Use of Image Processing Techniques in the Diagnosis of Brain Tumor

Domenico de Gioia (d.degioia1@studenti.poliba.it)

ABSTRACT

Identifying brain tumors is a crucial challenge in medicine, given the complexity of the brain and the importance of early detection to improve treatments. This study proposes an approach based on MRI image processing techniques to automatically detect and analyze brain tumors through a four-step pipeline: pre-processing to improve visual quality, skull removal (skull stripping) to isolate brain tissue, segmentation of tumor areas, and feature extraction by testural analysis based on the gray-level co-occurrence matrix (GLCM). The results show high accuracy in segmentation and characterization of tumors, distinguishing between benign and malignant forms. The system, tested on MRI images in axial, coronal and sagittal views, proved to be an effective diagnostic aid, reducing the margin of error compared with traditional methods.

CONTENTS

1 INTRODUCTION **2**

2 RELATED WORKS **3**

3 DATASET **4**

4 METHOD **6**

4.1 PRE-PROCESSING 6
Loading data • Image Sharpening • Median filter • Gaussian filter • Contrast adjustment

4.2 SKULL STRIPPING 8
Thresholding • Removing small objects • Filling holes • Dilation and erosion • Brain extraction

4.3 SEGMENTATION 13

4.4 FEATURE EXTRACTION 16

5 RESULTS **17**

5.1 Evaluation metrics 17

5.2 Analysis of results 17

6 CONCLUSION **20**

References **21**

1 INTRODUCTION

What it is The brain, an integral part of the central nervous system, consists of the encephalon and spinal cord. The encephalon includes the brain, cerebellum, and brainstem, each with specific functions that regulate voluntary and involuntary activities of the body. A brain tumor is an abnormal growth of cells in brain tissues, which can be benign or malignant. These neoplasms may originate directly in the brain (primary tumors) or result from metastasis of tumors located elsewhere in the body (secondary tumors). Accurate segmentation of brain tumors using imaging techniques is critical for accurate diagnosis and planning effective therapeutic interventions [1].

Symptoms Symptoms of brain tumors vary according to the location and size of the lesion. Common signs include persistent headaches, nausea, vomiting, visual changes, speech difficulties, seizures, balance problems, and cognitive or behavioral changes. Early recognition of these symptoms is critical for early diagnosis and appropriate intervention.

Diffusion Central nervous system tumors are relatively rare, accounting for about 1.6% of all malignancies. In Italy, 6,122 new cases were registered in 2020, with a higher incidence in men than women. Survival at five years after diagnosis is around 25%, highlighting the need for improved diagnostic and therapeutic strategies.

Diagnosis The diagnosis of brain tumors is based on imaging techniques such as magnetic resonance imaging (MRI) and computed tomography (CT), which provide details on the location and extent of the lesion. Accurate tumor segmentation in images is essential to distinguish different tumor subregions and plan targeted interventions. Recently, the introduction of artificial intelligence algorithms has enabled more accurate and automated segmentation, improving diagnostic efficiency and assessment of treatment response.

2 RELATED WORKS

Segmentation of magnetic resonance imaging (MRI) images is an essential step in the detection of brain tumors. This process allows areas of interest (ROIs), such as tumor tissues, to be identified and separated from healthy tissues. There are several segmentation methodologies, each with specific characteristics, advantages and challenges. The main techniques used are reviewed below.

Thresholding Threshold-based segmentation is a direct approach that uses a threshold value to separate pixels based on their light intensity. For example, an MRI image can be divided into healthy tissue and tumor by choosing a threshold that discriminates typical intensity values for the tumor. This technique is particularly useful in situations where there is a sharp contrast between the tumor and the surrounding tissue. However, in images characterized by noise or uneven intensity changes, such as tumors with blurred edges, accuracy may be compromised.

Edge-based This method is based on detecting intensity changes between neighboring pixels, which indicate tumor boundaries. Edge detection is often improved through the use of noise reduction filters, followed by algorithms such as Canny's edge detector. This approach allows the identification of well-defined tumor contours, but may be ineffective in the presence of irregular or poorly delineated edges. In addition, sensitivity and specificity must be balanced to avoid the detection of irrelevant edges.

Region growing Region-based segmentation groups pixels into homogeneous areas using similarity criteria, such as light intensity or texture. A common method is region growth, which starts with a "seed" pixel and includes neighboring pixels that meet certain similarity criteria. This approach is particularly useful for identifying homogeneous tumors, but it requires careful seed selection and can be sensitive to noise and intensity variations within the tumor itself.

Gaussian models Gaussian Mixture Models (GMM) represent a statistical segmentation technique that uses a combination of Gaussian distributions to model the intensity of pixels in an image. Each Gaussian component represents a tissue class (e.g., tumor, white matter, gray matter, cerebrospinal fluid). The process begins with identifying the parameters of the Gaussian distributions using algorithms such as Expectation-Maximization (EM). Once the distributions are estimated, pixels are assigned to the class with the highest probability. GMMs are particularly useful in images with nonuniform intensities, as they can handle intraclass variations within a tumor. However, the accuracy of the method depends on the quality of the initial estimates and the ability to properly model the complexity of the image.

Atlas-based This methodology uses a predefined brain atlas, i.e., a standard anatomical map, to guide segmentation. The process involves comparing the patient's images with the atlas and identifying deviations that might indicate the presence of a tumor. Atlases are particularly useful for detecting abnormal tissue in complex brain areas, but they require high registration accuracy between the model and the patient's image.

Hybrid Techniques Hybrid techniques combine different methodologies to improve segmentation. For example, statistical models such as GMMs can be combined with edge- or region-based approaches to achieve greater accuracy. Another common hybrid approach involves integrating clustering algorithms (e.g., K-Means) with machine learning techniques to optimize the process.

Machine Learning Machine learning techniques, such as Support Vector Machines (SVMs), and deep learning techniques, such as Convolutional Neural Networks (CNNs), represent state-of-the-art solutions for MRI image segmentation. CNNs, for example, automatically analyze images to identify complex patterns associated with tumors. These patterns can segment tumors even under difficult conditions, such as noisy images or images with indistinct edges. However, they require large datasets for training and significant computational resources.

Ogni tecnica di segmentazione offre vantaggi specifici e presenta limiti legati alla complessità delle immagini e alle caratteristiche del tumore. La scelta della metodologia dipende dalle esigenze diagnostiche e dalla qualità dell'immagine MRI. L'uso di approcci ibridi e di algoritmi di apprendimento automatico avanzati rappresenta una promettente direzione per migliorare ulteriormente l'accuratezza e l'efficienza della segmentazione dei tumori cerebrali.

