



Processing MRI images of brain tumors in Python

GHITA GHOU LI

Polytech Lyon engineering school, specializing in Biomedical Engineering.

ABSTRACT: This paper presents an interactive Python-based framework for the semi-automatic segmentation and analysis of brain MRI images. The proposed approach combines multiple image-processing techniques Gaussian smoothing, canny edge detection, morphological refinement, texture mapping, and histogram-based peak analysis to enhance the detection and characterization of tumor regions. An interactive Dash interface enables real-time parameter adjustment and region-of-interest selection, facilitating reproducible exploration of MRI data and providing information on tissue heterogeneity and any anomaly. This method and result can be used for further MRI preprocessing and lesion delineation, and suggest a methodological foundation for future extensions toward fully automated segmentation or deep learning.

KEYWORDS: Brain MRI, Tumor Segmentation, Image Processing, Gaussian Smoothing, Canny Edge Detection, Texture Analysis, Histogram Peak Detection, Dash Application, Semi-automatic Segmentation, Medical Imaging.

Received 01 Aug., 2026; Revised 05 Aug., 2026; Accepted 07 Aug., 2026 © The author(s) 2026.

Published with open access at www.questjournals.org

I. INTRODUCTION

A tumor is a mass of varying size resulting from the abnormal multiplication of cells. The cell is the basic unit of all tissues in living organisms. The human body contains more than two hundred different types, each with a specific function: muscle cells, nerve cells, bone cells, epithelial cells, among others. Each cell has a precise role to play and a limited lifespan. Cells are constantly renewing themselves: some age, die naturally (a process called apoptosis), and are replaced by new ones. However, for reasons that are still poorly understood, some cells can malfunction: they cease to respond to the natural signals that regulate their life cycle and begin to divide uncontrollably. This phenomenon can lead to the formation of a tumor.

Brain tumors constitute a very heterogeneous group of pathologies. They can be benign or malignant, that is, respectively non-cancerous or cancerous. Their severity and impact on the patient's health depend on several factors. Location plays a significant role; a small growth in a vital area of the brain, such as areas responsible for speech, movement or breathing, can result in significant complications. The size of the tumor can also affect its clinical presentation: A large mass can put pressure on adjacent brain structures and significantly affect how they function. Additionally, how aggressive the tumor is, meaning its rate of growth, will have a significant impact on how quickly the patient's overall health deteriorates, requiring emergency medical intervention. Brain tumors can cause a wide range of clinical symptoms, depending on their characteristics (i.e., their nature, location, size and rate of growth). Some of the most common symptoms include headaches that are persistent; visual defects; seizures; weakness / paralysis of an extremity; speech deficits; difficulty with memory or concentration; etc.

Therefore, an accurate diagnosis of the tumor is essential in determining appropriate treatment options. The diagnosis of a brain tumor is achieved via a series of critical steps. The first step is a thorough clinical and neurological examination to evaluate the patient's overall motor, sensory, and cognitive function. The next step is obtaining imaging tests that will allow for visualization of the tumor in the patient. The final step in all tumor diagnoses is biopsy, whereby a small sample of tumor tissue is obtained, and a histopathological examination is performed to determine the cytological characteristics of the tumor.

Numerous imaging tools are used to explore the brain. Among the most common are CT scans (or computed tomography), magnetic resonance imaging (MRI), positron emission tomography (PET), and brain scintigraphy. MRI is now considered a gold standard in the diagnosis of brain tumors.

In this article, we will focus on processing magnetic resonance imaging (MRI) using a Python program in the context of brain tumor diagnosis.

