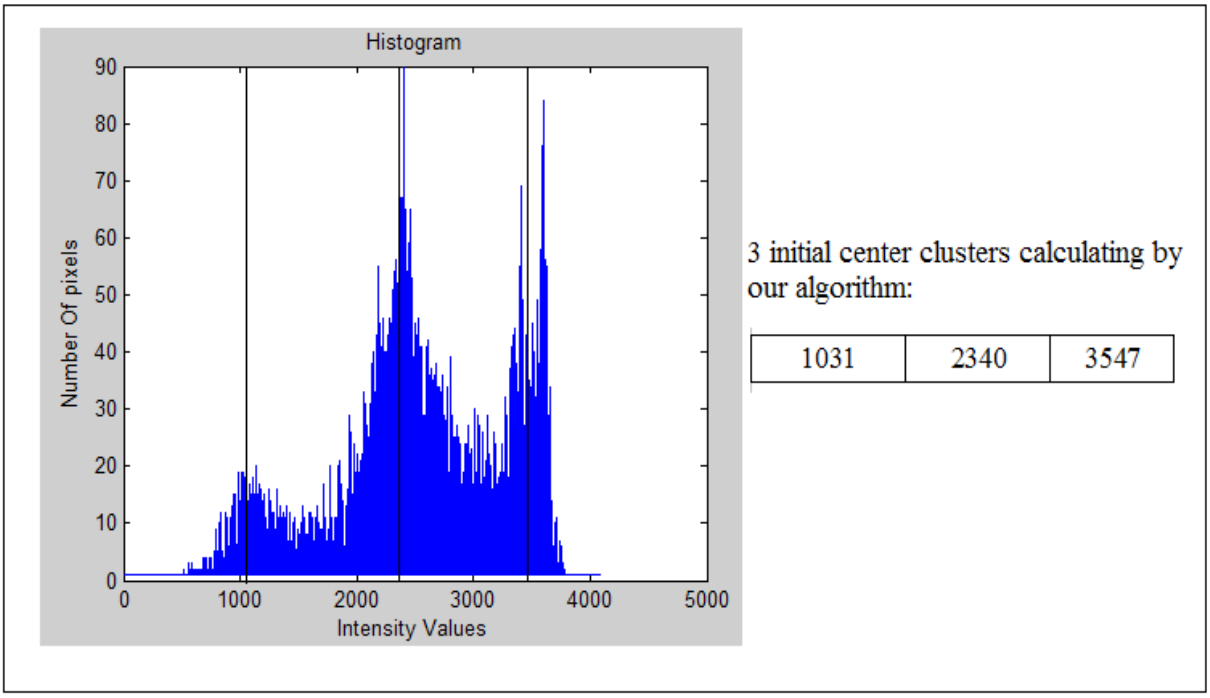


Fig. 5.10: Histogram of an MRI brain slice and the initial centers of the fuzzy k-means, calculated by our algorithm, according to the three main lobes of the histogram. These centers represent a good initialization in all the range of intensity.

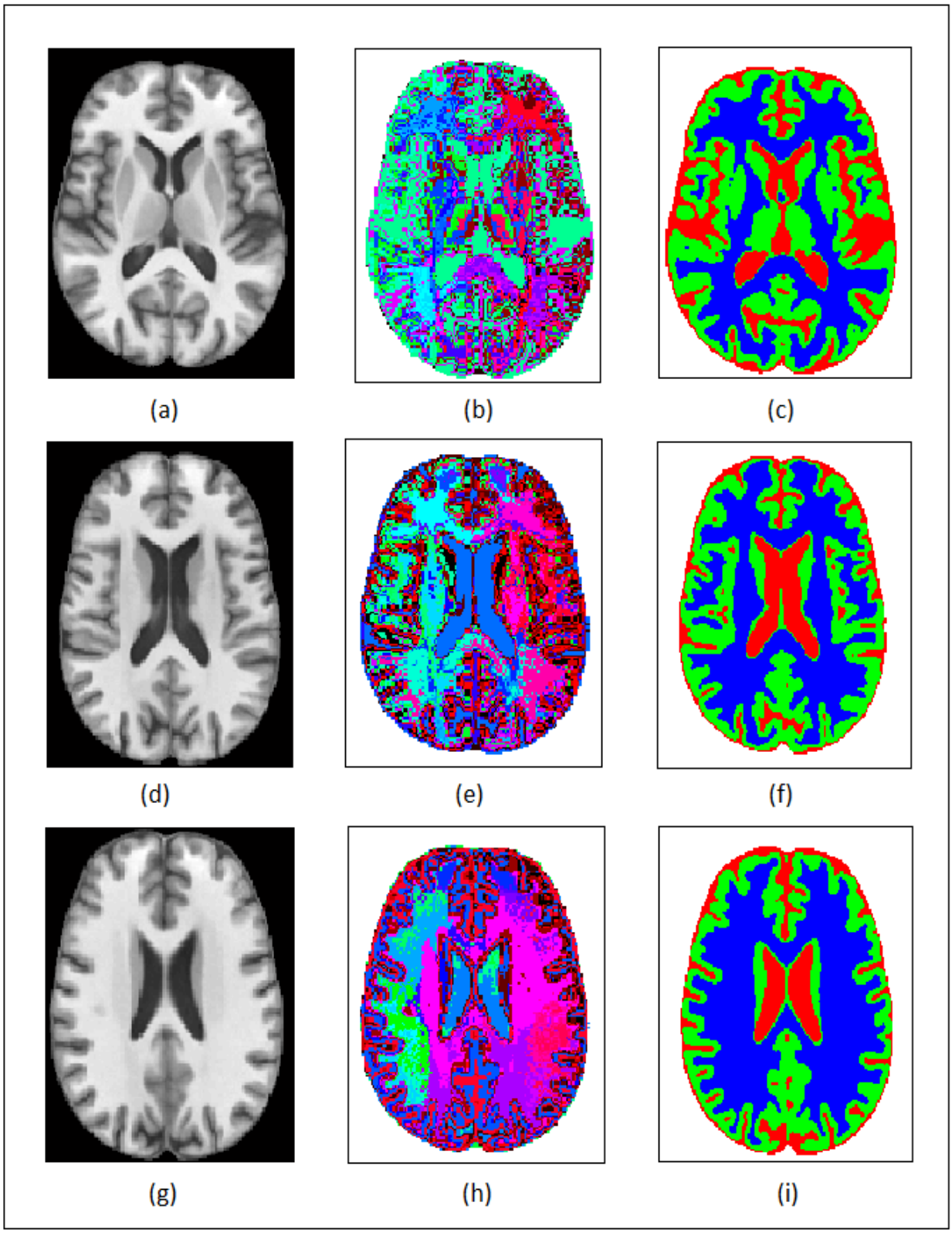


Fig. 5.11: In images (a), (d), (g) it is shown the simulated MRI initial images, in images (b), (e), (h) the images after the mean-shift clustering step and in images (c), (f), (i) the final segmented results. In images (c), (f) and (I), CSF tissue pixels are appeared in red color, WM in blue and GM in green color.

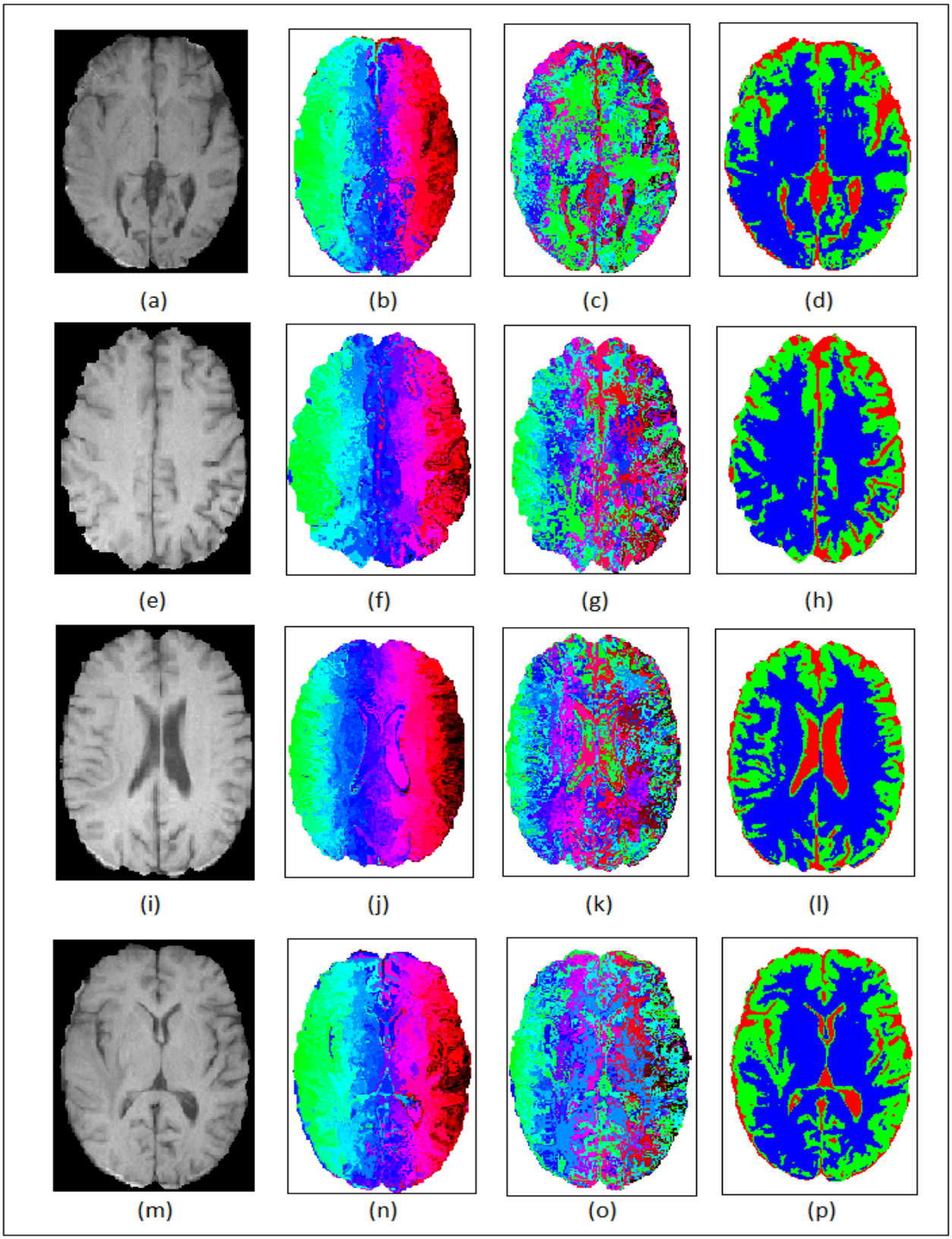


Fig. 5.12: In images (a), (e), (i), (m) it is shown the initial MRI images, in images (b), (f), (j), (n) the result after the mean-shift clustering step, in images (c), (g), (k), (o) the images after the mahalanobis pruning step and finally in images (d), (h), (l), (p) the images after the fuzzy c-means step where we can observe the final segmented results in which CSF tissue pixels are appeared in red color, WM in blue and GM in green color. All images in this figure are from real datasets.

5.6 Tumor and Edema Pixels Recognition

The Mean-Shift clustering algorithm, in combination with prior knowledge about the nature of the T1 and T2 MR images can be used, not only to classify the various brain tissues, but also to detect possible cancer cells. Because of the nature of the enhanced T1-weighted modality, the tumor necrotic area appears hypointense, while the solid area of the tumor around the necrotic area appears hyperintense and the edema cannot be distinguished from the GM and the WM, which both share medium intensities. Similarly, in T2-flair images, the edema and the solid tumor area appear hyperintense, while the necrotic area appears hypointense. An example of these two modalities is shown in Fig. 5.13 (a) and (b), where the skull has been removed manually from the two registered images. Using that information, we can perform clustering on two corresponding registered T1-enhanced and T2-flair images, using 4 clusters. After identifying the cluster with the higher mean value, since it is the most probable to include the tumor region, we remove the connected components having total area below a certain threshold. The largest connected component will be most likely the tumor area, in both MR modalities. If we subtract the tumor area of T1 from the tumor area of T2-flair, then we obtain the edema region. An example of this is shown in Fig. 5.13 (c), where the tumor area obtained from T1 (after morphological closing, in order to include the necrotic area, as well) is highlighted with green, superimposed on the original image, while in (d) the edema is also shown with blue, superimposed on the T2-flair image. In figure 5.14 we can observe various images that contain tumor or edema pixels and the results produced by the proposed procedure.

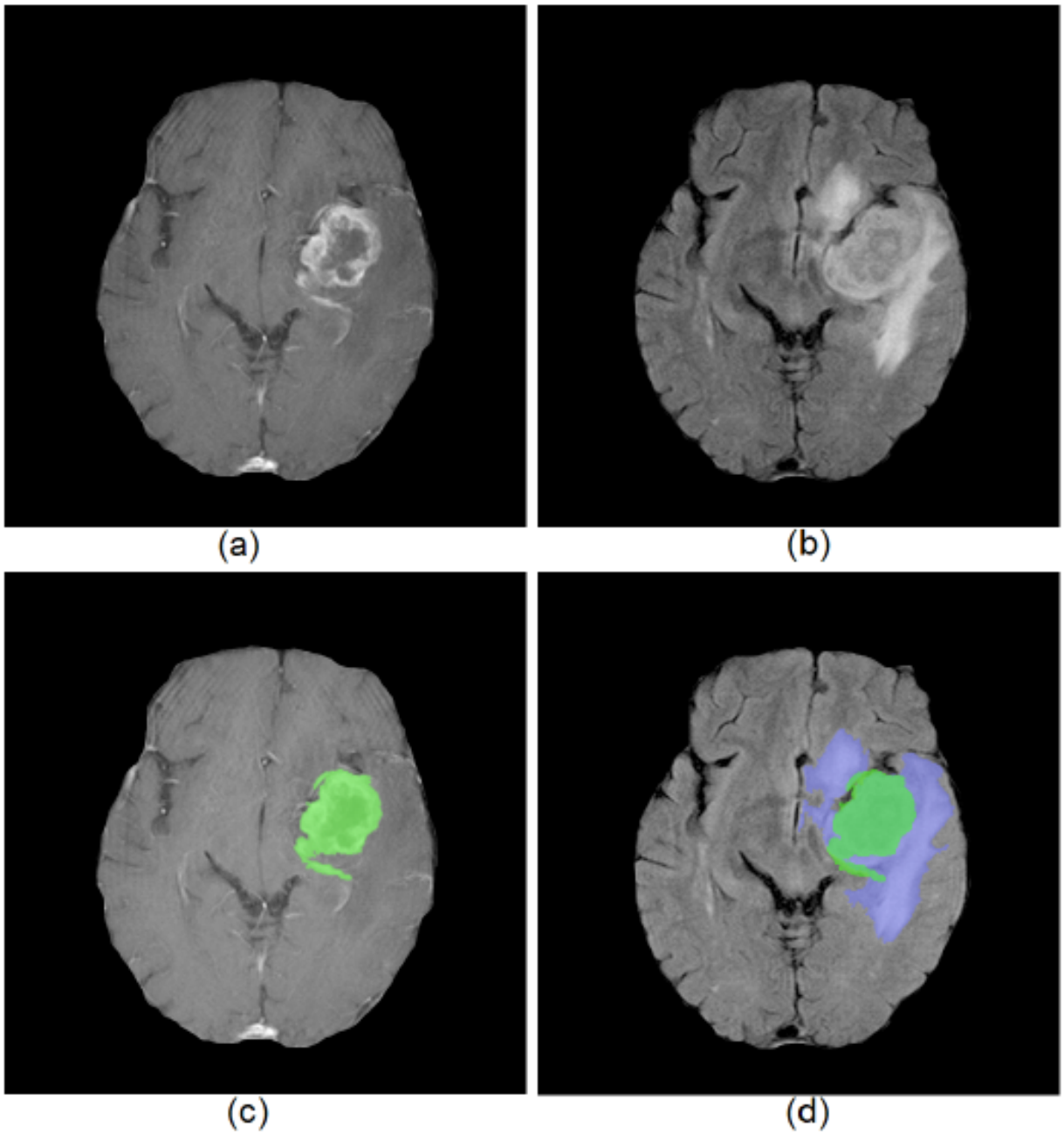


Fig. 5.13: (a) Enhanced T1-weighted image after the skull has removed, (b) Corresponding registered T2-flair image, (c) Solid tumor area obtained from enhanced T1, (d) Solid tumor area and edema obtained from T2-flair.