3 DATASET

The brain tumor dataset presented by Cheng, Jun (2017) [2] is a key resource for the study and classification of brain tumors based on magnetic resonance imaging (MRI). This dataset contains a total of 3064 contrast-enhanced T1-weighted images (CE-MRI) collected from 233 patients with three types of brain tumors (Fig. 1):

- Meningioma: 708 images.
- Glioma: 1426 images.
- Pituitary tumor: 930 images.

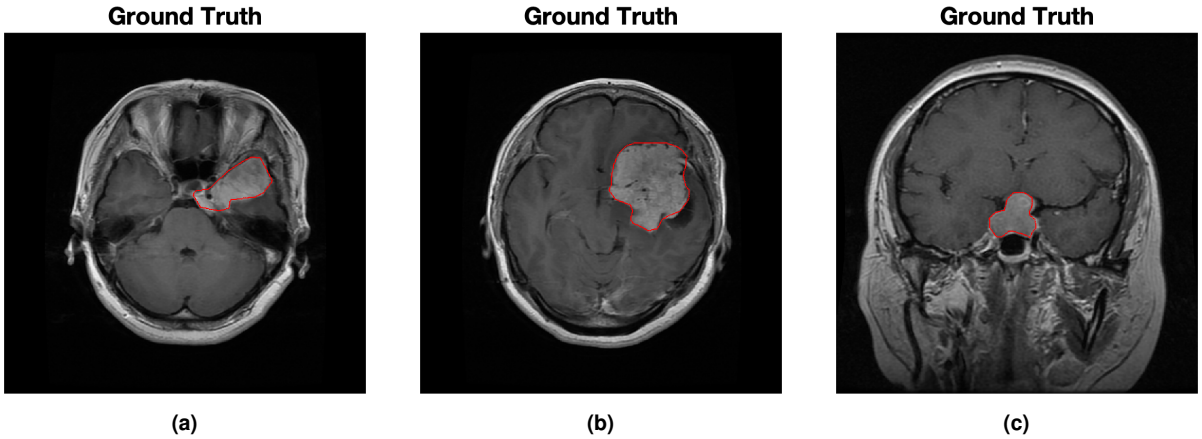


Figure 1. Illustrations of three typical brain tumors: (a) meningioma; (b) glioma; and (c) pituitary tumor. Red lines indicate the tumor border.

Due to file size limitations for the repository, the dataset was split into 4 .zip archives, each containing 766 images. In addition, indexes for 5-fold cross-validation are provided, making the dataset particularly useful for machine learning experiments.

The images are saved in .mat (Matlab) format, and each file contains a structure with the following fields:

- cjdata.label: Tumor label (1 for meningioma, 2 for glioma, 3 for pituitary tumor).
- cjdata.PID: ID of the patient.
- cjdata.image: Image data of the image.
- cjdata.tumorBorder: Coordinates of the discrete points on the tumor border. The points are represented as a vector, for example, $[x1, y1, x2, y2, ...]$, where $(x1, y1)$ are planar coordinates. This edge is manually outlined by radiologists and can be used to generate a binary mask of the tumor.
- cjdata.tumorMask: Binary mask in which pixels with value 1 indicate the region of the tumor.

In fig. 2 an example of the images related a single patient is provided.

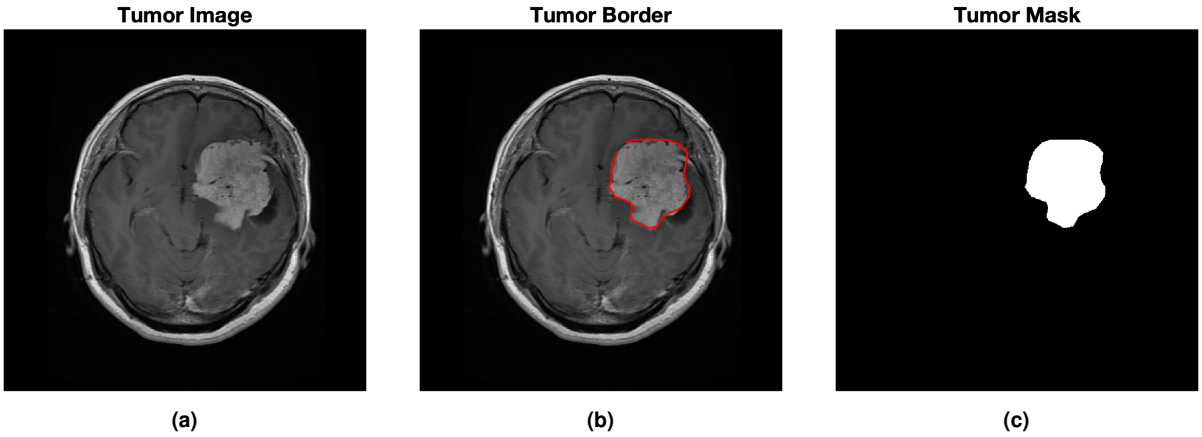


Figure 2. Complete case in the file *3.mat*

The images have an in-plane resolution of 512×512 pixels with a pixel size of 0.49×0.49 mm². Slice thickness is 6 mm with an interslice space of 1 mm.

Tumor boundaries were manually delineated by three experienced radiologists, ensuring high quality in image segmentation [3]. The dataset includes meaningful representations of lesions, selecting slices with large tumors.

4 METHOD

The paper "*Identification of Brain Tumor and Extracting its' Features through Processing of MRI*" by Lakmi et al. [5] presents a digital image processing method to detect brain cancer. The method proposed in this study focuses on MRI image processing, exploiting advanced techniques to improve accuracy and reduce the margin of human error. The procedure consists of several key stages, as shown in the figure 3. The first stage involves enhancing the original MRI images, through techniques such as grayscale conversion, application of median filters and contrast enhancement. Skull removal, called skull stripping, is essential to isolate brain tissue from external structures. This operation uses thresholding and morphological techniques to create a binary image and identify the actual brain. Once the skull is removed, the method detects the tumor by analyzing the nonzero pixels in the binary images, allowing the location of the tumor to be precisely identified. This approach not only facilitates localization but also provides a solid basis for further clinical analysis. The approach is designed to be effective and support medical professionals in diagnosing brain tumors and conducting further analysis, improving accuracy and reducing the margin of error compared with traditional methods.

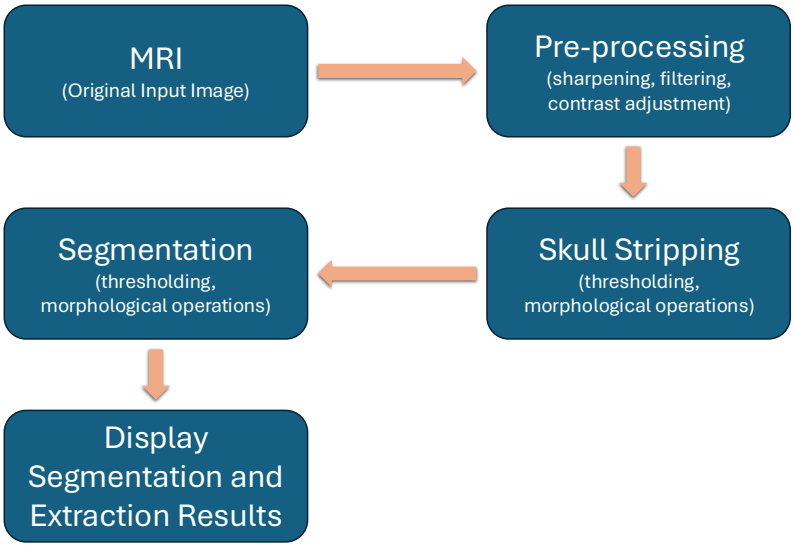


Figure 3. General outline of the proposed algorithm.

4.1 PRE-PROCESSING

Initially, the image is rescaled and converted to uint8 format to ensure standard compatibility with subsequent operations, then resized to a uniform size of 512x512 pixels. Next, a sharpening filter is applied to improve visual details and a median filter to reduce noise while preserving edges. Next, a Gaussian filter is used to further smooth the image. Finally, a contrast adjustment is performed to emphasize significant image features. This type of processing is crucial to prepare the data for further analysis.

4.1.1 Loading data

The code deals with loading and extracting information from a preformatted data file. The *load* function loads the specified file and stores it in the *data* variable. From this structure, the *cjdata* field, which contains several key pieces of information, is accessed. Specifically:

- *cjdata.image* is assigned to the *img* variable and represents the original MRI image data.
- *cjdata.tumorMask* is stored in *trueMask* and is a binary image in which values of 1 indicate tumor regions.
- *cjdata.tumorBorder* contains the coordinates of discrete points on the tumor border (e.g., $[x1, y1, x2, y2, ...]$), obtained through manual delineation.

This loading and mapping process allows easy access to essential data for later analysis.

```
data = load(filename);
cjdata = data.cjdata;
img = cjdata.image;
trueMask = cjdata.tumorMask;

img = rescale(img);
img = im2uint8(img);
img = imresize(img, [512 512]);
```

4.1.2 Image Sharpening

Sharpening makes the details and edges of an image more distinct and defined, emphasizing changes in intensity between neighboring pixels to enhance visual perception.

The Laplacian filter is a fundamental tool for detecting edges and details in an image [4]. It is a derivative operator that calculates the sum of the second partial derivatives of an image. The Laplacian of a function $f(x,y)$, representing a two-dimensional image, is defined as:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{1}$$

with

- $\frac{\partial^2 f}{\partial x^2}$ is the second partial derivative with respect to the horizontal direction x,
- $\frac{\partial^2 f}{\partial y^2}$ the second partial derivative with respect to the vertical direction y.

The Laplacian calculates the local variation of brightness in an image: it is positive or negative at points where there is a rapid change in intensity (e.g., at the edges), and close to zero in areas of uniform intensity. The second derivative has the property of amplifying abrupt changes, such as those present at the edges of an image. If an image contains an edge or sharp intensity transition, the first derivative of $f(x,y)$ will be elevated to that point. The second derivative is even more sensitive to these transitions, allowing the Laplacian filter to detect edges with high accuracy.

In a digital context, the Laplacian is approximated with a convolution mask (or kernel), which is applied directly to the pixels of the image. The most common mask for the Laplacian in 2D is:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

This mask operates on one pixel and its eight neighbors (both horizontally and vertically, and diagonally). The central value is -8, which implies that the value of the current pixel is compared with the sum of the values of its neighbors. This creates a local average subtraction effect, emphasizing areas where abrupt changes in intensity are observed.

In the process of image sharpening, the output of the Laplacian operator is subtracted from the original image to emphasize high-frequency details. This process is called unsharp masking. The final formula becomes:

$$f_{\text{sharp}}(x,y) = f(x,y) - \nabla^2 f(x,y) \tag{2}$$

This strengthens edges and fine details, as pixels in uniform areas, with $\nabla^2 f(x,y) \approx 0$, remain unchanged, while edges become more prominent.

4.1.3 Median filter

The median filter is a type of nonlinear filter designed to reduce "salt and pepper" noise, characterized by anomalous pixels that appear very light or very dark compared to neighboring pixels [4]. Unlike linear filters, such as the mean filter, the median filter replaces each pixel with the median of the surrounding pixels in a specific window (e.g., 3x3), making it particularly suitable for preserving relevant details, such as edges and lines. This operation does not use summation or multiplication, but sorting, making it nonlinear and robust with respect to extreme values, which do not affect the central value. Compared with linear filters that tend to blur edges, the median filter preserves high-contrast boundaries, making it ideal for detailed image analysis. Applied locally to each pixel in the image, the filter acts on the entire matrix, removing noise without compromising important structural details.

4.1.4 Gaussian filter

Gaussian filtering is a fundamental technique in image processing, used to reduce noise and improve visual quality. It is a linear filter that applies a convolution between the original image and a Gaussian function, described by the formula:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

In this formula, x and y represent the spatial coordinates, while σ is the **standard deviation**, which determines the intensity of blurring. Values of σ small produce light **blurring**, preserving details such as edges, while high values create more intense blurring, removing fine details but effectively reducing high-frequency noise.

The process involves constructing a two-dimensional kernel based on the Gaussian distribution and applying it by convolution, where each pixel is replaced by the weighted average of surrounding pixels.

In MATLAB, the *imgaussfilt* command implements the Gaussian filter in an optimized way.

4.1.5 Contrast adjustment

The MATLAB command *imadjust(img, stretchlim(img), [])* is a powerful function used to improve the contrast of an image by adjusting the intensity levels of pixels in a targeted manner. This process is based on a technique known as intensity level stretching, which enables the identification and expansion of the range of significant values in the image. The *stretchlim(img)* function calculates the lower and upper intensity limits that contain most of the relevant information, excluding any extreme values or noise that might adversely affect processing. Once these limits are calculated, the *imadjust* function applies a linear mapping, remapping the original pixel intensity values into the full available range, typically 0 to 1 for normalized images or 0 to 255 for 8-bit images.