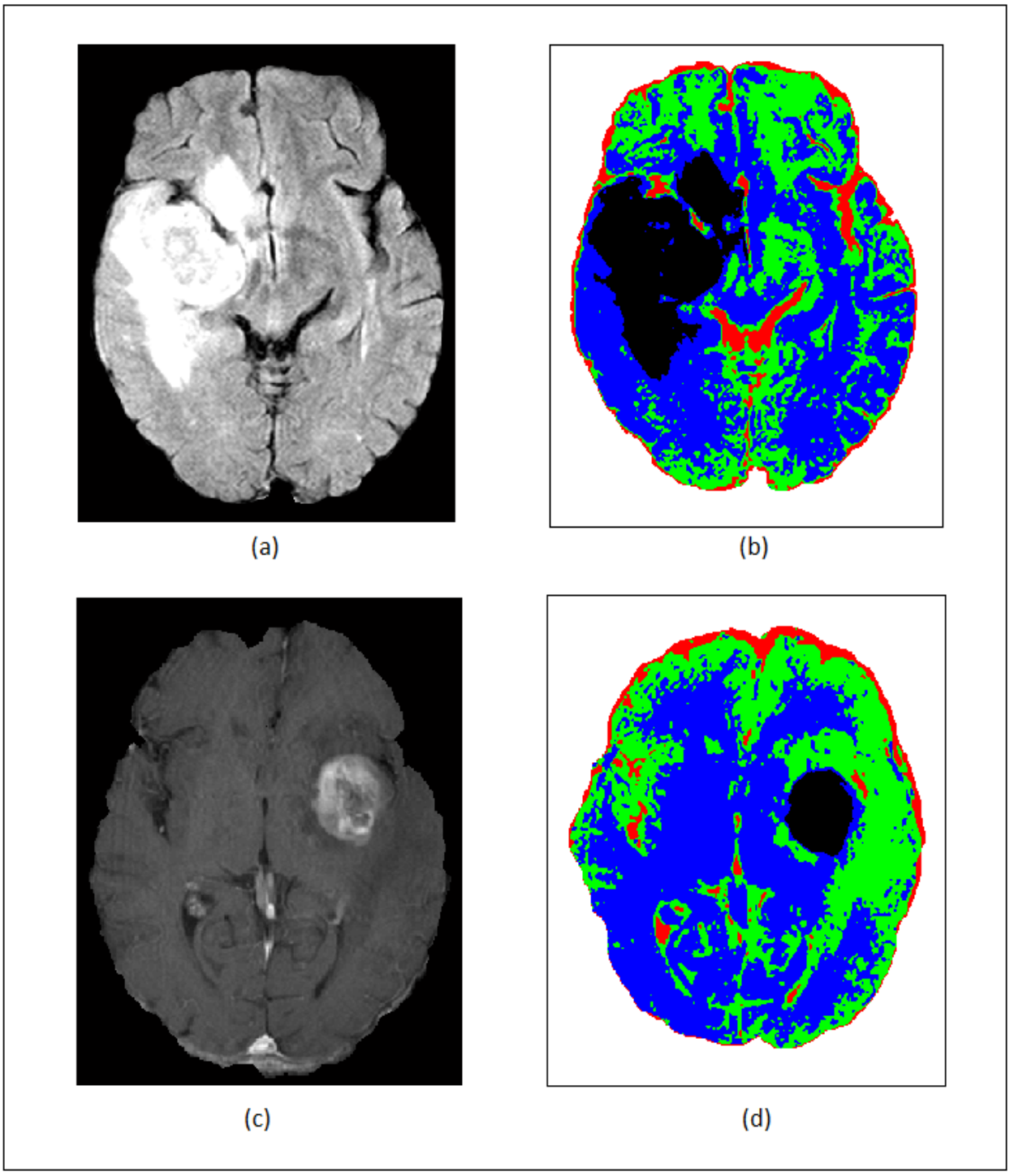
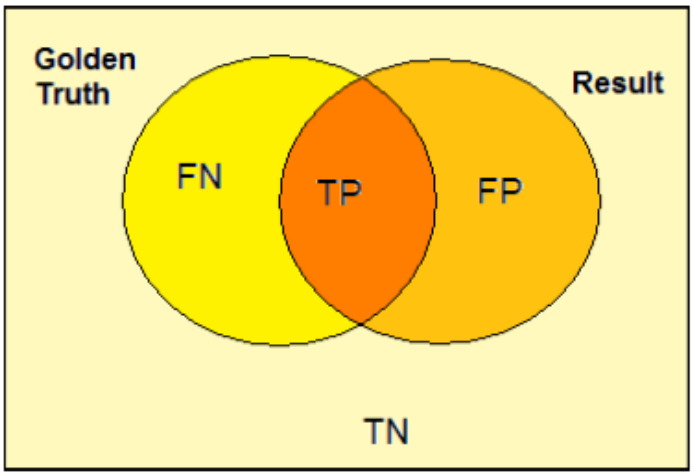


Fig. 5.14: (a) The initial T2 modality picture. Edema is appeared hyperintense. (b)The segmented final image using 4 clusters with the mean-shift procedure and afterward automated remove of irrelevant components. (c) The initial T1-enhanced modality picture and (d) the same,segmented picture.

6. EXPERIMENTAL RESULTS

6.1 Introduction

In order to make an evaluation of our proposed algorithm, it was necessary to check the algorithm in a dataset in which the golden truth was available. Golden truth is named the method where a doctor categorizes by himself all the pixels of the dataset, in the three categories, CSF, GM and WM. At this point, we demonstrate some metrics where there are defined ways to grade the proposed algorithm. All metrics are based on the following definitions:



where:

- True Positive (TP) is the number of pixels that the algorithm correctly segmented them in the right category.
- False Negative (FN) is the number of the pixels that the algorithm should but unfortunately failed to segment in the right category.
- False Positive (FP) is the number of the pixels that the algorithm mistakenly segmented them to the same category with the pixels of the TP category.
- True Negative (TN) is the number of pixels that correctly the algorithm didn't segment them to the same category with the pixels of the TP category.

With these definitions, a great variety of metrics can be defined, in order to evaluate a segmentation algorithm:

- False Positive Rate (α) = FP / (FP + TN), False Negative Rate (β) = FN / (TP + FN)
- Sensitivity (Power)=1-β, Specificity=1-α
- Likelihood-ratio Positive =sensitivity / (1 − specificity)
- Likelihood-ratio Negative = (1 − sensitivity) / specificity
- Jaccard similarity *(JC)* coefficient is a statistic used for comparing the similarity as is defined as JC=*TP/(FP+TP+FN)* .The bigger this value is, the better for the quality of the algorithm, as it means that the amount of TP pixels , is much bigger than the amount of pixels that the algorithm should but failed to segment in the right category or he algorithm mistakenly segmented them to the same category with the pixels of the TP category.

- *Dice Similarity = DS =2TP/(2TP+FP+FN)* .Dice Similarity attains the value of one if both segmentaions (Golden Truth and the evaluated algorithm) fully agree and zero if there is no overlap at all.
- *Tanimoto (TN)* is also a metric of similarity related to Jaccard Coeffiient, also known as extended Jaccard coefficient and it is defined as:
  TN=*(TP+TN)/(TP+2FP+2FN+TN)* .As we observe from the definition of *Tanimoto,* the closer to one is the value of Tanimoto, the better the evaluated algorithm. Tanimoto expresses the amount of pixels that both segmentations agree as for the amount of pixels that they are in one or the other segmentation, but not in both of them.
- Segmentation Accuracy (SA) has a great similarity with Tanimoto and it is defined as: *SA=(TP+TN)/(TP+TN+FP+FN)* .It expresses the amount of pixels that both segmentaions (Golden Truth and the evaluated algorithm) agree. Segmentation Accuracy attains the value of one if both Golden Truth and the evaluated algorithm fully agree and zero if there is no overlap at all.

In the various experiments that we perform, we chose to demonstrate two metrics in order to evaluate the various experiments with the proposed algorithm. In the end of the section, we will demonstrate the final results using the metrics. The experiments performed are the following:

- The effect of the neighborhood size for each pixel, in calculating the mean-shift vector
- The effect of parameter K ,whose value affects the significance of each pixel, in estimating the mean-shift vector.
- The effect of which kernel should we use. Normal or Uniform?
- The effect of additive noise effect
- The effect of ignoring random pixels, in order to gain time.
- The effect of the Mahalanobis Pruning Modes Step usage
- The effect of K-means and Fuzzy K-means.
- Summary of optimal solutions
- Efficiency of algorithm in other modalities
- Comparison with other segmentation methods

6.2 The effect of neighborhood size for each pixel

We remind that the Eq. 4.24 :

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right) \qquad \textbf{(6.1)}$$

estimates the Probability Density Function (pdf) for each pixel according to the Parzen Windows theory (Section 4.1). In this section, we investigate how the amount of the neighbor pixels, (the parameter *n* of the equation) affects the percentage of True Positive and Dice Similarity, and though the quality of the results of the proposed algorithm. In all the experiments we used a constant value for k=120 and Gaussian Kernel. We must emphasize though, that a joint investigation of neighbor size and the value of *k* is discussed in Section [6.4]. Figure 6.1 illustrates the experimental results for the Dice Similarity metric, while in figure 6.2 it is shown he experimental results for True Positive. In Table 6.1 is shown the Dice percentage values for all images for each neighborhood for CSF, GM,WM.

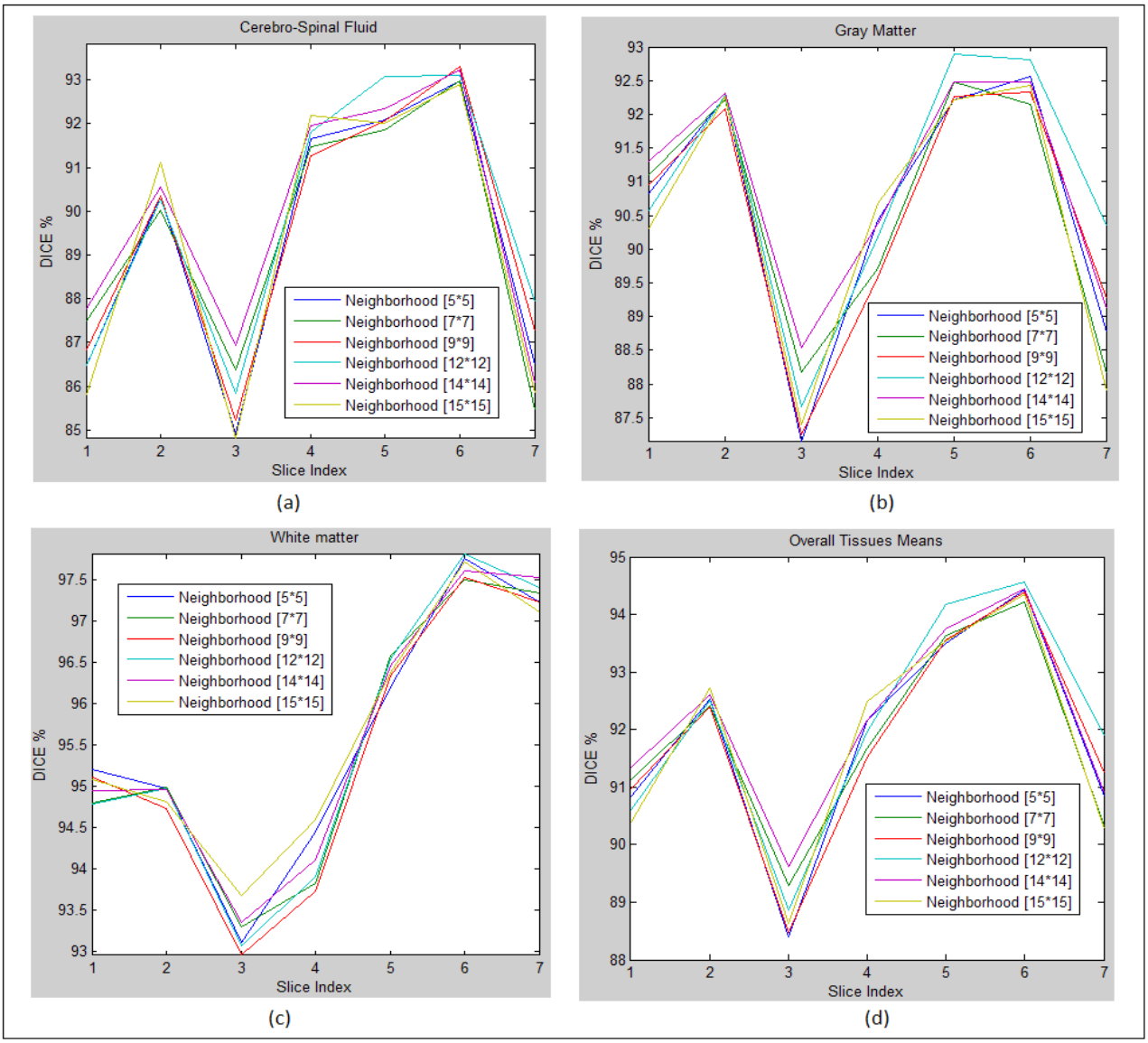


Fig. 6.1: (a) The Dice percentage of Cerebro Spinal Fluid for 7 MRI slices for 6 different neighborhoods (b) The Dice percentage of Gray Matter for the same slices and neighborhoods (c) The Dice percentage of White matter for the same slices and neighborhoods. (d) The overall mean values for the 7 MRI slices, for the 6 different neighborhoods that they were investigated. As a conclusion we can point out that the differences between the neighborhoods are small, with neighborhoods 12*12 and 14*14 to have an edge on better results.

Neighborhood	5*5	7*7	9*9	12*12	14*14	15*15
Dice CSF %	89,308	89,466	89,485	89,833	89,942	89,273
Dice GM %	90,583	90,600	90,493	90,884	90,940	90,470
Dice WM %	95,729	95,659	95,561	95,690	95,745	95,780
Overall %	91,812	91,808	91,792	92,083	92,117	91,772

Table 6.1: The Dice percentage (from all seven images presented in figure 6.1) for each Brain Tissue. Once again we observe that the differences are small between the neighborhoods, with neighborhoods 12*12 and 14*14 to have an edge on better results.

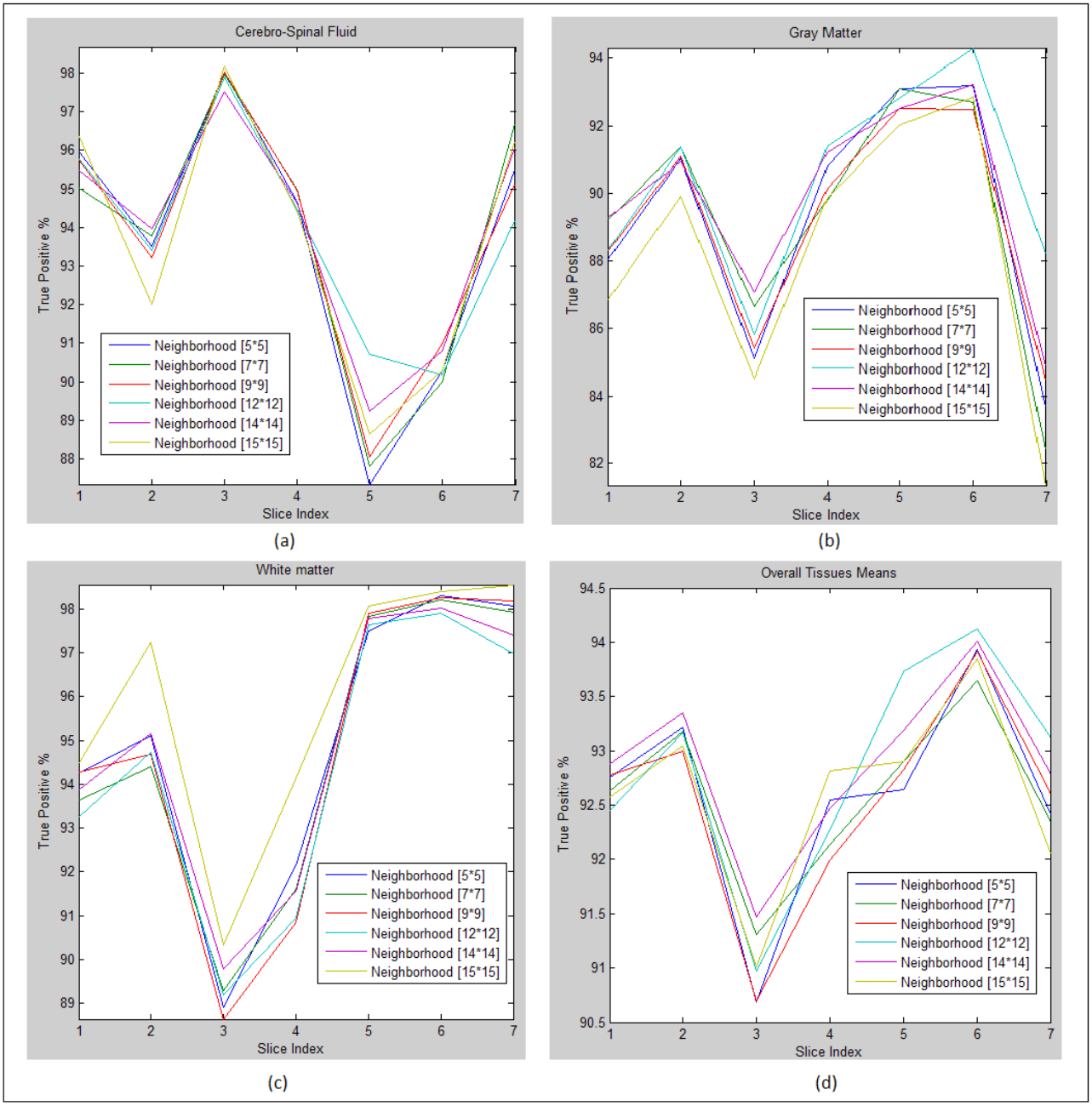


Fig. 6.2: Here there are presented the True Positive percentages , for the same 7 MRI brain slices as in the fig 6.1, for the three brain tissues, CSF, GM, WM. Here, we can exclude the conclusion, that the neighborhoods 12*12 and 14*14 produces the optimal results.