This adjustment has a direct effect on the image: dark regions become darker, bright regions become brighter, and most importantly, the details present between these areas are amplified, significantly improving visual quality. The operation also acts on the image histogram, redistributing intensity values more evenly and widening the dynamic range, reducing congestion of similar values. This makes less obvious details in the original image more easily distinguishable.

```
img_sharpen = imsharpen(img);
img_median = medfilt2(img_sharpen, [3 3]);
img_smoothed = imgaussfilt(img_median, 0.5);
img_enhanced = imadjust(img_sharpen,stretchlim(img_sharpen),[]);
```

4.2 SKULL STRIPPING

This process focuses on processing magnetic resonance imaging (MRI) images to isolate brain tissue, removing skull and other irrelevant details. It begins by identifying regions of interest through an intensity threshold, followed by cleaning the image to remove small artifacts and fill in any gaps. Morphological operations, such as erosion with a disc-shaped structuring element, are then applied to further refine the result.

Eventually, the extracted brain tissue is combined with the original data, producing an image containing only the brain, ready for further analysis, such as detection and classification of any tumors.

4.2.1 Thresholding

One of the simplest approaches to image segmentation is the use of thresholding, a mechanism for switching from a grayscale image to a binary image, thus allowing the extraction of objects of interest from the image background. Binarization of the image is done by going to compare the value of each pixel with a threshold value T, verifying that the intensity of the pixel is greater or less than T. There are several methods for being able to choose the threshold value; in the present case, the Otsu [6] method was chosen because of its simplicity.

The idea behind this method is to maximize the interclass variance. Considering an image having L levels of gray intensity of pixels and assuming a histogram of normalized image gray levels, we will have components $p_i = n_i/MN$, where n_i is the number of pixels with intensity i, while M and N denote the size of the image. Assuming we want to divide the different pixels into two classes C_0 and C_1 using a threshold k such that pixels with levels between 1 and k belong to C_0 and those with levels from $k+1$ to L belong to C_1, it becomes possible to calculate the probability of occurrence of each class as:

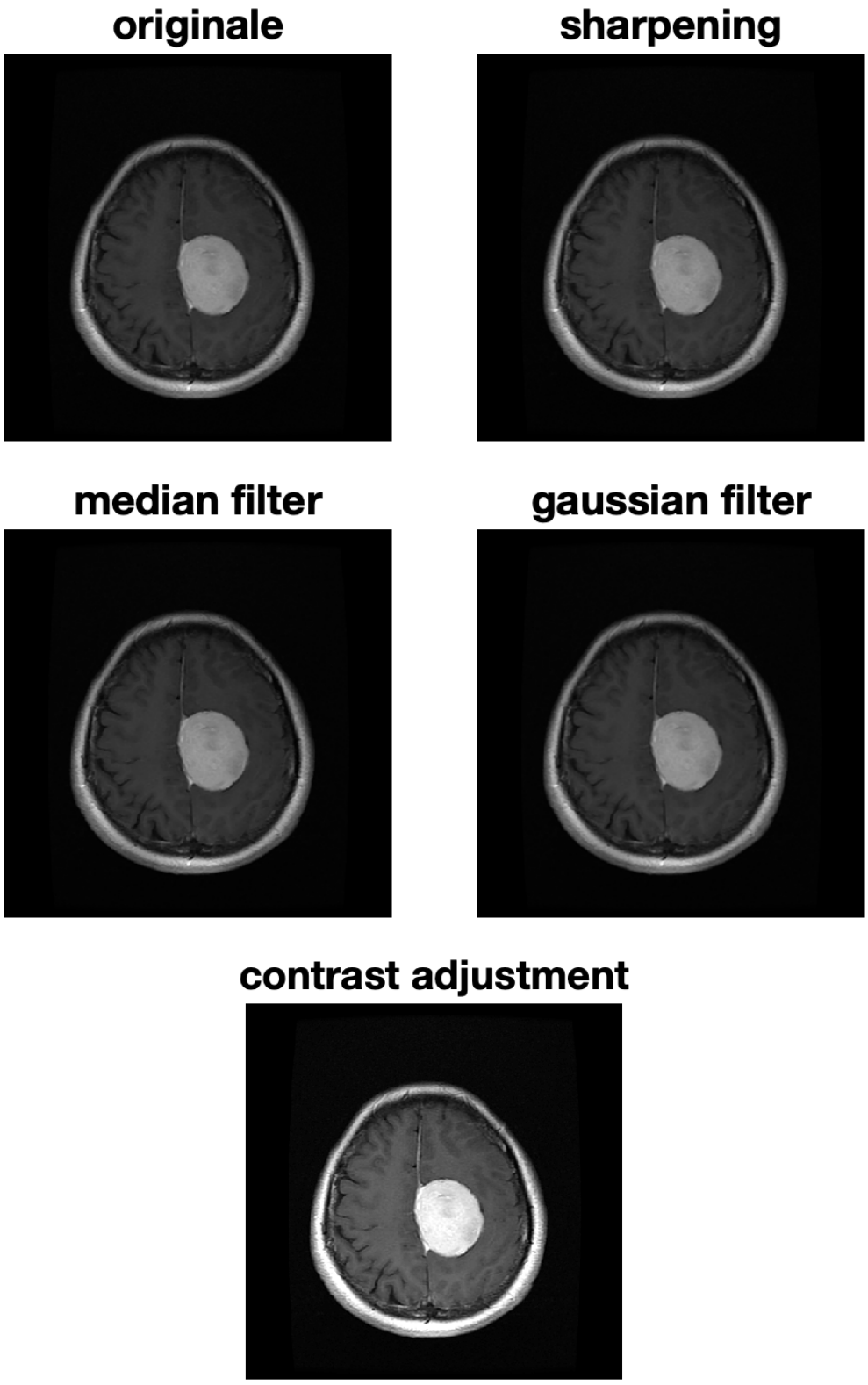


Figure 4. Steps of preprocessing.

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^{k} p_i = \omega(k) \tag{4}$$

$$\omega_1 = \Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \tag{5}$$

and the average level of each class respectively as:

$$\mu_0 = \sum_{i=1}^{k} i\Pr(i|C_0) = \sum_{i=1}^{k} ip_i/\omega_0 = \mu(k)/\omega(k) \tag{6}$$

$$\mu_1 = \sum_{i=k+1}^{L} i\Pr(i|C_1) = \sum_{i=k+1}^{L} ip_i/\omega_1 = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \tag{7}$$

with

$$\mu_T = \mu(L) = \sum_{i=1}^{L} ip_i \tag{8}$$

is the total average level of the initial image.
The variance of the two classes will be:

$$\sigma_0^2 = \sum_{i=1}^{k} (i-\mu_0)^2 \Pr(i|C_0) = \sum_{i=1}^{k} (i-\mu_0)^2 \Pr(i|C_0) p_i/\omega_0 \tag{9}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (i-\mu_1)^2 \Pr(i|C_1) = \sum_{i=k+1}^{L} (i-\mu_1)^2 \Pr(i|C_0) p_i/\omega_1 \tag{10}$$

In order to evaluate the effectiveness of the threshold at the k level, the normalized value is used:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2} \tag{11}$$

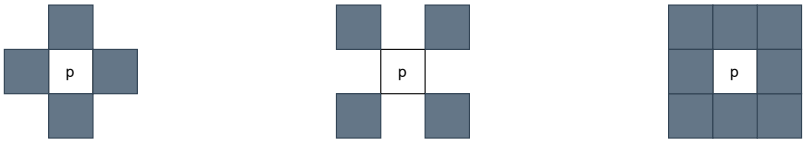
where σ_B^2 is the interclass variance and σ_T^2 is the global variance of image pixels, obtained as:

$$\sigma_T^2 = \sum_{i=1}^{L} (i-\mu_T)^2 p_i \tag{12}$$

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \tag{13}$$

The measure of interclass variance represents a measure of class separability, so since σ_T^2 is a constant value, η will also be a measure of separability, and therefore maximizing this value will be equivalent to maximizing σ_B^2. The threshold value k will be chosen, as mentioned by maximizing the interclass variance, that is:

$$\sigma_B^2(k^*) = \max_{1 \le k < L} \sigma_B^2(k) \tag{14}$$



(a) Pixel p and its 4-*neighborhood*.

(b) Pixel p and its diagonal neighborhood.

(c) Pixel p and its 8-*neighborhood*.

Figure 5. Connected components.

4.2.2 Removing small objects

A key concept in binary image processing is that of connected components. Specifically, a pixel p of coordinates $p(x,y)$ has two horizontal and as many vertical neighbors whose coordinates are $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, and $(x,y-1)$. This set of p neighbors, called $N_4(p)$, is shown in figure 5a. Instead, the four diagonal neighbors of p have coordinates $(x+1,y+1)$, $(x+1,y-1)$, $(x-1,y+1)$, and $(x-1,y-1)$. Figure 5b shows this set of diagonal neighbors of p, called $N_D(p)$. The union of the sets $N_4(p)$ and $N_D(p)$ in figure 5c represents the 8-*neighbor* of p, named $N_8(p)$.

Two pixels p and q are said to be 4-*adjacent* if $q \in N_4(p)$. Similarly, p and q are said to be 8-*adjacent* if $q \in N_8(p)$. A path between pixels p_1 and p_n is a sequence of pixels $p_1, p_2, ..., p_{n-1}, p_n$ such that p_k is adjacent to p_{k+1}, for $1 \leq k < n$. A path is 4-*connected* or 8-*connected*, depending on the type of adjacency used.

Two foreground pixels p and q are said to be 4-*connected* if there exists a 4-*connected* path between them, consisting entirely of foreground pixels. Similarly, they are 8-textitconnected if there exists an 8-textitconnected path between them. For any foreground pixel p, the set of all foreground pixels connected to it is called the connected component containing p.

A connected component is defined in terms of a path, and the path definition in turn depends on the type of adjacency used. This implies that the nature of a connected component depends on the form of adjacency chosen, with 4- and 8- adjacencies being the most common, which also affects the final number of connected components. In general, an adjacency of 8 reduces the final number of connected components compared to an adjacency of 4. Next, the pixels in each connected component are assigned a unique integer from 1 to the total number of connected components identified.

The Matlab function *bwareaopen(BW,P)* is an effective technique for cleaning binary images by removing components too small to be meaningful. Specifically, it uses the connected component labeling algorithm to identify all connected components in the image, calculates the size (number of pixels) of each component, and removes all components that have a number of pixels less than a specified threshold (P). This means that all pixels in these components will be set to 0 (black).

4.2.3 Filling holes

The algorithm for filling holes in a binary image takes advantage of morphological reconstruction, a fundamental operation in image processing. In general, in a binary image I there are foreground pixels, with value 1, and background pixels, with value 0. A hole is defined as a region of background pixels (0) completely surrounded by foreground pixels (1). The goal of the algorithm is to identify and fill these foreground regions, while keeping the background areas connected to the edges of the image unaffected.

The algorithm is based on a concept of morphological reconstruction, an operation used to extract or restore certain features in a binary image with respect to a given "mask." Suppose we have two images:

- a mask M defined as the complement of the original image, i.e., $M = 1 - I$;
- a marker J_0 as a starting point that will be expanded iteratively, initialized as a zero image that has value 1 only along the edges of the image where the original image has value 0.

Formally, the marker is defined as:

$$J_0(x,y) = \begin{cases} 1 - I(x,y) & \text{se } (x,y) \text{ è sul bordo dell'immagine,} \\ 0 & \text{altrimenti.} \end{cases} \tag{15}$$

At this point, morphological reconstruction is applied, which consists of iteratively expanding the marker under the constraint imposed by the M mask. The expansion is done using the expansion operation followed by an intersection with the mask, repeated until convergence is reached, that is, until the image no longer changes. Mathematically, this iteration is described by the formula:

$$J_{n+1} = (J_n \oplus B) \cap M \tag{16}$$

with J_n being the marker image at iteration n, $\oplus$ representing the dilation operation using a structural element B (typically a 3×3 matrix of all 1's), $\cap$ the intersection operation with the mask image M. The iteration continues until a steady state is reached in the process, i.e., $J_{n+1} = J_n$ which means that the expansion process no longer changes the image since no more pixels can be added without violating the constraints imposed by the mask.

Once the reconstruction is complete, the resulting image J contains all the background areas that are connected to the edges. To obtain the final image with the filled holes, the logical OR operation between the original image and the reconstructed image is used:

$$\text{filledImage} = I \vee J \tag{17}$$

This step ensures that the inner holes (which are not connected to the edge) are filled in without altering the outer background areas.

In MATLAB, this algorithm can be implemented manually using the function *imfill(I, "holes")*, which automatically fills the inner holes in a binary image using a procedure similar to the one just described.

4.2.4 Dilation and erosion

The morphological opening operation consists of two steps:

- Erosion: removes white pixels that are not completely surrounded by the structuring element. This reduces small protrusions or imperfections in the shape.
- Dilution (or dilation): rebuilds eroded areas, preserving larger shapes but removing small isolated areas.