Neighborhood	5*5	7*7	9*9	12*12	14*14	15*15
TP CSF %	93,616	93,749	93,727	93,801	93,955	93,729
TP GM %	89,277	89,323	89,217	90,323	89,881	88,205
TP WM %	94,908	95,706	94,688	94,379	94,803	95,888
Overall %	92,600	92,597	92,544	92,834	92,879	92,607

Table 6.2: The True Positives percentage values (from all seven images presented in figure 6.1) for each Brain Tissue.. Once again we observe that the differences are small between the neighborhoods, with neighborhoods 12*12 and 14*14 to have an edge on better results.

We were expected that the bigger the neighborhood is, the better the results. From figures 6.1-6.2 and tables 6.1-6.2 we can point out that this fact isn't necessary right, that even small neighborhoods can produce equivalent good results with bigger neighborhoods. So it is not a matter of fact the size of the neighborhood, but the **appropriate** neighborhood, in order to produce the optimal results. In this work, we have preferred a neighborhood of [12*12] as it produces almost equal results with neighborhood of [14*14] but it takes less run time , as the proposed algorithm (including all stadiums Mean-Shift-Mahalanobis-Fuzzy K-means) needs 16.5 minutes to be executed with a neighborhood of [12*12] while with a neighborhood of [14*14] it needs 26 minutes (more neighborhood queries for each pixel).

6.3 The effect of *k* parameter

We remind that the Eq.(6.1) :

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)$$

estimates the pdf for each pixel according to the Parzen Windows theory (Section 4.1). In this equation, h_i is called the kernel bandwidth or window size, and determines the range of influence of the kernel located in x_i and it's influence on the obtained results is very significant.Many methods exist to determine an adaptive window size for the Adaptive Mean-Shift algorithm.Asimple method is to define the windowsize as the distance, h_i , between x_i and its *k*-nearest neighbor $x_{i,k}$:

$$h_i = \| x_i - x_{i,k} \| \qquad \textbf{(6.2)}$$

The neighbors of x_i are sorted by order of increasing distance to x_i. Following the ordering process $x_{i,k}$, is the *k*-th distant neighbor from x_i, and h_i is its distance to x_i . The number of ***k*** neighbors considered for h_i should be chosen large enough to ensure that there is an increase in density within the support of most kernels. In this section we investigate the Dice and True Positives results, for various values of ***k***, with an upper limit of 225, which corresponds to a neighborhood of 15*15.That's because when we are searching for a neighborhood for a pixel x_i in order to calculate it's *k*-th distant neighbor from x_i , we save it's neighborhood for the later mean-shift vector queries. The upper neighborhood size allowed is [15*15] because for larger neighborhoods the results aren't better than those presented in the previous 6.2 section , and indeed in huge datasets, with a number of pixels even more than 100000, there may be presented memory problems, in relevance with the platform this algorithm is executed. In figure 6.3 it is shown the Dice percentage for various values of ***k***, while respectively, in figure 6.4 it is presented the True Positive percentages. The Dice and True Positive percentage for all 7 images for each ***k*** value is demonstrated in table 6.3 and 6.4 for respectively, and the figures produced by these tables in order to estimate the optimal ***k*** value is presented in figures 6.5 and 6.6 for Dice and True Positive respectively. In all the experiments we have used Gaussian kernel, in a neighborhood of [12*12].

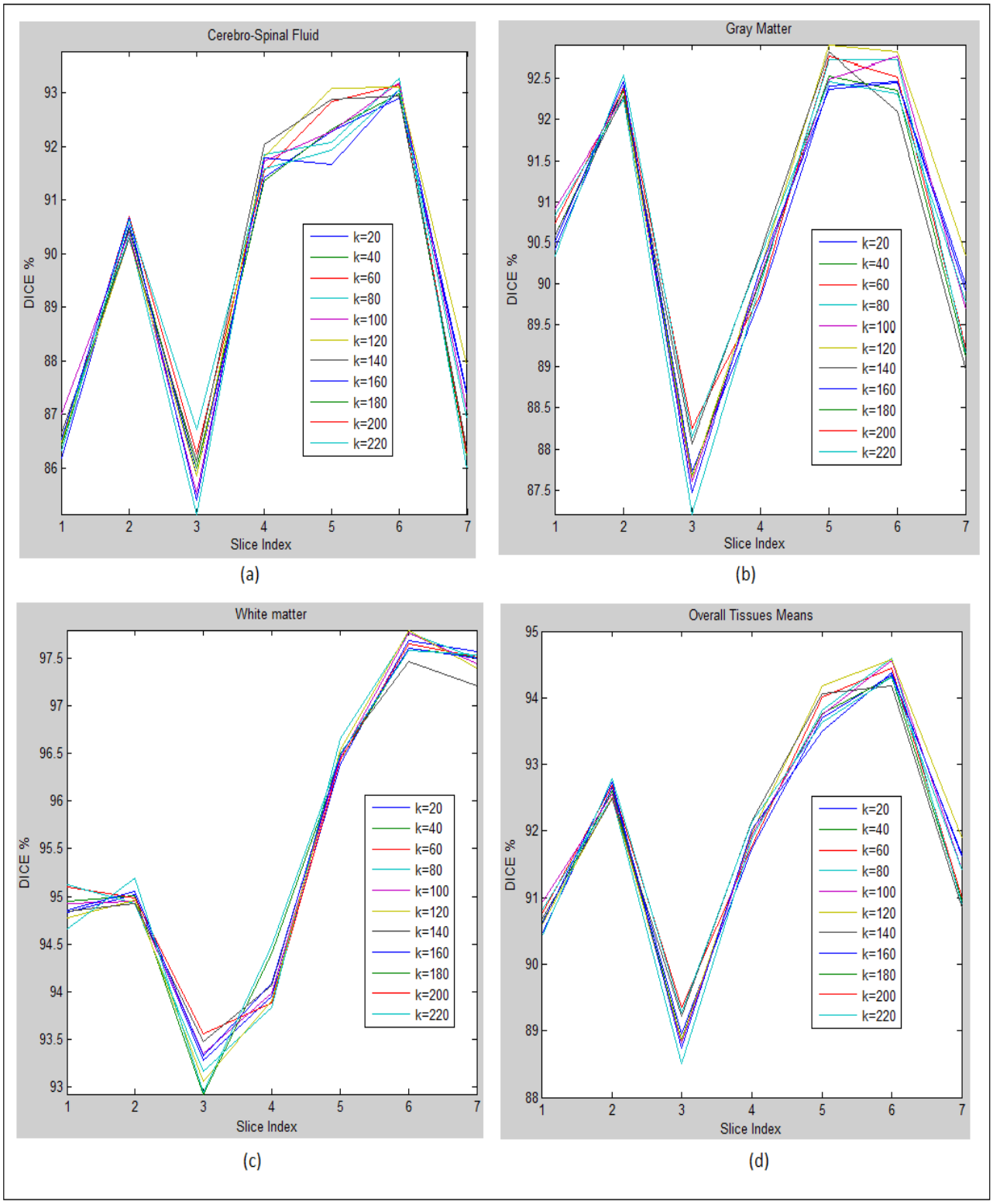


Fig. 6.3: (a) The Dice percentage of Cerebro Spinal Fluid for 7 images for 11 different k values (b) The Dice percentage of Gray Matter for 7 images for 11 different ***k*** values (c) The Dice percentage of White matter for 7 images for 11 different ***k*** values .As a conclusion we can figure out that the differences are small between the ***k*** values, but undoubtedly, the value k = 120 seems to produces the optimal results.

k values	k=20	k=40	k=60	k=80	k=100	k=120	k=140	k=160	k=180	k=200	k=220
Dice CSF %	89,51	89,4	89,61	89,32	89,6	89,79	89,61	89,5	89,6	89,59	89,6
Dice GM %	90,75	90,67	90,82	90,7	90,84	90,97	90,74	90,76	90,84	90,84	90,84
Dice WM %	95,53	95,55	95,59	95,57	95,55	95,49	95,5	95,56	95,56	95,56	95,56
Overall	91,93	91,87	92,01	91,86	92,00	92,08	91,95	91,94	92,00	92,00	92,00

Table 6.3: The Dice percentage (from all seven images presented in figure 6.1) for each Brain Tissue. Once again we observe that the differences are small between the various values of ***k***, with the value k = 120 to have an edge on better results.

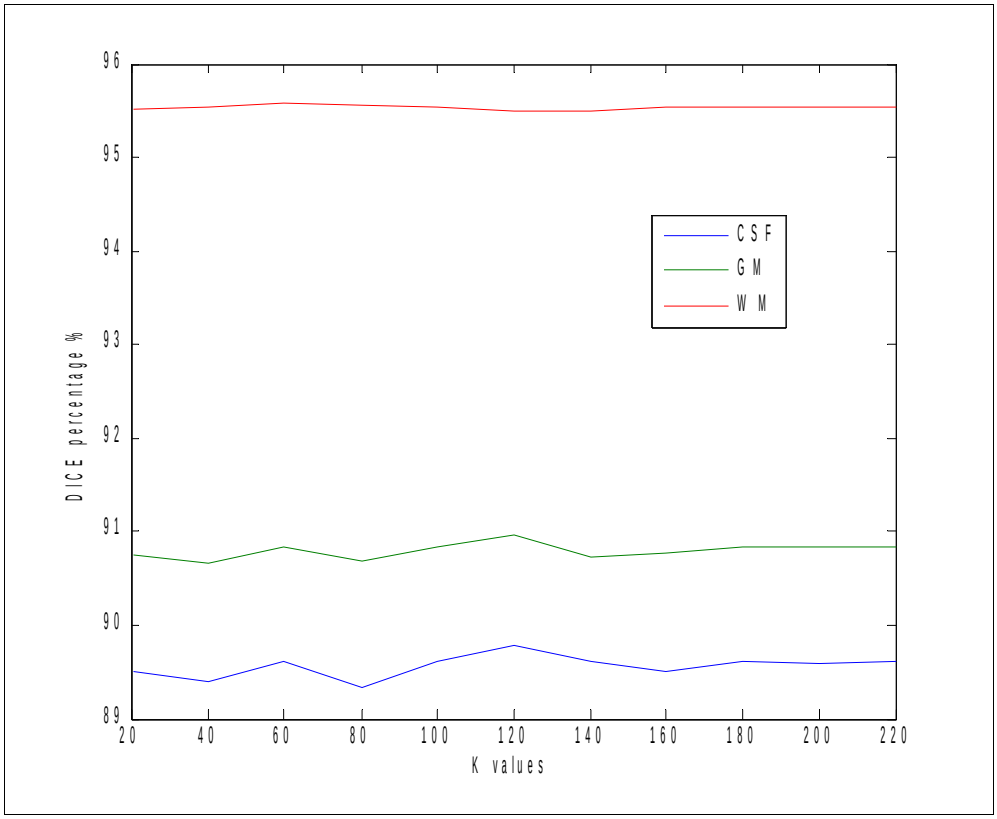


Fig. 6.4: The diagram made by table 6.5 showing the overall Dice percentage for each ***k*** value,confirming that value k=120 is the optimal selection.

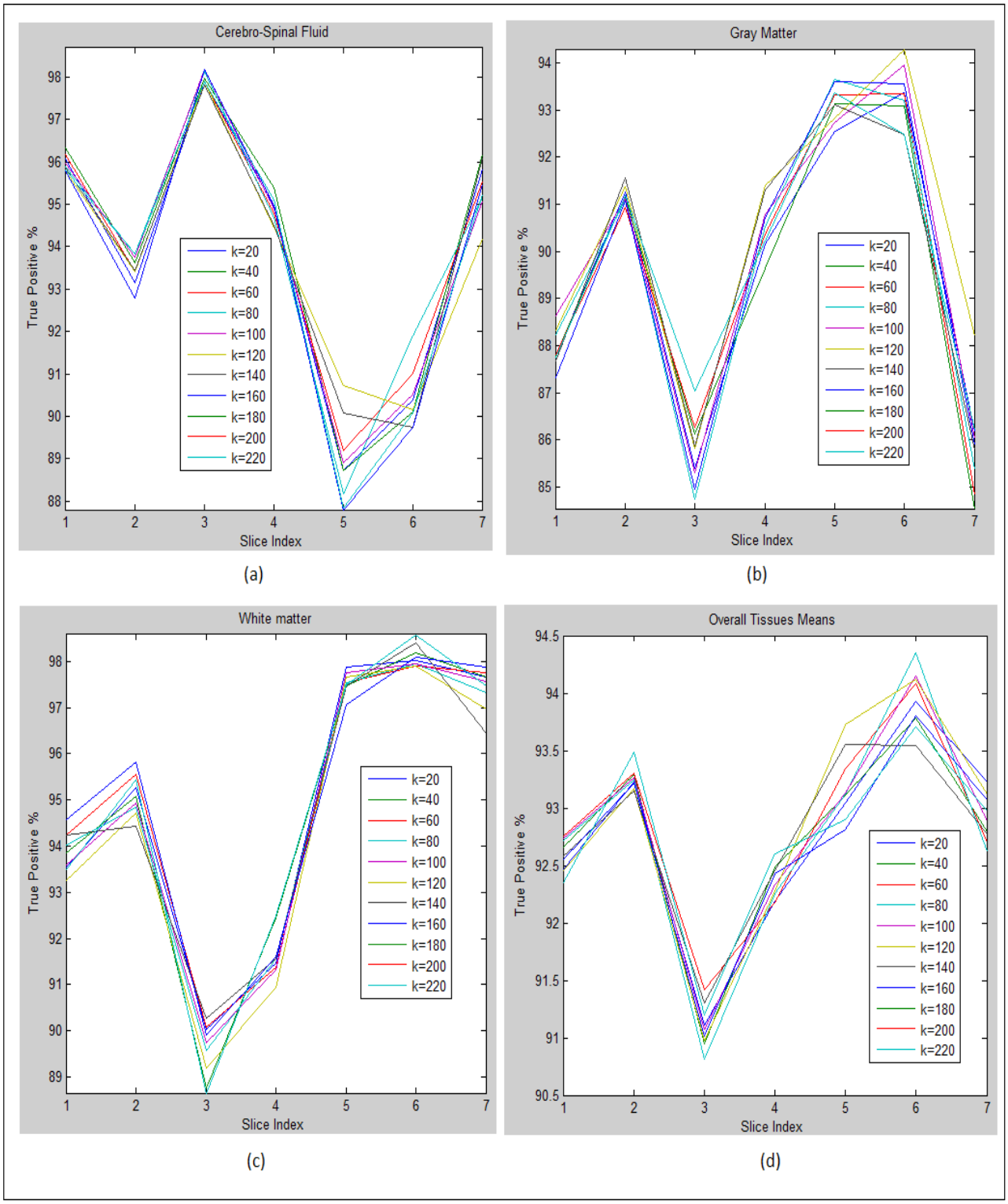


Fig. 6.5: Here there are presented the True Positive percentages, for the same 7 images as in the fig. 6.3, for the three brain tissues, CSF, GM, WM. As a conclusion, the value k=60 and k=120 produces the optimal results.

k values	k=20	k=40	k=60	k=80	k=100	k=120	k=140	k=160	k=180	k=200	k=220
TP CSF %	93,858	94,04	94	93,96	93,9	93,8	93,93	93,53	93,68	93,68	93,68
TP GM %	89,52	89,35	89,55	89,542	89,793	90,32	89,68	89,58	89,75	89,75	89,748
TP WM %	94,836	94,782	94,92	94,689	94,69	94,379	94,68	94,987	94,81	94,81	94,81
Overall	92,74	92,73	92,82	92,73	92,79	92,83	92,76	92,70	92,75	92,75	92,75

Table 6.4: The True Positive percentage (from all seven images presented in figure 6.5) for each Brain Tissue. In this table it is confirmed that the differences are small between the various values of ***k***, with the value k = 120 to have an edge on better results.