Given a binary image we consider A a set of foreground pixels and B a structural element, both sets in $\mathbb{Z}^2$, we can then define A erosion through B as:

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{18}$$

In the formula we denote by z, the pixels in the foreground (equal to 1) and by B_z the translation of B by z, i.e., we assume that B is moved on each pixel of A and that for each pixel the indicated condition is verified. In the case where instead the image is grayscale, as in the present case, the image will be a digital function in the form $f(x,y)$ and the structural element will be $b(x,y)$. Assuming that these can take on discrete integer values contained in the set $\mathbb{Z}$, we define the grayscale erosion of f by b in the coordinates (x,y) as the minimum value of the image in the region coincident with $b(x,y)$ when the origin of b is on (x,y), i.e.:

$$[f \ominus b](x,y) = \min_{(s,t) \in b} \{f(x+s, y+t)\} \tag{19}$$

In both cases the operation involves running the structural element over the image in such a way that the origin is placed on all pixels of the image. In the case of a grayscale image, the result of the operation is obtained by assigning to each pixel the minimum value of f within the region covered by b.

Similarly, it is possible to define the dual operation called dilation, in which given A and B as in the equation 18, the dilation of A through B will be given by:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \tag{20}$$

In this formula, $\hat{B}$ represents the reflection of B with respect to its origin, that is, the set in which pixels having coordinates (x,y) are replaced with pixels having coordinates $(-x,-y)$. In practice, the result will be given by the set of all displacements z for which $\hat{B}$ overlaps at least one pixel of A.

Similarly, it is possible to define grayscale dilation as the maximum value taken by the image f within the window covered by $\hat{b}$, when the origin of $\hat{b}$ is in (x,y):

$$[f \oplus b](x,y) = \min_{(s,t)\in b} \{f(x-s, y-t)\} \tag{21}$$

The morphological opening of a set A with the structural element B, denoted as $A \circ B$, is defined as the erosion of A with B, followed by the dilation of the result with B. Formally:

$$A \circ B = (A \ominus B) \oplus B \tag{22}$$

with:

- $A \ominus B$ represents the erosion of A by B;
- $(A \ominus B) \oplus B$ represents the dilation of the result by B.

An equivalent formulation of the opening is as follows:

$$A \circ B = \cup\{(B)_z \mid (B)_z \subseteq A\} \tag{23}$$

where $\cup(\cdot)$ denotes the union operation between sets. This formulation has a simple geometric interpretation: $A \circ B$ is the union of all translations of B that fall entirely within A. Thus, it completely removes the regions of an object that cannot contain the structural element so as to

4.2.5 Brain extraction

This operation aims to isolate the tumor region within the previously processed brain image. The process uses a binary mask that identifies areas of interest (potentially containing the tumor) and applies it to the extracted brain tissue. This step allows the tumor to be clearly separated from the rest of the brain for further analysis or visualization. In figure 6, brain extraction steps are shown.

```
binary = img > thr;
binary_cleaned = bwareaopen(binary, 100);
binary_filled = imfill(binary_cleaned, "holes");
se = strel('disk', 40);
binary_mask = imerode(binary_filled, se);

brain = double(img) .* double(binary_mask);
brain = uint8(brain);
```

4.3 SEGMENTATION

A series of image processing techniques is employed to detect and segment a potential tumor in a brain MRI scan. The initial phase involves binarizing the image, which distinguishes regions of higher intensity, likely corresponding to significant tissues such as the brain or tumor, from less relevant areas. Subsequently, morphological erosion is performed using a diamond-shaped structuring element to eliminate noise and minor connections, thereby retaining only the most prominent regions. Following this, all connected components in the eroded image are analyzed, and the component with the highest pixel count is chosen, which is likely the primary tumor. This selected area is then reconstructed into a new binary image and dilated to recover any details that may have been lost during the erosion process. To ensure thorough coverage, any gaps within the region are filled, resulting in a binary mask that accurately delineates the area of interest. Finally, this mask is applied to the original image through element-wise multiplication, effectively isolating the tumor-associated pixels. This methodology facilitates precise tumor extraction while minimizing the impact of noise and extraneous details, representing a systematic and effective approach to medical image processing. In figure 7, tumor segmentation steps are shown.

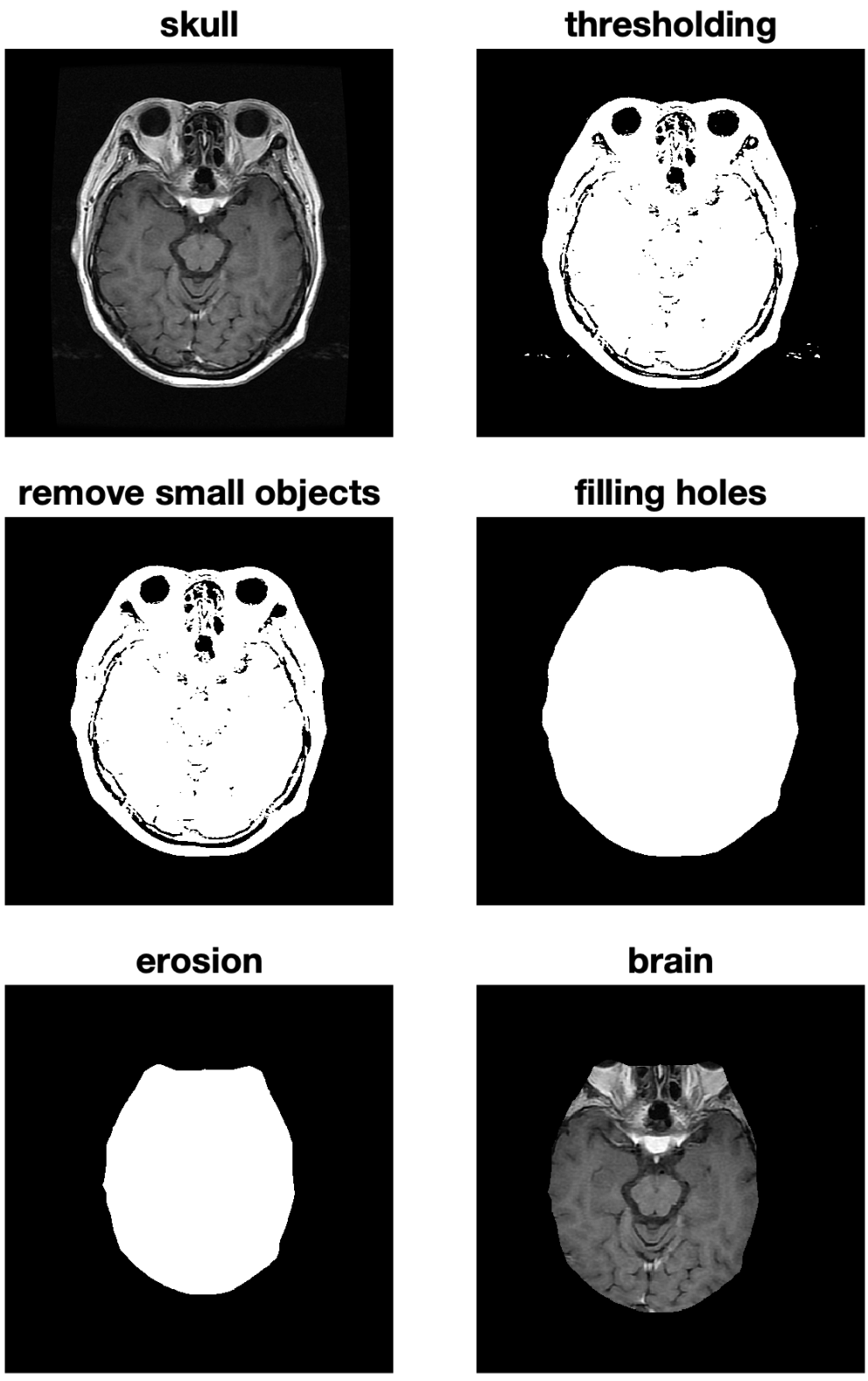


Figure 6. Skull stripping.

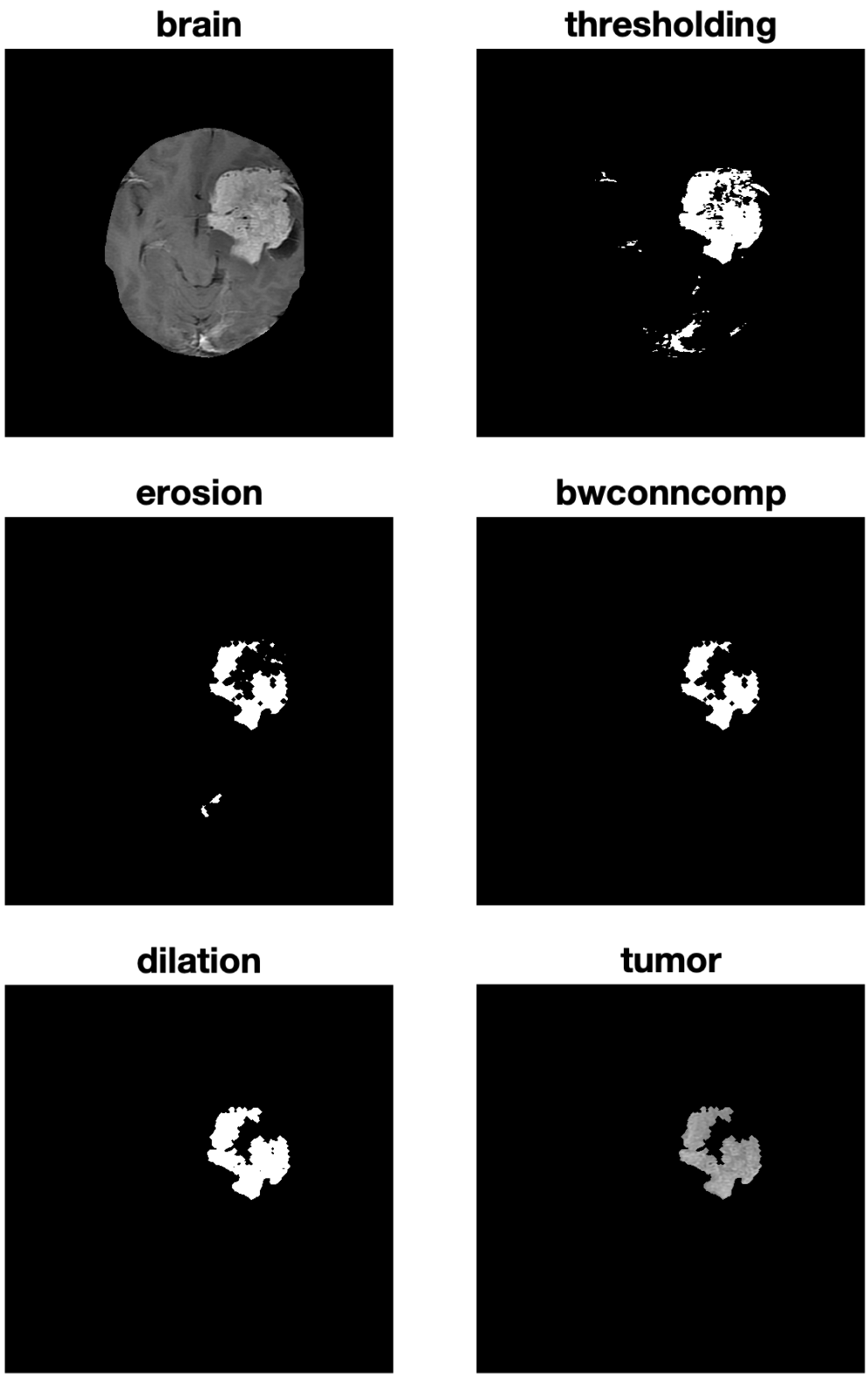


Figure 7. Tumor detection.

4.4 FEATURE EXTRACTION

Feature extraction based on textural analysis with Grey Level Co-occurrence Matrix (GLCM) is an advanced approach used to analyze textures and statistical features in brain MRI images. The GLCM is a matrix that represents the frequency with which specific combinations of gray intensities between adjacent pixels occur in an image, considering a certain distance and direction. Through this matrix, several textural properties useful for characterizing the tumor are calculated. These include contrast, which measures the variation in intensity between pixels, reflecting the level of detail in the tumor region; correlation, which assesses the linearity of the relationship between neighboring pixels, useful for identifying regular or disorganized structures. In addition, homogeneity quantifies the compactness of intensity distributions, while energy provides a measure of pattern repetition, indicative of texture uniformity. Finally, entropy represents the uncertainty or randomness in the distribution of intensities, revealing the complexity of the texture.

$$\text{Contrast} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-j)^2 P(i,j)$$

$$\text{Correlation} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j}$$

$$\text{Homogeneity} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{P(i,j)}{1+|i-j|}$$

$$\text{Entropy} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}P(i,j)\log(P(i,j))$$

$$\text{Energy} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}P(i,j)^2$$

These parameters, extracted from the GLCM, could allow the distinction between benign and malignant tumors and provide insight into tumor type and stage, facilitating medical classification and treatment.

5 RESULTS

5.1 Evaluation metrics

The Jaccard, Dice, and BFS metrics are key tools for assessing the quality of image segmentation, particularly in the context of brain tumor segmentation from MRI images.

Let M_s be the segmented tumor mask by the analyzed algorithm, and M_d be the tumor mask provided by the dataset.