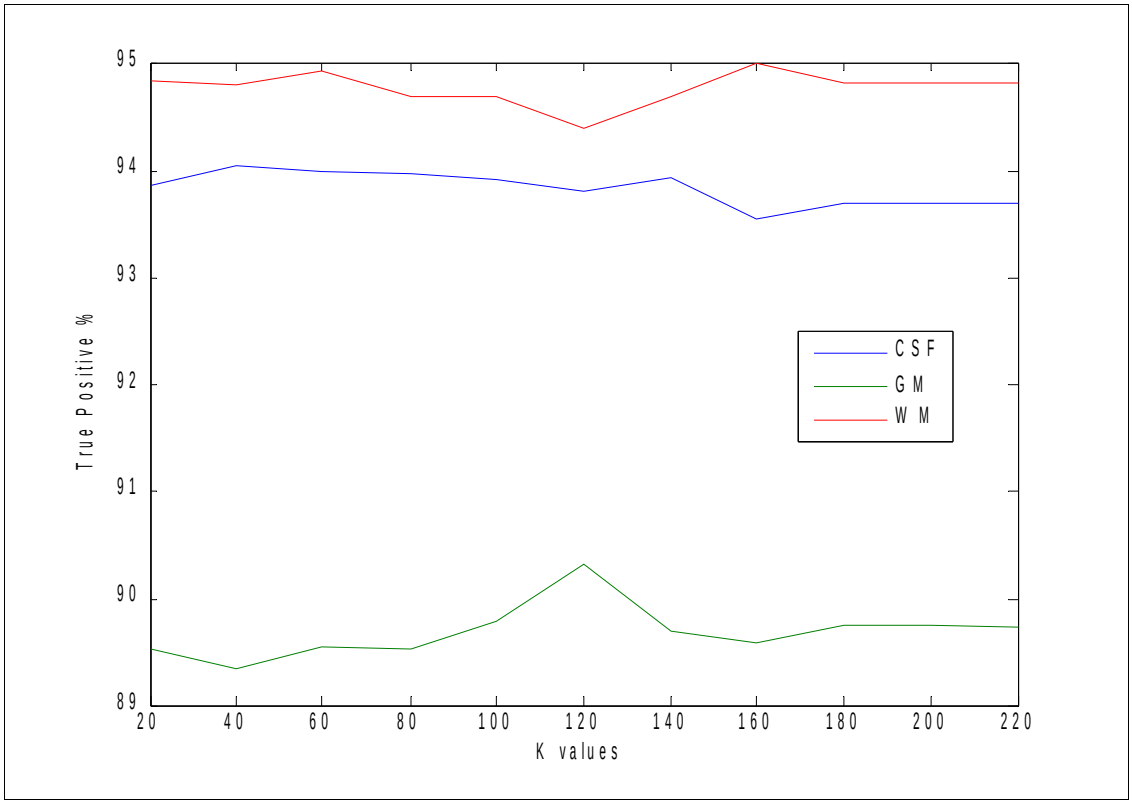


Fig. 6.6: The figure for True Positives presenting the results of table 6.4. It shows the overall True Positive percentage for the various values of ***k*** for all 7 images.

In this section we investigated the influence of *k* parameter. We already mentioned in Section 5.3 that the value of *k* should be chosen large enough to ensure that there is an increase in density within the support of most kernels. The results confirm this point, giving the optimal value for *k=120*.

6.4 Joint Effect of neighborhood size and *k* parameter

Having investigated a great variety of different values for both neighborhood size and k parameter, we further investigate, according to the experiments of Sections 6.2 and 6.1, the optimal combination for both neighborhood size and k parameter. We have selected some promising values, according to our experiments, for neighborhood size the values [5*5], [9*9], [12*12] and [14*14] and for k parameter the values 60,100,120,180,200, in order to find the optimal combination. In the next tables (6.5 and 6.6) the experimental results for Dice and True Positive percentage are presented.

DICE CSF %	K=60	K=100	K=120	k=180	K=200
[5*5]	89,560	89,439	89,282	89,424	89,716
[9*9]	89,486	89,251	89,474	89,174	89,718
[12*12]	89,485	89,603	89,792	89,595	89,595
[14*14]	89,497	89,703	89,843	89,492	89,446

DICE GM %	K=60	K=100	K=120	k=180	K=200
[5*5]	90,818	90,543	90,603	90,759	91,018
[9*9]	90,749	90,678	90,533	90,372	90,890
[12*12]	90,820	90,836	90,965	90,843	90,843
[14*14]	90,755	90,820	90,945	90,746	90,579

DICE WM %	K=60	K=100	K=120	k=180	K=200
[5*5]	95,622	95,458	95,552	95,578	95,661
[9*9]	95,590	95,586	95,369	95,497	95,631
[12*12]	95,587	95,553	95,492	95,555	95,555
[14*14]	95,517	95,496	95,562	95,540	95,355

Table 6.5: The Dice Similarity percentage for various neighborhoods in combination with various k values for the same images (total results) as in the previous 6.1-6.4 tables for CSF-GM-WM are presented. Here it is confirmed that we can choose both the neighborhood size and ***k*** value from a great variety of values and the influence in the results is not so significant. Some optimal choices are neighborhood [5*5] in combination with k=200, [12*12] with k=120 and [14*14] with k=120.

TP CSF %	K=60	K=100	K=120	k=180	K=200
[5*5]	93,830	94,548	93,616	93,531	93,100
[9*9]	93,491	93,511	93,727	93,891	93,779
[12*12]	94,001	93,896	93,801	93,680	93,680
[14*14]	93,338	94,162	93,955	93,468	93,502

TP GM %	K=60	K=100	K=120	k=180	K=200
[5*5]	89,585	88,995	89,277	89,556	90,105
[9*9]	89,412	89,700	89,217	88,907	89,733
[12*12]	89,554	89,793	90,323	89,747	89,747
[14*14]	89,647	89,702	89,880	89,641	89,661

TP WM %	K=60	K=100	K=120	k=180	K=200
[5*5]	94,955	94,616	94,908	94,883	95,104
[9*9]	95,121	94,699	94,688	94,829	94,945
[12*12]	94,919	94,694	95,179	94,809	94,809
[14*14]	94,870	94,599	95,103	94,859	94,464

Table 6.6: The True Positive percentage for various neighborhoods in combination with various k values for the same images (total results) as in the tables 6.1-6.5. Here we again conclude that we can choose both the neighborhood size and ***k*** value from a great variety of values and the influence in the results is not so significant. Some optimal choices are neighborhood [5*5] in combination with k=200, [12*12] with k=120 and [14*14] with k=120.

As a conclusion from this joint investigation of neighborhood size and k parameter experiment, we are now sure that both the neighborhood size and *k* parameter value, can be chosen from a great variety of values without any significant influence in the results. Some options seem to be the optimal ones, but they are influenced by the kind of modality (T1,T2,Pd) by the total amount of brain tissue pixels of the image, and of course by the characteristics of each image. In this work, we chose the combination of a neighborhood of [12*12] and k=120 as they produce better results in various datasets and the algorithm is executed in an acceptable timeframe of time. (for a dataset of 117 images (240*240) it needs about 17 minutes in matlab).

6.5 The effect of Kernel Usage

For the rest of the experiments we are going to demonstrate only the Dice percentage of the three brain tissues, as it is considered to be the most reliable metric. In this section, we will investigate the influence of the Kernel usage, with two Kernels testing, Epanechnikov and Normal. We remind that in order to estimate the pdf we use the equation:

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}k\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)$$

with the Epanechnikov Kernel:

$$K_E(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|x\|^2) & \|x\|\leqslant 1 \\ 0 & otherwise \end{cases} \qquad \textbf{(6.3)}$$

whereas the Gaussian Kernel is:

$$K_N(x) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}\|x\|^2\right) \qquad \textbf{(6.4)}$$

where in both cases *d* stands for the dimensionality, in our case 3 (spatial-intensity).

In Table 6.7 we present the Dice percentage for 7 brain slices for Adaptive-Mean Shift Algorithm using the Epanechnikov and Gaussian Kernel.

	Epanechnikov	Gaussian
Dice CSF %	88,13	89,792
Dice GM %	89,47	90,965
Dice WM %	94,02	95,592
Overall Running Time (For 7 images)	71,317 sec	109,67 sec

Table 6.7: The mean Dice percentage for 7 images for the three Brain Tissues using two Kernels, Epanechnikov and Gaussian. In this table we must emphasize that the Gaussian Kernel provides better results than the Epanechnikov, but it is more time consuming. So there is a trade off between an algorithm with less good results but with very good running time and an algorithm with better results though more time requiring.

In summary, we tested the mean-shift algorithm using two kernels: The Epanechnikov and the Gaussian Kernel. Undoubtedly, Gaussian Kernel produces results with better quality, but needs more time to be executed (about 16.5 minutes in a pc with processor 2.26 Ghz for a dataset of 117 images of size 240*240) whereas Epanechnikov Kernel produces results with less quality (but still acceptable segmentation) but needs less time to be executed (about 11 minutes for the same dataset). As will be further discussed in the Conclusions and Recommendations Section [7], a C/C++ implementation of the proposed Gaussian-Mean-Shift algorithm will definitely reduce significantly the running time, in half.

6.6 The Effect of Additive Noise

In this section we are going to investigate the effect of additive noise in the overall perfomance of the Gaussian Adaptive-Mean-Shift algorithm. Firstly, we will demonstrate the Dice percentage results for additive noise for the proposed algorithm without applying any filter, in order to estimate the concequences of additive noise in the mean-shift procedure, then the Dice results for the proposed algorithm with applying a Gaussian filter and finally the Dice results for the proposed algorithm with applying a Median filter. Tables 6.8-6.9 present these results:

2 % Additive Noise	Without any filter	Gaussian filter	Median filter
Dice CSF %	84,886	84,946	85,406
Dice GM %	86,911	86,870	87,219
Dice WM %	88,667	88,593	89,187

Table 6.8: Dice percentage for 2% additive noise in the proposed algorithm without implying any filter, with Gaussian filter and with median filter.

6 % Additive Noise	Without any filter	Gaussian filter	Median filter
Dice CSF %	83,988	83,958	84,273
Dice GM %	86,245	86,180	86,612
Dice WM %	88,801	88,690	88,903

Table 6.9: Dice percentage for 6% additive noise in the proposed algorithm without implying any filter, with Gaussian filter and with median filter.

10 % Additive Noise	Without any filter	Gaussian filter	Median filter
Dice CSF %	83,730	84,034	84,242
Dice GM %	86,208	85,967	86,553
Dice WM %	88,622	88,426	88,750

Table 6.10: Dice percentage for 10% additive noise in the proposed algorithm without implying any filter, with Gaussian filter and with median filter.

From tables 6.8,6.9 and 6.10 we come to the conclusion that the mean-shift proposed algorithm is robust at all levels of noise, low, medium and high. Another conclusion taken from the tables presented is that the application of a median filter improves the results. This presumption is confirmed even in no noise occasions. In figure 6.7 we present the Dice results for various additive noise levels, with a median filter used in the proposed algorithm.

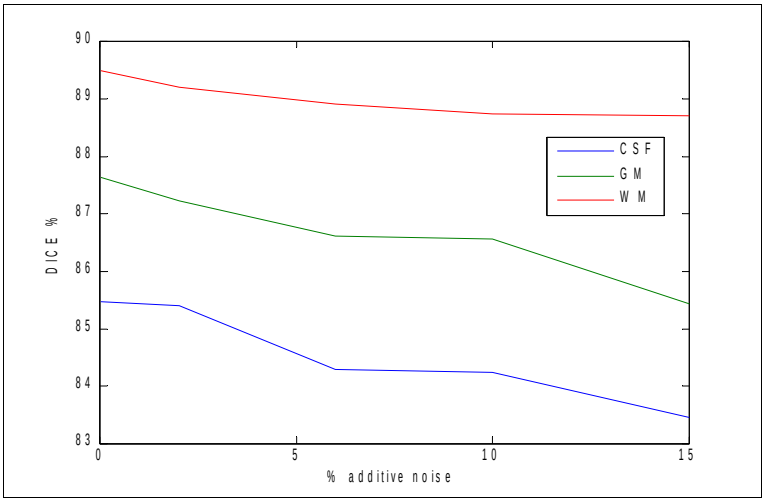


Fig 6.7: The effect of additive noise in the efficiency of the proposed mean-shift algorithm. We can observe that the proposed algorithm is adequate for all levels of noise.

6.7 The effect of Mahalanobis Pruning Modes Step usage

Observing the overall proposed algorithm and each stadium of it, one may wonder why we have to prune the modes, produced by the mean-shift procedure, and not use directly the fuzzy k-means step in order to produce the final segmented results. For this case, in this section we investigate the necessity of the Mahalanobis Pruning Modes step. After having experimented with several images from the dataset where the Golden Truth was available, as shown in table 6.11, we came to the conclusion that for small to medium size pictures (for example 240*240), and more specifically, for small number of remaining modes (50-400) this step **does not** improve significantly the results. Instead, it requires increased running time in order to make all the appropriate comparisons and delays the whole procedure of the algorithm. Usually, the algorithm, using the Mahalanobis Pruning step, produces almost the same results as using directly the fuzzy k-means algorithm in the case where the remaining modes are between 150 and 200. In less cases, it produces slightly better results. On the other hand, when the remaining modes after the mean-shift procedure are more than 400, the Mahalanobis Pruning Modes step seem to produce better results (figure 6.8), because using the fuzzy k-means stadium directly segments some pixels to wrong brain tissue cluster. However, if we over-prune the remaining modes, that is to say prune the remaining modes with the Mahalanobis Pruning Modes step until they become less than 150, we end up to misclassification.

Image 1 Remaining Modes:274	Without Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step
Dice CSF %	91,040	91,040	90,890	86,523
Dice GM %	90,121	90,120	89,724	85,512
Dice WM %	93,412	93,412	92,836	89,310
Remaining modes after	274	194	138	37
Running time	18,43 sec.	61,796 sec.	78,93 sec.	98,95 sec.

(a)

Image 2 Remaining Modes:263	Without Mahal.Prun. Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step
Dice CSF %	91,815	91,815	91,427	86,923
Dice GM %	90,200	90,200	89,327	85,684
Dice WM %	93,907	93,907	93,273	90,307
Remaining modes after	263	200	142	42
Running time	17,9 sec.	41,718 sec.	50,66 sec.	88,72 sec

(b)

All Dataset (117 images of size 240*240)	Without Mahal.Prun. Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step	With Mahal.Prun. Modes Step
Dice CSF %	85,455	85,467	84,518	80,204
Dice GM %	87,619	87,631	86,268	82,414
Dice WM %	89,490	89,490	88,658	85,755
Remaining modes after	No pruning 200-400 remaining modes	< 200	< 150	<50
Running time	16,54 min.	34 min.	40,4 min.	56,42 min.

(c)

Table 6.11: In Table 6.11 a and b, we demonstrate the Dice percentage results for two images, showing that whenever we use without this Mahalanobis Pruning Modes step or not, the segmented results are the same for a number of remaining modes less than 200. With less than 150 modes remaining, we over-pruned our modes leading to misclassification. In table 6.11 c in order to be sure for our conclusions, we have tested for all the dataset the use of Mahalanobis Pruning Modes step.

Pruning the modes until they become less than 200, may improve the results, but not significantly. Comparing the running times of the two different implementations, we definitely should choose the no-mahalanobis pruning modes implementation, as it is executed in much less time.(16,54 minutes while in the implementation with pruning modes until they become less than 200 needs 34 minutes). Finally, we observe that over-pruning of the modes (< 150) end up to lower Dice percentages in all tissue types and in combination with the high running times makes these implementations not acceptable.

From table 6.11 we came to the conclusion, that when the remaining modes, after the mean-shift procedure, are less than 400, then the mahalanobis pruning stadium produces the same or at least not significantly better results, than fuzzy-k-means by itself only, and in combination with the increased running time that it requires, turns to be not acceptable. But what happens in bigger datasets with remaining modes more than 400? In that case, as we can observe in figure 6.8 , the mahalanobis pruning modes step is useful as it gives the opportunity to the fuzzy-k-means step to successfully segment the remaining modes whereas if we solely use the fuzzy- k-means step, we increase the possibility of misclassifying the remaining modes.

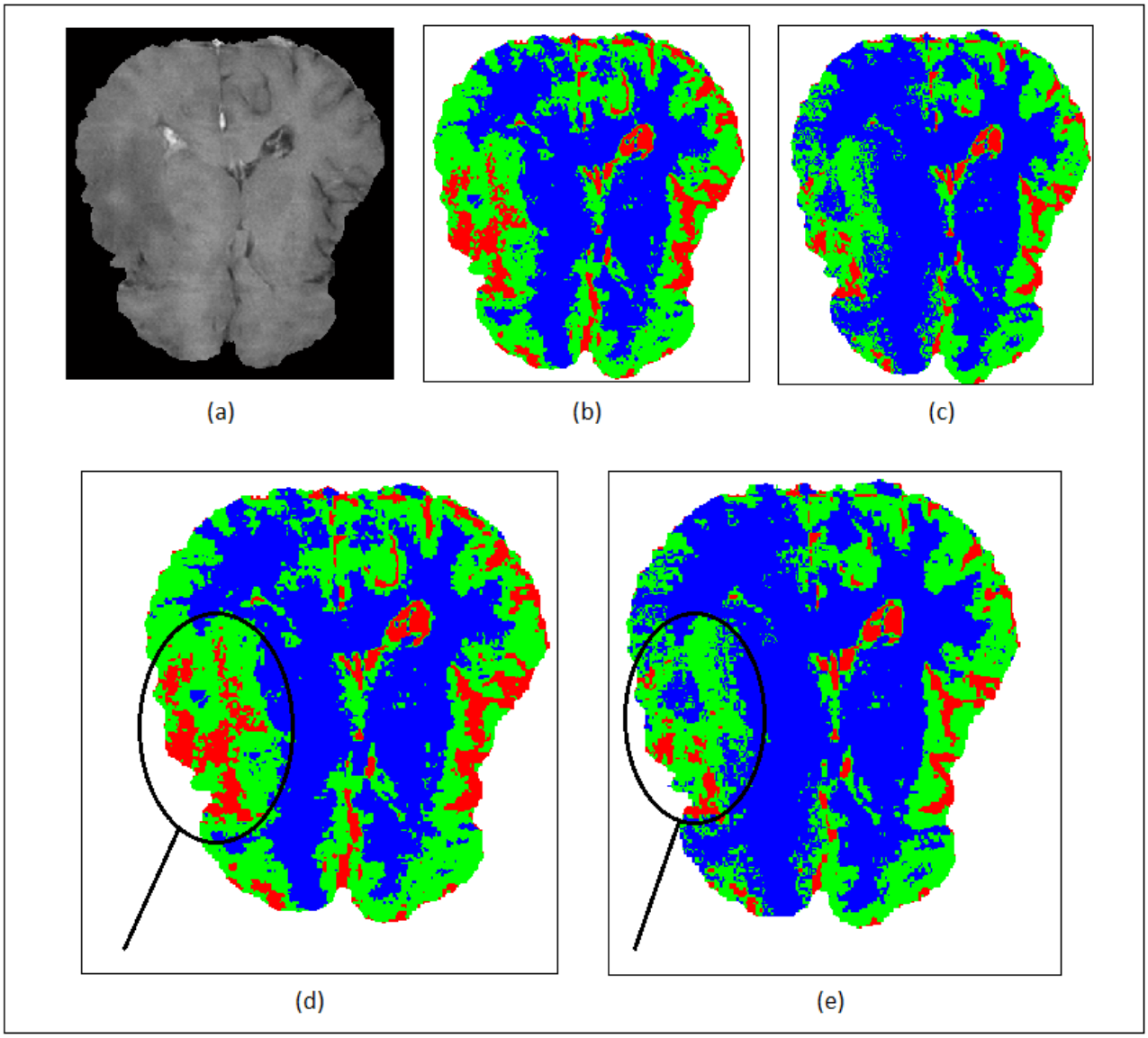


Fig. 6.8: (a) The initial image (b) The segmented image without the mahalanobis pruning modes step and (c) the segmented image with the mahalanobis pruning modes step, with 283 modes remaining. In both images the number of the remaining modes after the mean-shift procedure was 812.In order to make it more clear in pictures (d) and (e) we have marked the central controversial segmented area. From the initial image (a) we can observe that the more accurate segmentation image is (e) as in (d) some pixels from the Gray matter and White matter category have been mistakenly segmented in the CSF category.

After tested the segmented results in various images, with the most indicative example being the image in fig 6.8, we came to the conclusion that if the remaining modes, after the mean-shift procedure, is more than 400, then to increase the possibilities of a better segmentation result, is to use the mahalanobis pruning stadium until the remaining modes reach a number of less than 400, and afterwards continue the segmentation with the final step, the fuzzy k-means.

6.8 Comparison of K-means - Fuzzy K-means

In this section we are going to compare the use of a simple K-means implementation of the mean-shift algorithm with the use of fuzzy-K-means, which gives us the opportunity to provide a possibility map, in which there will be contained the probability for each pixel to belong to each brain tissue category. As mentioned earlier the basic difference between the two implementations is that in fuzzy k-means , for the calculation of the centers of each cluster, we are taking into account the probability of each pixel to belong to this cluster [Sections 4.4, 4.5]. The problem of initialization of the membership matrix is solved in the following way: We classify all the pixels of the remaining modes according to K-means algorithm for the first and last time. To be more specific we classify all pixels to clusters, by finding the minimum distance from the initial centers of clusters (which they are defined by the histogram of the image) and then we calculate the new centers of the clusters according to the mean values of the members of each cluster. Then we calculate the initial membership matrix according to the new mean values, calculated with the K-means procedure with the following equation (Section 4.5, Eq. 4.56):

$$u_{ik} = \frac{[1/(|x_k - v_i|)^2]}{\sum_{j=1}^{c} [1/(|x_k - v_j|)^2]^{1/(m-1)}} \quad , i = 1,\ldots, c \text{ and } k = 1,\ldots, n$$

where c is the number of the clusters and $k = 1,\ldots, n$ the amount of pixels of each remaining mode.

We prefer to initialize the membership function in this way and not in a random way, in order to take advantage of all available information and thus optimize the results. In table 6.12 we compare these two different implementations using the Dice Similarity metric. In both implementations we have used Euclidean distance, in order to calculate for each pixel that is the shortest distance from the centers of clusters and decide where this pixel should be classified.

Dice %	K-means	Fuzzy K-means
CSF	84,422%	85,455%
GM	86,220%	87,619%
WM	88,661%	89,490%
Running Time:	16,44 min	16,54 min.

Table 6.12: Dice percentage for k-means and fuzzy k-means for a dataset of 117 images, size 240*240

From table 6.12 we came to the conclusion that taking into account the possibility of each pixel to belong in each brain tissue we improve the results, in comparison with the simple k-means version of the algorithm, plus the fuzzy k-means implementation does not require more running time. Using the fuzzy k-means algorithm we provide a qualitative segmentation map and a membership matrix of possibilities for each brain tissue type.

6.9 Optimal Parameters Selection

After having investigated the influence of various parameters we question, what parameters values are optimal. Of course, there isn't only one optimal solution, as the final segmented results depends on the special characteristics of the image, the modality we use. From the experiments presented in this section, we set the *neighborhood size* to be [12*12], the *k* value 120, *to use a median filter* in order to enhance the procedure not only in noise cases and in general, a mahalanobis pruning modes step if the remaining modes after the mean-shift procedure are more than *400.* Finally t*he fuzzy-k-means algorithm* instead of a simple k-means was selected. The dataset used contained 117 brain slices of size [240*240] and in tables 6.13 and 6.14 we present the metrics that there were defined in the introduction of this section (6.1) using the proposed mean-shift algorithm, which needs about 16.5 minutes in a 2.26 Ghz processor computer:

Metrics:	True Positive %	False Positive Rate	False Negative Rate	Sensitivity	Specificity	Likelihood-ratio Positive
CSF	91,284	0,0077	0,0872	0,913	0,992	224,694
GM	84,411	0,0070	0,1559	0,844	0,993	172,760
WM	93,231	0,0061	0,0677	0,932	0,994	348,623

Table 6.13: True Positive, False positive rate, False Negative Rate, Sensitivity, Specificity, Likelihood-ratio Positive metrics for the whole dataset of 117 images of size 240*240.

Metrics:	Likelihood-ratio Negative	Jaccard	Dice Similarity %	Tanimoto	Segmentation Accuracy
CSF	0,088	0,0377	85,455%	0,979	0,990
GM	0,157	0,0908	87,619%	0,962	0,981
WM	0,068	0,0817	89,490%	0,982	0,991

Table 6.14: Likelihood-ratio Negative, Jaccard, Dice Similarity %, Tanimoto, Segmentation Accuracy metrics for the whole dataset of 117 images of size 240*240.

6.10 Efficiency of algorithm in other modalities

In all previous experiments and result analysis, we used T1-weighted images. It was available though, for the same dataset with the golden segmented method available, two different modalities, T2 and Pd-weighted. We present in table 6.15 the True Positive and Dice Metrics for the other two modalities:

	T1	T2	Pd-weighted
TP CSF %	91,284	69,181	73,137
TP GM %	84,411	58,015	60,713
TP WM %	93,231	53,600	57,720
Dice CSF %	85,455	93,910	77,982
Dice GM %	87,619	63,496	70,334
Dice WM %	89,490	52,134	61,829
Running Time	16,54 min.	16,77 min.	16,758 min.

Table 6.15: TP and Dice percentage for T1, T2, Pd-weighted modalities. We also include the running times of the proposing algorithm in all these cases.

Consequently, from table 6.15 we point out that the proposed algorithm is better in T1 modality, then in T2 and in Pd-weighted. We must emphasize though that the mostly used modalities in brain segmentation domain, are T1 and T2.

6.11 Comparison with other segmentation methods

In this section we are going to compare the proposed adaptive-Mean-Shift algorithm with three other algorithms, that represent different approaches in pattern recognition domain: Classic K-means algorithm, Markov Random Fields and Gaussian Mixture Models.

The theory of classic K-means algorithm has been discussed in detail in section 4.4.

6.11.1 Markov Random Field

Markov Random Field (MRF), [81], is a stochastic process, used as *a priori* model to incorporate spatial correlations into a segmentation process. If we define clustering as pixel labeling and use the term *sites* instead of pixels, in order to be consistent with the theory [82], we will denote a set of image lattice sites $S = \{1,\ldots,n\}$, that represent the primitive objects to be labeled. In the two-dimensional image lattice S, the pixel values $\mathbf{y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_n\}$ are a realization of the random variables $\mathbf{x} = \{x_1,\ldots,x_n\}$. In general, the number of observation vectors does not need to be of the same size as the set of sites, however, in this application, the number of observations is equal to the number of sites.

An optimal labeling of the MRF satisfies the *maximum a posteriori* probability criterion (MAP-MRF), which requires the maximization of the posterior probability $P(x|y)$ of the labeling that is assumed to follow the Gibbs distribution [33], [82]:

$$P(x|y)=Z^{-1}e^{-(U(x|y)/T)} \qquad \textbf{(6.5)}$$

where Z is a normalizing constant, $U(\mathbf{x}|\mathbf{y})$ is the posterior energy and T is called system temperature and is assigned to the value of 1 in problems related to image segmentation.

The simplified Bayes rule $P(x|y)=P(x)\,p(y|x)$ is used to incorporate the Gibbs probability density function into the decision function:

$$P(x)\,p(y|x)=e^{(-U(x))}e^{(-U(y|x))}=e^{-(U(x)+U(y|x))} \qquad \textbf{(6.6)}$$

Hence (6.5) is now expressed as the sum of the prior and the likelihood energy tern:

$$U(x|y) = U(x)+U(y|x) \qquad \textbf{(6.7)}$$

The Gibbs distribution model defines the labeling of a site in dependence to the labeling of all the other sites of the lattice S . However, the MRF model itself is a conditional probability model, where the probability of a site label depends on the site labels within its neighborhood.

The equivalence of the Gibbs and the MRF models has is stated in the Hammersley-Clifford theorem. The incorporation of localized characteristics into the model reduces its complexity. Neighboring sites are grouped into *cliques*, whose order (that indicates the number of included sites) is variable, according to the desired complexity. If we define unary cliques, including one site $\{i\}$ and binary ones, including two neighboring sites $\{i,i'\}$, the energy can be calculated as the sum of the local potentials defined on the cliques. If $V(x_i)$ and $V(x_i,x_{i'})$ are the potentials of unary and binary clique respectively, and $V(\mathbf{y}_i\,|\,x_i)$ are the likelihood potentials, then the prior energy, defined as the sum of all clique potentials, can be written:

$$U(x)=\sum_{i\exists S} V(x_i)+\sum_{i\exists S}\sum_{i'\exists N_i} V(x_i,x_{i'}) \qquad \textbf{(6.8)}$$

where N_i is the set of all sites neighboring i , excluding i .

At the same time, the likelihood energy is the sum of all likelihood potentials under the assumption that the observations, conditioned by the labels x_i , are mutually independent:

$$U(y|x)=\sum_{i\exists S} V(y_i|x_i) \qquad \textbf{(6.9)}$$

The MAP-MRF solution, defined in (6.5), is equivalent to the minimization of the energy, as defined in (6.7):

$$\hat{x}=\arg\min_{x} U(x|y) \qquad \textbf{(6.10)}$$

In order to find the global minimum of that energy, in this work, the deterministic iterated-conditional mode algorithm (ICM) [81] was used.

6.11.2 Gaussian Mixture Models (GMM)

In image clustering problems, the observed image can be considered as a mixture of Gaussian distributions [83], with M >1 components in $\Re^n$ for n ≥1:

$$p(x|\theta)=\sum_{m=1}^{M} a_m p(x|\theta_m), \forall x \exists \Re^n \qquad \textbf{(6.11)}$$

where $\alpha_1,\ldots,\alpha_M$ are the mixing proportions, θ_m is the set of parameters defining the *m*th component, and $\theta = \{\theta_1,\ldots,\theta_M,\alpha_1,\ldots,\alpha_M\}$ is the complete set of parameters needed to specify the mixture. Being probabilities, the α_m must satisfy:

$$a_m \geq 0, m=1,..,M, \sum_{m=1}^{M} a_m = 1 \qquad \textbf{(6.12)}$$

For the Gaussian mixtures, each component density is a normal probability distribution with parameters $\theta_m = (\mu_m, \Sigma_m)$, where μ_m is the mean of each component and Σ_m is the corresponding covariance. This way, (6.11) can be rewritten as:

$$p(x|\theta)=\sum_{m=1}^{M} a_m N(x|\mu_m, \Sigma_m) \qquad \textbf{(6.13)}$$

where $N(x|\mu_m,\Sigma_m)$ is a Gaussian distribution.

The most common approach for estimating the parameters for a Gaussian mixture model, given a dataset, is to assume that the observed data are only part of the underlying complete data and use the Expectation-Maximization (EM) algorithm for the maximum-likelihood estimation [84]. Especially, for the MRI tissue classification problem, the observed data are the pixel intensities and the missing data are the classification of the images.

The usual EM algorithm is an iterative technique that consists of an E-step and an M-step. Suppose that $\Theta^{(t)}$ is the estimation of the parameters Θ after the *t*th iteration. Then, at the $(t+1)$th iteration, the E-step calculates the expected complete data log-likelihood function:

$$Q(\Theta, \Theta^t)=\sum_{k=1}^{K}\sum_{m=1}^{M} (\log a_m p(x_k|\theta_m)) P(m|x_k, \Theta^t) \qquad \textbf{(6.14)}$$

where $P(m|x_k, \Theta^t)$ is a posterior probability and is computed as:

$$P(m|x_k, \Theta^t)=\frac{a_m^{(t)} p(x_k|\theta^t)}{\sum_{l=1}^{M} a_l^{(t)} p(x_k|\theta^{(t)})} \qquad \textbf{(6.15)}$$

The M-step calculates the *(t+1)*th estimation $\Theta^{(t+1)}$ by maximizing the log-likelihood function:

$$a_m^{(t+1)} = \frac{1}{K}\sum_{k=1}^{K} P(m|x_k, \Theta^{(t)}) \qquad \textbf{(6.16)}$$

$$\mu_m^{(t+1)} = \frac{\sum_{k=1}^{K} x_k P(m|x_k, \Theta^t)}{\sum_{k=1}^{K} P(m|x_k, \Theta^t)} \qquad \textbf{(6.17)}$$

$$\sum_{m}^{2^{(t+1)}} . = \frac{\sum_{k=1}^{K} P(m|x_k, \Theta^{(t)})(x_k - \mu_m^{(t+1)})(x_k - \mu_m^{(t+1)})^T}{\sum_{k=1}^{K} P(m|x_k, \Theta^{(t)})} \qquad \textbf{(6.18)}$$

The above equations (6.16-6.18) can be solved numerically by alternating iteratively between E-step and M-step. The algorithm fills in the missing data during E-step and then finds the parameters that maximize the log-likelihood for the complete data in the M-step. After the parameters of the GMM have been calculated, clusters are assigned to each observed data by selecting the component with the largest posterior probability weighted by the component probability.

6.11.3 Comparison

In tables 6.16, 6.17,6.18, 6.19,6.20,6.21 and 6.22 the presented methods are compared.

Dice %	Classic K-means	Markov Random Fields	Gaussian Mixture Models	Adaptive-Mean-Shift
CSF	79,513	61,525	68,308	85,455
GM	82,938	57,918	73,618	87,619
WM	86,255	71,027	71,918	89,409

Table 6.16: Dice percentage for the four comparing techniques: Classic K-means, Markov Random Fields, Gaussian Mixture Models, Adaptive-Mean-Shift.

Sensitivity	Classic K-means	Markov Random Fields	Gaussian Mixture Models	Adaptive Mean-Shift
CSF	87,317	47,348	62,891	91,284
GM	77,709	51,629	82,939	84,411
WM	92,243	95,504	65,235	93,231

Table 6.17: The Sensitivity for the four comparing techniques: Classic K-means, Markov Random Fields, Gaussian Mixture Models, Adaptive-Mean-Shift.

Tanimoto Coefficient	Classic K-means	Markov Random Fields	Gaussian Mixture Models	Adaptive Mean-Shift
CSF	96,883	95,624	87,324	97,944
GM	94,767	87,895	84,072	96,235
WM	97,102	91,369	86,525	98,170

Table 6.18: The Tanimoto Coefficient for the four comparing techniques: Classic K-means, Markov Random Fields, Gaussian Mixture Models, Adaptive-Mean-Shift.