The **Jaccard** metric measures the ratio of the intersection and union of the segmented and reference regions, providing an indication of the overall accuracy of segmentation: higher values indicate greater overlap and thus more accurate segmentation.

$$J(M_s, M_d) = \frac{|M_s \cap M_d|}{|M_s \cup M_d|} \tag{24}$$

The **Dice** metric represents a similar measure, but emphasizes the intersection of segmented regions over their overall size, being particularly useful for assessing the consistency of segmentation over areas of different sizes.

$$D(M_s, M_d) = \frac{2|M_s \cap M_d|}{|M_s| + |M_d|} \tag{25}$$

Finally, the **Boundary F1 Score** (BFS) assesses the correspondence between the boundaries of the segmented region and those of the reference region, highlighting the accuracy of contour detection.

$$BFS = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{26}$$

with

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{27}$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{28}$$

Together, these metrics provide an overall view of segmentation performance, combining overall accuracy, consistency, and detail precision.

5.2 Analysis of results

Analysis of the results shows that thresholding based on an empirically chosen fixed value (table 1) for test images demonstrated better performance than automatic thresholding using Otsu's method (table 2). Specifically, for metrics such as Jaccard, Dice and BFS, the fixed threshold method provided higher values in multiple images, indicating more accurate and consistent tumor segmentation. In addition, while fixed thresholding maintained a high level of acceptability for almost all images, Otsu's automated method showed greater variability in results and, in some cases, completely incorrect segmentations (e.g., Jaccard and Dice values of zero). These data suggest that automatic thresholding, despite its adaptability to illumination conditions, may not be as reliable in contexts with complex structures such as brain MRI images.

Img	Jaccard	Dice	BFScore	Acceptable
1	0.7886	0.8818	0.8260	Yes
2	0.5619	0.7195	0.5615	Yes
3	0.6195	0.7650	0.5361	Yes
4	0.7572	0.8618	0.6095	Yes
5	0.3454	0.5134	0.1791	Yes
6	0.7266	0.8416	0.8004	Yes
7	0.0000	0.0000	0.0000	No
8	0.6800	0.8095	0.7793	Yes
9	0.0650	0.1220	0.0000	-
10	0.8932	0.9435	0.9845	Yes
11	0.9242	0.9606	0.9903	Yes
12	0.9462	0.9723	0.9844	Yes
13	**0.9542**	**0.9765**	0.9916	Yes
14	0.9355	0.9666	**1.0000**	Yes
15	0.9016	0.9482	0.9723	Yes
16	0.0000	0.0000	0.0000	No
17	0.6802	0.8097	0.7019	Yes
18	0.8602	0.9248	0.9712	Yes
19	0.5430	0.7038	0.6974	Yes
20	0.6126	0.7598	0.4462	Yes
21	0.8047	0.8918	0.7617	Yes
22	0.9025	0.9487	0.9123	Yes
23	0.9268	0.9620	**1.0000**	Yes
24	0.7231	0.8393	0.7198	Yes
25	0.6040	0.7531	0.6449	Yes

Table 1. Valori di confronto delle metriche tramite metodo di sogliatura fissa.

Img	Jaccard	Dice	BFScore	Acceptable
1	0.0000	0.0000	0.0092	No
2	0.5811	0.7351	0.6016	Yes
3	0.8455	0.9162	0.7117	Yes
4	0.7788	0.8756	0.6224	Yes
5	0.3253	0.4909	0.1189	Yes
6	0.0916	0.1679	0.2515	No
7	0.0000	0.0000	0.0000	No
8	0.6659	0.7994	0.7728	Yes
9	0.0976	0.1779	0.0000	No
10	0.8025	0.8904	0.7937	Yes
11	0.9181	0.9573	0.9782	Yes
12	0.9429	0.9706	0.9894	Yes
13	**0.9430**	**0.9707**	0.9932	Yes
14	0.9250	0.9610	**1.0000**	Yes
15	0.8637	0.9268	0.9320	Yes
16	0.0000	0.0000	0.0000	No
17	0.6669	0.8001	0.6936	Yes
18	0.8016	0.8899	0.9163	Yes
19	0.0246	0.0480	0.0816	No
20	0.0825	0.1525	0.0000	No
21	0.1010	0.1835	0.0000	No
22	0.1129	0.2029	0.0000	No
23	0.0988	0.1799	0.0000	No
24	0.7340	0.8466	0.7199	Yes
25	0.6062	0.7548	0.6470	Yes

Table 2. Valori di confronto delle metriche tramite metodo di sogliatura Otsu.

6 CONCLUSION

In conclusion, this report has demonstrated how the application of image processing techniques on MRI scans is an effective and innovative approach for the detection and analysis of brain tumors. Through a well-structured process, which includes fundamental steps such as preprocessing, skull stripping, segmentation, and feature extraction, it is possible to accurately identify tumor areas and gather key information to support diagnostic and medical decision making. The results obtained indicate high accuracy in the detection of tumor lesions, as well as the ability to distinguish between different tissue types and tumor features, thus making a significant contribution to classification and therapeutic planning.

This methodology not only reduces the margin of error compared with traditional methods, but also offers significant support to health care professionals by improving their ability to analyze complex images and make informed decisions. The approach proposed here can thus be considered a step toward more precise and personalized medicine, especially in the sensitive context of brain tumor diagnosis and treatment.

Looking to the future, there are many interesting and potentially revolutionary developments ahead. One of the most promising areas is the integration of advanced machine learning and artificial intelligence techniques, particularly through the use of deep neural networks (Deep Learning). These tools have the potential to further improve the accuracy of segmentations and analyses, while reducing the need for manual intervention and optimizing processing time. In addition, the expansion of available datasets, both in terms of quality and diversity, may help increase the robustness and reliability of the proposed algorithms, making them more applicable to a wide range of real-world clinical cases.

In summary, this research represents a solid foundation on which to build further technological and scientific advances, with the ultimate goal of improving patients' quality of life and providing physicians with increasingly reliable tools for dealing with the complexity of oncology diagnoses.

REFERENCES

[1] AIRC. Tumore del cervello. https://www.airc.it/cancro/informazioni-tumori/guida-ai-tumori/tumore-al-cervello. Accesso: 10 dicembre 2024.

[2] Jun Cheng. brain tumor dataset. *aaa*, 4 2017.

[3] Jun Cheng, Wei Huang, Shichuan Cao, Chuansheng Yang, Jia Yang, Xu Shan, Dagan Feng, and Zhigang Chen. Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PloS One*, 10(10):e0140381, 2015.

[4] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc., USA, 2003.

[5] K. W. D. T. Lakmi, G. P. S. N. Pathirana, and T. C. Sandanayake. Identification of brain tumor and extracting its' features through processing of mri. In *Proceedings of the 2020 International Conference on Information Technology Research (ICITR)*, pages 1–8. IEEE, IEEE, 2020.

[6] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(1):62–66, jan 1979.