K-means	CSF	GM	WM
True Positive %	87,317	77,709	92,243
False Positive Rate	0,0116	0,0087	0,0110
False Negative Rate	0,1268	0,2229	0,0776
Sensitivity	0,8732	0,7771	0,9224
Specificity	0,9884	0,9913	0,9890
Likelihood Ratio Positive	0,129	0,226	0,080
Likelihood Ratio Negative	75,276	89,322	83,854
Jaccard	0,036	0,083	0,797
Dice Similarity %	79,513	82,938	86,255
Tanimoto Coefficient	0,969	0,948	0,971
Segmentation Accuracy	0,984	0,973	0,985

Table 6.19: All metrics calculated for the K-means implementation.

Mean-Shift	CSF	GM	WM
True Positive %	91,284	84,411	91,284
False Positive Rate	0,0077	0,0070	0,0610
False Negative Rate	0,0872	0,1559	0,0677
Sensitivity	0,9128	0,8441	0,9323
Specificity	0,9923	0,9930	0,9939
Likelihood Ratio Positive	224,69	172,76	348,62
Likelihood Ratio Negative	0,088	0,157	0,068
Jaccard	0,038	0,091	0,082
Dice Similarity %	85,455	87,619	89,409

Tanimoto Coefficient	0,979	0,962	0,982
Segmentation Accuracy	0,990	0,981	0,907

Table 6.20: All metrics calculated for the proposed Mean-Shift algorithm.

MRF	CSF	GM	WM
True Positive %	47,350	51,629	95,504
False Positive Rate	0,0013	0,0244	0,0474
False Negative Rate	0,5265	0,4837	0,0450
Sensitivity	0,4735	0,5163	0,9550
Specificity	0,999	0,976	0,953
Likelihood Ratio Positive	364,23	21,16	20,15
Likelihood Ratio Negative	0,528	0,496	0,050
Jaccard	0,018	0,056	0,083
Dice Similarity %	61,525	57,918	71,027
Tanimoto Coefficient	0,956	0,879	0,914
Segmentation Accuracy	0,978	0,935	0,954

Table 6.21: All metrics calculated for the MRF implementation.

GMM	CSF	GM	WM
True Positive %	62,89	82,94	65,24
False Positive Rate	0,0027	0,0300	0,0706
False Negative Rate	0,2711	0,0706	0,2477
Sensitivity	0,6289	0,8294	0,6523
Specificity	0,8973	0,8700	0,8984
Likelihood Ratio Positive	6,120	6,380	6,421
Likelihood Ratio Negative	0,272	0,072	0,248
Jaccard	0,025	0,091	0,058
Dice Similarity %	68,308	73,618	71,918
Tanimoto Coefficient	0,873	0,841	0,865
Segmentation Accuracy	0,886	0,869	0,882

Table 6.22: All metrics calculated for the GMM implementation.

Consequently, from the four techniques that we compared using the same dataset, Adaptive-Mean-Shift yields the best results, in all metrics (most important of which are Dice Similarity, Sensitivity and True Positive), proving that , undoubtedly it is an extremely useful tool in segmentation field, with great prospects. The disadvantage of the proposed algorithm, is that it requires more running time that the other techniques checked, but as it is mentioned in Section 7 “Conclusions and Recommendations” a complete C or C++ implementation of the proposed algorithm (this algorithm has been programmed in matlab only) would certainly reduce the required running time , by more than the half.

7. CONCLUSIONS AND RECOMMENDATIONS

The overall goal of this thesis is to propose a reliable brain tissues segmentation algorithm and furthermore, by using prior knowledge about the nature of the T1 and T2 MR images, to detect possible cancer pixels. The main conclusions and recommendations drawn from this work are summarized next.

7.1 Conclusions

Basic brain anatomy concepts were presented. Human brain, the center of the human nervous system, consists of three main brain tissue types: Cerebro-Spinal Fluid (CSF), bodily fluid that provides a basic mechanical and immunological protection to the brain inside the skull, Gray Matter (GM), a major component of the central nervous system (CNS) which function is to route sensory or motor stimulus to interneurons of the CNS, and White Matter (WM) which connects various GM areas of the brain to each other, and carry nerve impulses between neurons. The function of Magnetic resonance imaging (MRI) and basic modalities which are commonly used such as T1, T2 and Proton Density (Pd) were also discussed. Finally, the major issue of brain tumor was discussed, defining his characteristics, his different types and lastly diagnostic methods that are commonly used.

Basic common techniques that are used in image processing like the histogram of an image and probability density function (pdf) were detailed discussed. Furthermore, the theory behind estimating an unknown pdf was presented. Moreover, the issue of brain tissues segmentation was analyzed, presenting plenty techniques that have tried to provide a reliable and reproducible algorithm. These techniques can in general be separated according to whether they use training samples in order to "train" their classifier or not (following supervised or unsupervised approach respectively) and whether the forms of the density functions, which are formed according to the way that the data are being modeled, are known or at least assumed according to some common parametric forms or not (following parametric or non parametric approach respectively). Finally, basic tumor modeling techniques were discussed, presenting various methods that were used in the past trying to provide an accurate method to detect tumor pixels.

A common technique, widely used in pattern recognition for estimating an unknown pdf, Parzen Windows, was detailed discussed. Based on Parzen Windows, Mean-Shift algorithm guarantees detection of the local maximum of the unknown pdf, **without estimating it.** In this way it is earned valuable time as in a segmentation algorithm which uses both spatial and intensity information, the curse of dimensionality, as it is named the problem of slow data processing in multi-dimensional feature space, makes estimating pdf, a time consuming task. Furthermore, the adaptive implementation of mean-shift algorithm was presented. Continually, conventional distances calculation approaches were discussed, in particular Euclidean and Mahalanobis distance. Finally, there were defined two common clustering techniques, k-means algorithm and the fuzzy version of it, fuzzy k-means clustering algorithm.

In this work we have demonstrated an automated segmentation framework for brain MRI volumes, based on adaptive mean-shift clustering in the joint spatially and intensity feature space. The proposed algorithm consists of four steps. Firstly, preprocessing of the image is required. The brain extraction tool (BET) is utilized in order to remove brain scull and scalp pixels and afterwards a median filter for noise removal. Then an intensity normalization process sets the darkest percent of pixels to zero and rescales the brightest percent to 4095 in order to obtain similar dynamic ranges for all the three tissues and finally we recognize and extract the background of the image, so to store

only the brain tissues pixels for the rest of the procedure. The next step, is adaptive mean-shift procedure. As already mentioned, clustering takes place in the joint spatially and intensity feature space, for an overall dimensionality of three. Taking the gradient of Parzen Windows density estimation, we get the mean-shift vector which estimates the local maximum of probability density function (pdf) in an elegant way, as it doesn't need to estimate each pixel's pdf. Afterwards, by retaining only the maximum of the local maximum points withing small neighborhoods, we define the categories, "modes" as they called in which we classify all our pixels. A large compression of the data is occurred, as from tens of thousands pixels we lead to some few to several hundreds of modes. The next step, if required, that is to say the remaining modes after the mean-shift procedure is larger than 400, is Mahalanobis pruning modes step. Two modes are merged, if the mahalanobis distance of their intensity vector is smaller than a threshold. At each iteration, the threshold increases in a small ratio, until the remaining modes become less than 400. Finally, at the next step, pixels are classified to the three tissues, with the fuzzy k-means clustering algorithm. The output of the algorithm is both a segmentation map for all pixels and a membership matrix which include the probability of each pixel to belong on each brain tissue.

Knowing that in enhanced T1-weighted modality, the tumor necrotic area appears hypointense, while the solid area of the tumor around the necrotic area appears hyperintense and the edema cannot be distinguished from the GM and the WM, and that in T2-flair images, the edema and the solid tumor area appear hyperintense, while the necrotic area appears hypointense, we propose a way to automate tumor and edema pixel detection, using instead of three, for each brain tissue, four clusters. After identifying the cluster with the higher mean value, since it is the most probable to include the tumor region, we identify the largest connected component which it will be most likely the tumor area, in both MR modalities. If we subtract the tumor area of T1 from the tumor area of T2-flair, then we can also obtain the edema region.

We have tested the proposed algorithm in both simulated and real datasets investigating the effect of various parameters in its efficiency. The number of neighbors considered in mean-shift vector calculation and k parameter value can be chosen between various values, without affecting significantly the results, making the proposed algorithm **reproducible**. The algorithm also stays unaffected by the present of noise, even in high levels. Furthermore we compared and demonstrated the advantage of our proposed algorithm, with other common implementations that are used in brain segmentation field. For a dataset of 117 240*240 images, the proposed mean-shift algorithm needs about 17 minutes in order to be executed in a 2.26 GHz pc with 3 GB ram.

7.2 Recommendations for future work

A major advantage of the proposed algorithm is the fact that there is room for improvement. The bottleneck of this algorithm is computation of neighborhood queries. As mean-shift runs a large loop on the whole pixels matrix, where we have stored all brain tissue pixels, trying to calculate various neighborhood queries, we believe that a full C/C++ (the current implementation is fully in matlab) can reduce more than half the running time.

The current proposed algorithm can be expanded. For instance, we can use instead of pixel's spatial information voxel's spatial information, use more than one modality simultaneously, use also edge information, so to expand the feature space and take advantage of more information available, producing in this way better segmentation results.

Moreover, in this work we have proposed a way to deal with the major fact of detecting tumor and edema pixels. The initial results are very promising but a detailed validation work on this should be

carried out in the future. Additionally, since the brain structures usually appear very symmetric, a high level of asymmetry, around an axis of symmetry would strongly indicate some kind of abnormality (tumor, edema, cysts, etc.). Such information can be incorporated in order to refine the tumor detection and segmentation result.

8. REFERENCES

[1] http://en.wikipedia.org/wiki/Human_brain

[2] http://en.wikipedia.org/wiki/Cerebro-spinal_fluid

[3] http://en.wikipedia.org/wiki/Grey_matter

[4] http://en.wikipedia.org/wiki/White_matter

[5] "Training induces changes in white-matter architecture". Nature Neuroscience. http://www.nature.com/neuro/journal/v12/n11/full/nn.2412.html/. Retrieved 2009-10-11.

[6] http://en.wikipedia.org/wiki/Magnetic_resonance_imaging

[7] http://en.wikipedia.org/wiki/Computed_tomography

[8] Squire LF, Novelline RA (1997). Squire's fundamentals of radiology (5th ed.). Harvard University Press. ISBN 0-674-83339-2.

[9] http://www.mr-tip.com/serv1.php?type=db1&dbs=Gradient%20Echo

[10] http://www.mr-tip.com/serv1.php?type=db1&dbs=Echo%20Time

[11] http://www.mr-tip.com/serv1.php?type=db1&dbs=Repetition%20Time

[12] http://www.mr-tip.com/serv1.php?type=db1&dbs=Spin%20Echo

[13] http://en.wikipedia.org/wiki/Brain_tumor

[14] Frappaz D, Mornex F, Saint-Pierre G, Ranchere-Vince D, Jouvet A, Chassagne-Clement C, Thiesse P, Mere P, Deruty R. (1999). "Bone metastasis of glioblastoma multiforme confirmed by fine needle biopsy". *Acta neurochirurgica (Wien)* **141** (5): 551–552. PMID 10392217.

[15] http://en.wikipedia.org/wiki/Astrocytoma

[16] Νικόλαος Παπαμάρκος, "Ψηφιακή Επεξεργασία και Ανάλυση εικόνας" , Β. Γκιούρδας , pp 121-123

[17] Richard O. Duda, Peter E. Hart, David G. Stork ,"Pattern Classification ", 2nd Edition, Chapter 4, pp 3-6

[18] Richard O. Duda, Peter E. Hart, David G. Stork ,"Pattern Classification ", 2nd Edition, Chapter 1, pp 3-10

[19] R. Cardenes, S. K. Warfield, E. M. Macfas, J.A. Santana, and J. Ruiz-Alzola,"An efficient algorithm for multiple sclerosis segmentation from brain MRI", in Int. Workshop Comput. Aided Syst. Theory (EUROCAST), 2003 , pp 542-551

[20] Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley &amp (1973)

[21] http://nms.lcs.mit.edu/~aklmiu/6.838/L7.ppt

[22] Sang Young Lee, Young Kug Ham, Young-Hwan Kim and Rae-Hong Park, "Automatic Segmentation of Multi-Spectral MR Brain Images Using a Neuro-Fuzzy Algorithm", Department of Electronic Engineering, Sogang University, C.P.O. Box 1142, Seoul 100-611, Korea, NGT Co., Jayang B/D, 31-8, Munjong-dong, Songpa-Gu, Seoul, Korea

[23] http://fourier.eng.hmc.edu/e161/lectures/classification/node13.html

[24] http://www.myreaders.info/03_Back_Propagation_Network.pdf

[25] Douglas Reynolds "Gaussian Mixture Models ", MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA, pp.1-5

[26] K. Van Leemput, F.Maes, D. Vandeurmeulen, and P.uetens,"Automated model-based tissue classification of MR images of the brain" IEEE Trans. Med. Imag. Vol 18, no 10 , pp 897-908, Oct 1999

[27] A. Dmpster, N. Laird, and D. Rubin,"Maximum likelihood from incomplete data via EM algorithm" , J. Roy. Stat. Soc. B , vol 39, pp.1-38, 1977

[28] G. Dugas-Phocion, M. A Gonzalez Ballester , G. Malandain, C. Lebrun , and N.

Ayache , “Improved EM-based tissue segmentation and partial volume effect quantification in multi-sequence brain MRI” in Int. Conf. Med. Image Comput. Comput. Assist. Int. (MICCAI) , 2004 , pp 26-33

[29] K. Van Leemput, F. Maes, D. Vandermeulenand P.Suetens, “A unifying framework for partial volume segmentation of brain MR images” , IEEE Trans. Med. Imag,vol 22,no 1, pp 105-119 , Jan 2003

[30] H. Greenspan , A. Rurf , and J. Goldberger, “Constrained Gaussian mixture model framework for automatic segmentation of MR brain images ”, IEEE Trans. Med. Imag,vol 25, no 9 , pp 1233-1245, Sep 2006

[31] M. Prastawa, E.Bullitt, S.Ho, and G.Gerig , “Robust estimation for brain tumor segmentation” , in Int. Conf. Med. Image Comput. Comput. Assist. Int. (MICCAI),2003, pp 530-537

[32] J. P. W. Pluirn, J. B. A. Maintz, and M.A. Viergever, “Mutual information based registration of medical imges: A survey” , IEEE Trans. Med. Imag,vol 22, no8 , pp 9 86-1004 , Aug 2003

[33] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", IEEE Trans. Pattern Anal. Machine Intell., vol. 6, pp. 721-41, 1984.

[34] Karsten Held, Elena Rota Kops, J. Bernd Krause1, William M. Wells, Ron Kikinis, Hans-Wilhelm Müller-Gärtner, “Markov Random Field Segmentation of Brain MR Images”

[35] R. O. Duda and P. E. Hart, „Pattern Classification and Scene Analysis“, John Wiley and Sons, 1973.

[36] Y. Zhang, M. Brady and S. Smith , “Segmentation of brain MR images through a hidden markov random field model and the expectation maximization algorithm ” , IEEE Trans. Med. Imag,vol. 20, no 1, pp.45-57 , Jan 2001

[37] D. Comaniciu and P. Meer, “Mean-shift: A robust approach toward feature space analysis”, IEEE Trans. Pattern Ana. Mach. Intell. , vol. 24, pp. 603-619 May 2002

[38] W. Tao. H. Jin, and Y. Zhang, “Color image segmentation based on mean shift and normalized cuts” , IEEE Trans. Syst., Man, Cybern. B, Cybern., vol 37, no. 5, pp 1382-1389, Oct. 2007

[39] D. Comaniciu, V. Ramesh and P. Meer, “Kernel-based object tracking” , IEEE Trans. Pattern Ana. Mach. Intell. , vol. 25, no. 5, pp. 564-575 , May 2003

[40] K. Okada, D. Comaniciu and A. Krishnan, “Robust 3-D segmentation of pulmonary nodules in multi-slice CT images” in nt. Conf. Med. Image Comput. Comput. Assist. Int. (MICCAI) , 2004, pp. 881-889

[41] Arnaldo Mayer and Hayit Greenspan, “An adaptive Mean-shift Framework for MRI Brain Segmentation”, IEEE Trans. On Medical Imaging, vol. 28, no. 8 , pp 1238-1250 , Aug 2009

[42] Elsa D. Angelini, Olivier Clatz, Emmanuel Mandonnet, Ender Konukoglu, Laurent Capelle and Hugues Duffau, Current Medical Imaging Reviews, 2007, 3, 262-276 “Glioma Dynamics and Computational Models: A Review of Segmentation, Registration, and In Silico Growth Algorithms and their Clinical Applications”, Current Medical Imaging Reviews, 2007, 3, 262-276

[43] Tracqui P, Cruywagen GC, Woodward DE, “A mathematical model of glioma growth: the effect of chemotherapy on spatiotemporal growth”. Cell Prolif 1995; 28(1): 17-31.

[44] Woodward DE, Cook J, Tracqui P, ” A mathematical model of glioma growth: the effect of extent of surgical resection”, Cell Prolif 1996; 29(6): 269-88.

[45] Swanson KR, Alvord EC, Jr, Murray JD. “A quantitative model for differential

motility of gliomas in grey and white matter". Cell Prolif 2000; 33(5): 317-29.
[46] Jbabdi S, Mandonnet E, Duffau H, "Simulation of anisotropic growth of low-grade gliomas using diffusion tensor imaging", Magn Reson Med 2005; 54(3): 616-24.
[47] Gibbs P, Buckley DL, Blackband SJ, "Tumour volume determination from MR images by morphological segmentation" Phys Med Biol 1996; 41(11): 2437-46.
[48] Kennedy DN, Filipek PA, Caviness V. ", Anatomic segmentation and volumetric calculations in nuclear magnetic resonance imaging", IEEE Trans Med Imag 1989; 8: 1-7.
[49] Droske M, Meyer B, Rumpf M, "An adaptive level set method for interactive segmentation of intracranial tumors", Neurol Res 2005; 27(4): 363-70.
[50] Vaidyanathan M, Clarke LP, Velthuizen R.P, "Comparison of supervised MRI segmentation methods for tumor volume determination during therapy", Magn Reson Imaging 1995; 13(5):719-28.
[51] Clark M, Hall LO, Goldgof, DB, "Automatic tumor segmentation using knowledge-based techniques", IEEE Trans Med Imag 1998; 17(2): 187-201.
[52] Kaus MR, Warfield SK, Nabavi A, "Automated segmentation of MR images of brain tumors" Radiology 2001; 218(2): 586-91.
[53] Kaus M.R, Warfield SK, Jolesz FA, "Segmentation of meningiomas and low grade gliomas in MRI" in International Conference on Medical Image Computing and Computer Assisted Intervention. 1999. Cambridge, UK.
[54] Warfield SK, Kaus MR, Jolesz FA, "Adaptive template moderated spatially varying statistical classification" in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). 1998. Cambridge, MA, USA.
[55] Moonis G, Liu J, Udupa JK, "Estimation of tumor volume with fuzzy-connectedness segmentation of MR images" AJNR Am J Neuroradiol 2002; 23(3): 356-63.
[56] Nyul LG, Udupa JK "On standardizing the MR image intensity scale" Magn Reson Med 1999; 42: 1072-1081.
[57] Liu J, Udupa JK, Odhner D, "A system for brain tumor volume estimation via MR imaging and fuzzy connectedness" Comput Med Imaging Graph 2005; 29(1): 21-34.
[58] Fletcher-Heath LM, Hall LO, Goldgof DB, "Automatic segmentation of non-enhancing brain tumors in magnetic resonance images" Artif Intell Med 2001; 21(1-3): 43-63.
[59] Mazzara GP, Velthuizen RP, Pearlman JL, "Brain tumor target volume determination for radiation treatment planning through automated MRI segmentation" Int J Radiat Oncol Biol Phys 2004; 59(1): 300-12.
[60] Beyer GP, Velthuizen RP, Murtagh FR, "Technical aspects and evaluation methodology for the application of two automated brain MRI tumor segmentation methods in radiation therapy planning" Magn Reson Imaging 2006; 24(9): 1167-78.
[61] Zou KH, Wells WM, 3rd, Kikinis R, "Three validation metrics for automated probabilistic image segmentation of brain tumours" Stat Med 2004; 23(8): 1259-82.
[62] Prastawa M, Bullitt E, Ho S, "A brain tumor segmentation framework based on outlier detection. Med Image Anal 2004; 8:275-283.
[63] Richard O. Duda, Peter E. Hart, David G. Stork ,"Pattern Classification ", 2nd Edition, Chapter 4, pp 6-10

[64] J. R. Jimenez-Alaniz, V. Medina-Banuelos, and O. Yanez-Suarez,"Data-driven brain MRI segmentation supported on edge confidence and a priori tissue information," IEEE Trans. Med. Imag., vol. 25, no. 1, pp. 74–83, Jan. 2006.
[65] A. Mayer and H. Greenspan, "Segmentation of brain MRI by adaptive mean-shift," in Int. Symp. Biomed. Imag. (ISBI), 2006, pp. 319–322.
[66] J. Jimenez-Alaniz, M. Pohl-Alfaro, V. Medina-Banuelos, and O. Yanez-Suarez, "Segmenting brain MRI using adaptive mean shift," in IEEE EMBS, 2006, pp. 3114–3117.
[67] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean-shift and data-driven scale selection," in IEEE Int. Conf. Comput. Vis. (ICCV), 2001, vol. 1, pp. 438–445.
[68] B. Georgescu, I. Shimshoni, and P. Meer, "Mean-shift based clustering in high dimensions: A texture classification example," in IEEE Int. Conf. Comput. Vis. (ICCV), 2003, pp. 456–463.
[69] D. Comaniciu, "An algorithm for data-driven bandwidth selection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 2, pp. 281–288, 2003.
[70] Y. Cheng,"Mean Shift, Mode Seeking and Clustering", IEEE Trans. Pattern Ana. Mach. Intell. , vol. 17, no. 8, pp.790-799, Aug. 1995
[71] K. Fukunaga, "Introduction to Statistical Pattern Recognition", second ed. Academic Press, 1990, p. 535
[72] K. Fukunaga and L. D. Hostetler, "The estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", IEEE Trans. Information Theory, vol. 21, pp. 32-40 , 1975
[73] D.P. Bertsekas, "Nonlinear Programming", Athena Scientific, 1995
[74] http://en.wikipedia.org/wiki/Euclidean_distance, Euclidean distance
[75] http://en.wikipedia.org/wiki/Mahalanobis_distance, Mahalanobis distance
[76] Νικόλαος Παπαμάρκος, "Ψηφιακή Επεξεργασία και Ανάλυση εικόνας" , Β. Γκιούρδας , pp 343-344
[77] Νικόλαος Παπαμάρκος, "Ψηφιακή Επεξεργασία και Ανάλυση εικόνας" , Β. Γκιούρδας , pp 402-405
[78] http://www.cabiatl.com/mricro/mricron/index.html
[79] http://www.mathworks.com/access/helpdesk_r13/help/toolbox/images/getting3.html
[80] http://www.nitrc.org/projects/sri24/
[81] J. E. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. R. Stat. Soc.* vol. B-36, pp. 192-236, 1974.
[82] S. Z. Li, "Markov Random Field modeling in Computer Vision," *Springer*, New York, 2001.
[83] G. McLanchlan, and D. Peel, "Finite Mixture Models," *John Wiley and Sons*, New York, 2000.
[84] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Stat. Soc. B, vol. 39, pp. 1–38, 1977.
[85] http://en.wikipedia.org/wiki/Binomial_distribution
[86] http://en.wikipedia.org/wiki/Normal_distribution
[87] http://en.wikipedia.org/wiki/Cauchy_sequence

9. APPENDIXES

Appendix A. Probability Theory

A.1 Random Variables

In mathematics a random variable (or stochastic variable), as mentioned in [17], is a variable whose value is a function of the outcome of a statistical experiment. Random variables are used in the study of probability. They were developed to assist in the analysis of games of chance, stochastic events, and the results of scientific experiments by capturing only the mathematical properties necessary to answer probabilistic questions. Further formalizations have firmly grounded the entity in the theoretical domains of mathematics by making use of measure theory.

The language and structure of random variables can be grasped at various levels of mathematical fluency. Beyond an introductory level, set theory and calculus are fundamental. The concept of a random variable is closely linked to the term "random variate": a random variate is a particular outcome of a random variable.

There are two types of random variables: discrete and continuous. A discrete random variable maps events to values of a countable set (e.g., the integers), with each value in the range having probability greater than or equal to zero. A continuous random variable maps events to values of an uncountable set (e.g., the real numbers). In a continuous random variable, the probability of any specific value is zero, although the probability of an infinite set of values (such as an interval of non-zero length) may be positive. However, a random variable can be "mixed", having part of its probability spread out over an interval like a typical continuous variable, and part of it concentrated on particular values, like a discrete variable. This categorisation into types is directly equivalent to the categorisation of probability distributions.

A random variable has an associated probability distribution and frequently also a probability density function. Probability density functions are commonly used for continuous variables. A random variable has an associated probability distribution and frequently also a probability density function. Probability density functions are commonly used for continuous variables.

In order to completely understand the meaning of random variables allow us to give a characteristic example:

For a coin toss, the possible events are heads or tails. The number of heads appearing in one fair coin toss can be described using the following random variable:

$$X = \begin{cases} head \\ tail \end{cases} \qquad \textbf{(A.1)}$$

and if the coin is equally likely to land on either side then it has a probability mass function given by Eq.(A.2) :

$$p_x(x) = \begin{cases} \frac{1}{2}, & if\ x = head \\ \frac{1}{2}, & if\ x = tail \end{cases} \qquad \textbf{(A.2)}$$

It is sometimes convenient to model this situation using a random variable which takes numbers as its values, rather than the values *head* and *tail*. This can be done by using the real random variable *Y* defined as follows :

$$Y = \begin{cases} 1\ , & \textit{if heads} \\ 0\ , & \textit{if tails} \end{cases} \qquad \textbf{(A.3)}$$

A.2 Probability Space

In probability theory, a probability space or a probability triple is a mathematical construct that models a real-world process (or "experiment") consisting of states that occur randomly [18]. A probability space is constructed with a specific kind of situation or experiment in mind. One proposes that each time a situation of that kind arises, the set of possible outcomes is the same and the probability levels are also the same.

A probability space consists of three parts:

- A sample space, Ω , which is the set of all possible outcomes.
- A set of events, where each event is a set containing zero or more outcomes.
- The assignment of probabilities to the events, that is, a function from events to probability levels.

An outcome is the result of a single execution of the model. Recognizing that individual outcomes could be of little practical use, we formulate more complex events to characterize groups of outcomes. The collection of all such events is a σ-algebra *F*. Finally, we have to specify each event's likelihood of happening. We do this using the probability measure function, *P*. These three components $\Omega\ ,\ F\ ,\ P$ together constitute the probability space. A probability space is characterized as a “triple” because it has three components.

Once the probability space is established, it is assumed that “nature” makes its move and selects a single outcome, ω, from the sample space Ω. Then we say that all events from *F* containing the selected outcome ω (recall that each event is a subset of Ω) “have occurred”. The selection performed by nature is done in such a way that if we were to repeat the experiment an infinite number of times, the relative frequencies of occurrence of each of the events would have coincided with the probabilities prescribed by the function *P*.

A.3 Expected Value

In probability theory and statistics, the expected value (or expectation value, or mathematical expectation, or mean, or first moment) of a random variable is the integral of the random variable with respect to its probability measure, as referred in [19].

For discrete random variables this is equivalent to the probability-weighted sum of the possible values. Respectively, for continuous random variables with a density function it is the probability density-weighted integral of the possible values.

The term "expected value" can be misleading. It must not be confused with the "most probable value". The expected value is in general not a typical value that the random variable can take on. It is often helpful to interpret the expected value of a random variable as the long-run average value of the variable over many independent repetitions of an experiment.

In general, if X is a random variable defined on a probability space $\{\Omega\ ,\Sigma\ ,\ P\}$ then the expected value of X, denoted by $E(X)$ is defined as :

$$E(X) = \int_{\Omega} X \, dP \qquad \textbf{(A.4)}$$

When this integral converges absolutely, it is called the expectation of *X*. The absolute convergence is necessary because conditional convergence means that different order of addition gives different result, which is against the nature of expected value. Here the Lebesgue integral is employed. Note that not all random variables have an expected value, since the integral may not converge absolutely (e.g., Cauchy distribution). Two variables with the same probability distribution will have the same expected value, if it is defined.

If *X* is a discrete random variable with probability mass function *p(x)*, then the expected value becomes the Eq.(A.5):

$$E(X) = \sum_{i} x_i \, p(x_i) \qquad \textbf{(A.5)}$$

as in the gambling example mentioned above. If the probability distribution of *X* admits a probability density function *f(x)*, then the expected value can be computed as:

$$E(X) = \int_{-\infty}^{\infty} x \, f(x) \, dx \qquad \textbf{(A.6)}$$

It follows directly from the discrete case definition that if *X* is a constant random variable, i.e. $X = b$ for some fixed real number b, then the expected value of *X* is also *b*.

The expected value of an arbitrary function of *X*, *g(X)*, with respect to the probability density function *f(x)* is given by the inner product of *f* and *g* :

$$E(g(X)) = \int_{-\infty}^{\infty} g(x) \, f(x) \, dx \qquad \textbf{(A.7)}$$

A.4 Variance

In probability theory and statistics, the variance [20] is used as one of several descriptors of a distribution. In particular, the variance is one of the moments of a distribution. In that context, it forms part of systematic approach to distinguishing between probability distributions. While other such approaches have been developed, that based on moments has advantages of mathematical and computational simplicity.

The variance is a parameter describing a theoretical probability distribution, while a sample of data from such a distribution can be used to construct an estimate of this variance: in the simplest cases this estimate can be the sample variance.

If a random variable *X* has the expected value (mean) $\mu = E[X]$, then the variance of *X* is given by:

$$Var(X) = E[(x - \mu)^2] \qquad \textbf{(A.8)}$$

The variance of random variable *X* is typically designated as *Var(X)*, σ_x^2, or simply σ^2 (pronounced “sigma squared”). If a distribution does not have an expected value, as is the case for

the Cauchy distribution [21], it does not have a variance either.

If the random variable X is continuous with probability density function $f(x)$,

$$Var(X)=\int (x-\mu)^2 f(x)\,dx \qquad \textbf{(A.9)}$$

where:

$$\mu=\int x f(x)\,dx \qquad \textbf{(A.10)}$$

A.5 Covariance

Covariance of two random variables is a measure that shows the co-dependence of these two random variables [22]. If the entries in the column vector are:

$$X=\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \qquad \textbf{(A.11)}$$

that is to say random variables, each with finite variance, then the covariance matrix Σ is the matrix whose (i,j) entry is the covariance:

$$\Sigma_{ij}=cov(X_i, X_j)=E\left[(X_i-\mu_i)(X_j-\mu_j)\right] \qquad \textbf{(A.12)}$$

where:

$$\mu_i=E(X_i) \qquad \textbf{(A.13)}$$

is the expected value of the ith entry in the vector X. In other words, we have:

$$\Sigma=\begin{bmatrix} E[(X_1-\mu_1)(X_1-\mu_1)] & E[(X_1-\mu_1)(X_2-\mu_2)] & \dots & E[(X_1-\mu_1)(X_n-\mu_n)] \\ E[(X_2-\mu_2)(X_1-\mu_1)] & E[(X_2-\mu_2)(X_2-\mu_2)] & \dots & E[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n-\mu_n)(X_1-\mu_1)] & E[(X_n-\mu_n)(X_2-\mu_2)] & \dots & E[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix} \qquad \textbf{(A.14)}$$

In order to understand the meaning of this co-dependence, if the covariance of two random variables (for instance X and Y)is positive, practically it means that these variables are “changing” in the same way, in relevance with their mean value (when X takes higher values than his mean value $E(X)$, then Y in a respect way will take higher values than his mean value $E(Y)$ and when X takes lower values than his mean value $E(X)$, then Y in a respect way will take lower values than his

mean value *E(Y)*). Respectively, when the covariance of two random variables is negative, it means that these variables are "changing" in the opposite way, in relevance with their mean value (when *X* takes higher values than his mean value *E(X)*, then *Y* in a respect way will take lower values than his mean value *E(Y)* and when *X* takes higher values than his mean value *E(X)*, then *Y* in a respect way will take higher values than his mean value *E(Y)*). When the co variance of two random variables is zero, it means that these variables are irrelevant.

A.6 Binomial distribution

As referred in [85] the binomial distribution is the discrete probability distribution of the number of successes in a sequence of *n* independent *yes/no* experiments, each of which yields success with probability *p*. Such a success/failure experiment is also called a Bernoulli experiment or Bernoulli trial. In fact, when *n = 1*, the binomial distribution is a Bernoulli distribution. The binomial distribution is the basis for the popular binomial test of statistical significance.

It is frequently used to model number of successes in a sample of size *n* from a population of size *N*. Since the samples are not independent (this is sampling without replacement), the resulting distribution is a hypergeometric distribution, not a binomial one. However, for *N* much larger than *n*, the binomial distribution is a good approximation, and widely used.

In general, if the random variable *K* follows the binomial distribution with parameters *n* and *p*, we write $K \sim B(n, p)$. The probability of getting exactly *k* successes in *n* trials is given by the probability mass function:

$$f(k; n, p) = P(K = k) = \frac{n!}{(n-k)!\,k!}\, p^k (1-p)^{n-k} \quad \textbf{(A.15)}$$

The above equation can be understood as follows: we want *k* successes (p^k) among the *n* trials and $n-k$ failures ($(1-p)^{n-k}$).

The expected value of *K* is:

$$E[k] = np \quad \textbf{(A.16)}$$

and the variance is:

$$Var[K] = np(1-p) \quad \textbf{(A.17)}$$

This fact is easily proven. Suppose first that we have a single Bernoulli trial. There are two possible outcomes: 1 and 0, the first occurring with probability *p* and the second having probability $1-p$. The expected value in this trial will be equal to $\mu = 1 \cdot p + 0 \cdot (1-p) = p$. The variance in this trial is calculated similarly: $\sigma^2 = (1-p)^2 \cdot p + (0-p)^2 \cdot (1-p) = p(1-p)$.

A.7 Normal distribution

The normal (or Gaussian) distribution, is a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value. The graph of the associated probability density function is "bell"-shaped, and is known as the Gaussian function or bell curve. For a random variable ***x*** with mean value μ and variance σ^2,

the Gaussian distribution is described by the following equation:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \textbf{(A.18)}$$

The distribution with $\mu = 0$ and $\sigma^2 = 1$ is called the standard normal. The normal distribution is considered the most basic continuous probability distribution due to its role in the central limit theorem according to which the mean of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed. For this reason, the normal distribution is commonly encountered in practice, and is used throughout statistics, natural,social and biomedical sciences as a simple model for complex phenomena. For example, the observational error in an experiment is usually assumed to follow a normal distribution, and the propagation of uncertainty is computed using this assumption. Finally, it can be observed that a normally-distributed variable has a symmetric distribution around its mean. In figure A.1 it is shown some examples of normal distribution.

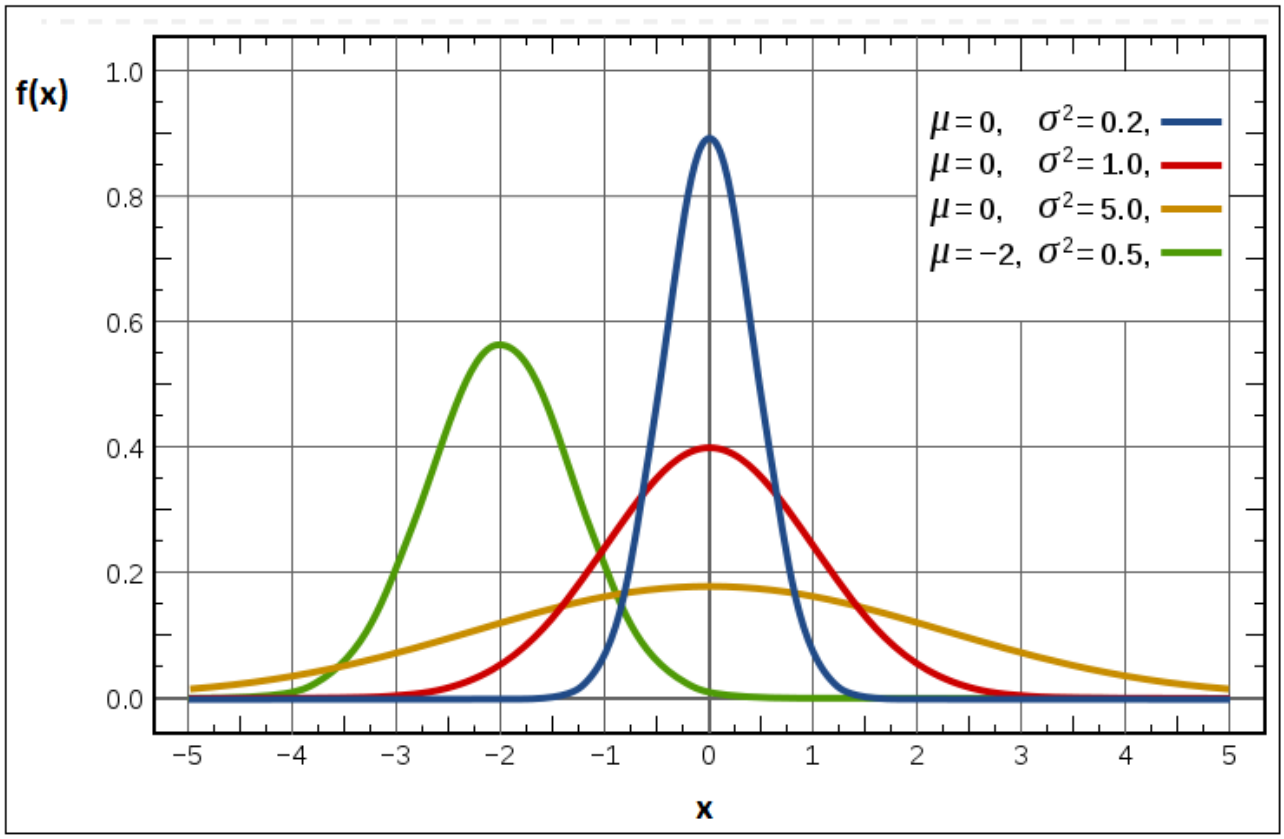


Fig. A.1: Four Normal (Gaussian) distributions. The blue with $\mu=0$, $\sigma^2 = 0.2$, the red with $\mu=0$, $\sigma^2 = 1$, the yellow $\mu=0$, $\sigma^2 = 5$ and last, green with $\mu=-2$, $\sigma^2 = 0.5$, We can observe that the lower the variance value is, the more the variable values are gathered around the mean value [86].

Appendix B. Proves of Mean-Shift Theorems

B.1 Proof of Theorem 4.1

In this point we should remind the Theorem 4.1 , presented in Section 4.2.4 *Convergence's Sufficient Condition*

Theorem 4.1 (Capture Theorem):

If the kernel K has a convex and monotonically decreasing profile, the sequences $\{y_j\}$ *j=1,2,... and* $\{\hat{f}_{h,K}(j)\}$ *j=1,2,.. . converge and* $\{\hat{f}_{h,K}(j)\}$ *j=1,2,.. . is monotonically increasing.*

Proof:

As mentioned in [37], because of the fact that the sequence $\hat{f}_{h,K}$ is bounded, since *n* is finite, it is sufficient to prove that $\hat{f}_{h,K}$ is strictly monotonic increasing, that is to say, prove that if $y_j \neq y_{j+1}$ then :

$$\hat{f}_{h,K}(j) < \hat{f}_{h,K}(j+1)$$

for *j=1,2,...* Without loss of generality, it can be assumed that $y_j = 0$ and thus, from equations (4.34) :

$$\hat{f}_{h,G}(x) = \frac{c_{g,d}}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}g\left(\left\|\frac{x-x_i}{h_i}\right\|^2\right)$$

and 4.39:

$$\hat{f}_{h,K}(j) = \hat{f}_{h,K}(y_j) \quad j = 1,2,...$$

we have:

$$\hat{f}_{h,K}(j) - \hat{f}_{h,K}(j+1) =$$
$$= \frac{c_{k,d}}{n}\sum_{i=1}^{n}\frac{1}{h_i^d}\left[k\left(\left\|\frac{y_{j+1}-x_i}{h_i}\right\|^2\right) - k\left(\left\|\frac{x_i}{h_i}\right\|^2\right)\right] \qquad \textbf{(B.1)}$$

The convexity of the profile k(x) shows that:

$$k(x_2) \geq k(x_1) + k'(x_1)(x_2 - x_1) \qquad \textbf{(B.2)}$$

for all $x_1, x_2 \exists [0,\infty)$, $x_1 \neq x_2$ and since $g(x) = -k'(x)$, equation (B.2) becomes:

$$k(x_2) - k(x_1) \geq g(x_1)(x_1 - x_2) \qquad \textbf{(B.3)}$$

Now using the equations (B.2) and (B.3) we obtain:

$$
\begin{aligned}
&\hat{f}_{h,K}(j+1) - \hat{f}_{h,K}(j) \\
&\geqslant \frac{c_{k,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right) \left[\|x_i\|^2 - \|y_{j+1} - x_i\|^2 \right] \\
&= \frac{c_{k,d}}{n} \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right) \left[2y_{j+1}^T x_i - \|y_{j+1}\|^2 \right] \\
&= \frac{c_{k,d}}{n} \left[2y_{j+1}^T \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} x_i g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right) - \|y_{j+1}\|^2 \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right) \right] \qquad \textbf{(B.4)}
\end{aligned}
$$

and recalling Eq.(4.38) :

$$
y_{j+1} = \frac{\sum_{i=1}^{n} x_i \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x - x_i}{h_i} \right\|^2 \right)} \qquad j = 1,2,...
$$

yields:

$$
\hat{f}_{h,K}(j+1) - \hat{f}_{h,K}(j) \geqslant \frac{c_{k,d}}{n} \|y_{j+1}\|^2 \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right) \qquad \textbf{(B.5)}
$$

The profile *k(x)* being monotonically decreasing for all $x \geq 0$ the sum $\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{x_i}{h_i} \right\|^2 \right)$ is strictly positive. Thus as long as $y_{j+1} \neq y_j = 0$, the right term of (B.5) is strictly positive, that is to say :

$$
\hat{f}_{h,K}(j+1) > \hat{f}_{h,K}(j) \qquad \textbf{(B.6)}
$$

Consequently, from Eq. B.6 the sequence:

$$
\{\hat{f}_{h,K}(j)\}_{j=1,2,...}
$$

is convergent. To prove the convergence of the sequence $\{y_j\}$, $j = 1,2,...$ equation (C.5) is rewritten for an arbitrary kernel location, $y_j \neq 0$. After some algebra we have:

$$
\hat{f}_{h,K}(j+1) - \hat{f}_{h,K}(j) \geqslant \frac{c_{k,d}}{n h^{d+2}} \|y_{j+1} - y_j\|^2 \sum_{i=1}^{n} g\left(\left\| \frac{y_j - x_i}{h} \right\|^2 \right) \qquad \textbf{(B.7)}
$$

Now, summing the two terms of (B.7) for indices $j , j+1,... , j+m-1$, it results that :

$$
\begin{aligned}
&\hat{f}_{h,K}(j+m) - \hat{f}_{h,K}(j) \\
&\geqslant \frac{c_{k,d}}{n} \|y_{j+m} - y_{j+m-1}\|^2 \sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{y_{j+m-1} - x_i}{h_i} \right\|^2 \right) + ...
\end{aligned}
$$

$$+\frac{c_{k,d}}{n}\|y_{j+1}-y_j\|^2\sum_{i=1}^{n}\frac{1}{h_i^{d+2}}\,g\left(\left\|\frac{y_j-x_i}{h_i}\right\|^2\right)$$

$$\geqslant\frac{c_{k,d}}{n}\left[\left\|y_{j+m}-y_{j+m-1}\right\|^2+\ldots+\left\|y_{j+1}-y_j\right\|^2\right]M$$

$$\geqslant\frac{c_{k,d}}{n}\left[\left\|y_{j+m}-y_j\right\|^2\right]M \qquad \textbf{(B.8)}$$

where M represents the minimum (always strictly positive) of the sum:

$$\sum_{i=1}^{n}\frac{1}{h_i^{d+2}}\,g\left(\left\|\frac{y_j-x_i}{h_i}\right\|^2\right) \qquad \textbf{(B.9)}$$

for all $\{y_j\}$, $j=1,2,\ldots$. Since $\{\hat{f}_{h,K}(j)\}$, $j=1,2,\ldots$ is convergent, it is also a Cauchy sequence [87]. This property, in conjunction with Eq. B.8 implies that $\{y_j\}$, $j=1,2,\ldots$ is a Cauchy sequence, hence, it is convergent in the Euclidean space.

B.2 Proof of Theorem 4.2

In this point we should remind the Theorem 4.2, presented in Section 4.2.6:

Theorem 4.2:

The cosine of the angle between two consecutive mean shift vectors is strictly positive when a normal kernel is employed:

$$\frac{m_{h,N}(y_j)^T\,m_{h,N}(y_{j+1})}{\left\|m_{h,N}(y_j)\right\|\left\|m_{h,N}(y_{j+1})\right\|}>0$$

Proof:

As discussed in [37], since convergence has already been achieved, we can assume, without loss of generality that $y_j=0$ and $y_{j+1}\neq y_{j+2}\neq 0$ (Eq. 4.22). Therefore, the mean shift vector $m_{h,N}(0)$ using the normal Kernel is given by:

$$m_{h,N}(0)=y_{j+1}=\frac{\sum_{i=1}^{n}\frac{1}{h_i^{d+2}}\,x_i\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right)}{\sum_{i=1}^{n}\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right)} \qquad \textbf{(B.10)}$$

Firstly, it will be showed that when the weights given by normal kernel centered at y_{j+1} , and adaptive *h,* the weighted sum of the projections of (y_j+1-x_i) onto y_{j+1} is strictly negative , i.e :

$$\sum_{i=1}^{n}\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{y_{j+1}-x_i}{h_i}\right\|^2\right) < 0 \qquad \textbf{(B.11)}$$

We can decompose space R^d into three domains:

$$D_1 = \left\{ x\in\mathbb{R}^d \;\middle|\; y_{j+1}^T x \le \frac{1}{2}\|y_{j+1}\|^2 \right\}$$
$$D_1 = \left\{ x\in\mathbb{R}^d \;\middle|\; \frac{1}{2}\|y_{j+1}\|^2 < y_{j+1}^T x \le \|y_{j+1}\|^2 \right\}$$
$$D_3 = \left\{ x\in\mathbb{R}^d \;\middle|\; \|y_{j+1}\|^2 < y_{j+1}^T x \right\} \qquad \textbf{(B.12)}$$

From (B.10) we can derive the equality :

$$\sum_{x_i\in D_2}\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right)$$
$$= \sum_{x_i\in D_1\cup D_3}\left(y_{j+1}^T x_i - \|y_{j+1}\|^2\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right) \qquad \textbf{(B.13)}$$

For $x \exists D_2$, we have:

$$\|y_{j+1}\|^2 - y_{j+1}^T x \ge 0 \qquad \textbf{(B.14)}$$

which implies:

$$\|y_{j+1}-x_i\|^2 = \|y_{j+1}\|^2 + \|x_i\|^2 - 2y_{j+1}^T x_i \ge \|x_i\|^2 - \|y_{j+1}\|^2 \qquad \textbf{(B.15)}$$

from where:

$$\sum_{x_i\in D_2}\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{y_{j+1}-x_i}{h_i}\right\|^2\right)$$
$$\le \exp\left(\left\|\frac{y_{j+1}}{h_i}\right\|^2\right)\sum_{x_i\in D_2}\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right) \qquad \textbf{(B.16)}$$

By introducing (B.13) in (B.16) we have:

$$\sum_{x_i\in D_2}\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{y_{j+1}-x_i}{h_i}\right\|^2\right)$$
$$\le \exp\left(\left\|\frac{y_{j+1}}{h_i}\right\|^2\right)\sum_{x_i\in D_1\cup D_3}\left(y_{j+1}^T x_i - \|y_{j+1}\|^2\right)\frac{1}{h_i^{d+2}}\exp\left(-\left\|\frac{x_i}{h_i}\right\|^2\right) \qquad \textbf{(B.17)}$$

By adding to both sides of (B.17) the quantity:

$$\sum_{x_i \in D_1 \cup D_3} \left(\|y_{j+1}\|^2 - y_{j+1}^T x_i \right) \frac{1}{h_i^{d+2}} \exp\left(-\left\| \frac{y_{j+1} - x_i}{h_i} \right\|^2 \right) \qquad \textbf{(B.18)}$$

It results that :

$$\begin{aligned} &\sum_{i=1}^{n} \left(\|y_{j+1}\|^2 - y_{j+1}^T x_i \right) \frac{1}{h_i^{d+2}} \exp\left(-\left\| \frac{y_{j+1} - x_i}{h_i} \right\|^2 \right) \\ &\leq \exp\left(\left\| \frac{y_{j+1}}{h_i} \right\|^2 \right) \sum_{x_i \in D_1 \cup D_3} \left(\|y_{j+1}\|^2 - y_{j+1}^T x_i \right) \frac{1}{h_i^{d+2}} \exp\left(-\left\| \frac{x_i}{h_i} \right\|^2 \right) \\ &* \left\{ \exp\left[-\frac{2}{h_i^2} \left(\|y_{j+1}\|^2 - y_{j+1}^T x_i \right) \right] - 1 \right\} \end{aligned} \qquad \textbf{(B.19)}$$

The right side of (B.19) is negative because the last product term has opposite signs in both the D_1 and D_3 domains and:

$$\left(\|y_{j+1}\|^2 - y_{j+1}^T x_i \right) > 0 \qquad \textbf{(B.20)}$$

Therefore, the left side of (B.19) is also negative, which proves (B.11).

We can use now (B.11) to write:

$$\|y_{j+1}\|^2 < y_{j+1}^T \frac{\sum_{i=1}^{n} x_i \frac{1}{h_i^{d+2}} \exp\left(-\left\| \frac{y_{j+1} - x_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} \exp\left(-\left\| \frac{y_{j+1} - x_i}{h_i} \right\|^2 \right)} = y_{j+1}^T y_{j+2} \qquad \textbf{(B.21)}$$

from where:

$$\frac{y_{j+1}^T \left(y_{j+2} - y_{j+1} \right)}{\|y_{j+1}\| \|y_{j+2} - y_{j+1}\|} > 0 \qquad \textbf{(B.22)}$$

and by taking into account Eq.(4.42):

$$\frac{m_{h,N}(y_j)^T m_{h,N}(y_{j+1})}{\|m_{h,N}(y_j)\| \, \|m_{h,N}(y_{j+1})\|} > 0$$