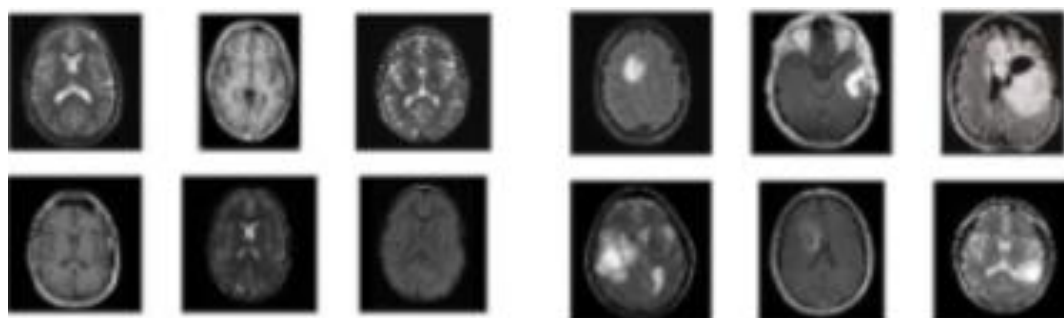
II. MAGNETIC RESONANCE IMAGING

Used in a medical context, magnetic resonance imaging (MRI) scans represent a particularly rich source of information for healthcare professionals. This high-resolution visual data allows for detailed observation of the internal structures of the human body and has become indispensable in the diagnosis and monitoring of numerous pathologies, especially neurological ones. Thanks to technological advances, MRI now generates digital images of exceptional quality, thus providing a solid foundation for increasingly sophisticated computerized processing.

In this context, MRI images provide image processors whether artificial intelligence software or medical imaging experts with essential raw material. This unprocessed data serves as the basis for many different analytical algorithms to process and analyze with the goal of extracting medically relevant information automatically. The tools developed using these data are primarily intended to assist practitioners in carrying out the complex tasks of diagnosing, classifying, and characterizing brain tumors. Consequently, these algorithms or processes make it possible to identify, segment, quantify, and analyze a tumor. Together, these capabilities contribute to the emergence of more personalized, predictive, and effective medicine [1].

However, despite the wealth of information these images provide, their interpretation remains a time-consuming task when performed traditionally, i.e., manually. It requires not only time and considerable expertise but is also subject to inter-observer variability. It is possible for two experts looking at the same image to come to very different conclusions about what they see. This fact has led to a great deal of research being done to automate this type of analysis. Automation is accomplished by using segmentation. Segmentation is the process of accurately identifying (delimiting) areas of interest in the image, including anatomical structures (brain, lung, and breast) as well as abnormalities (tumor).

Segmentation is the first step in the automated processing of medical images. Segmentation not only allows researchers to locate abnormal areas, but it also enables researchers to conduct quantitative analyses on those areas so that informed and faster decisions can be made.



The MRI dataset used in this work is a publicly available open-source dataset containing both tumors and tumor-free images [2]. The dataset is shared online [3].

III. TUMOR SEGMENTATION METHODS AND RESULTS FROM MRI IMAGES

Identifying and segmenting brain tumors are important parts of determining the patient's diagnosis and tracking their medical condition. Due to advancements in MRI technology, we can now acquire high-quality brain images that make it easier to see normal and abnormal structures. Because of this level of detail, analyzing medical images by hand is difficult, subject to bias and error.

With this in mind, I developed a semi-automatic system for doing the segmentation of brain tumors from MRI images. This application is written in Python, using the Dash framework, with functions for image processing, interactive display, and statistical calculations (histograms, peak detection).

The purpose of this application is to provide an intuitive platform for examining/gathering information from medical images.

IV. IMAGE LOADING AND PREPARATION

After the MRI image has been loaded via OpenCV, the image will then need to be converted from BGR to RGB, in order to be sure that the image can be displayed accurately, and afterward will need to undergo a conversion to grayscale followed by normalization (to a scale of 0 - 1) for providing a consistent foundation to support the determination of texture, Gaussian smoothing (blurring to create other derivative images), and to perform edge detection on the MRI image.

This preprocessing process is essential for achieving higher levels of accuracy in analysis and greater quality during segmentation.

The main operations are illustrated below:

```
img = cv2.imread(img_path)
img_rgb=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY) / 255.0
```



Figure 1 : Interface de l'application Python/Dash permettant la visualisation et le traitement d'images IRM cérébrales.

The application is built with Dash utilizing Bootstrap (dbc) in order to create an up-to-date and user-friendliness interface that has been specifically designed for interpreting MRI type images. The application consists of two sections.

-MRI Image Visualization (Left section) : This section allows for direct interaction with the image by allowing the user to draw a region of interest (ROI) directly over the image with the tool provided for drawing regions. This allows users to select specific areas, concentrating the processing there.

-Processing Controls (Right section): Adjustable settings for pre-processing and analysis can be established here. The settings include Canny filter thresholds, various smoothing settings (sigma) and the ability to select various display options such as texture, segmentation, and ROI rendering.

A dynamic histogram is also shown below these two sections, the histogram dynamically represents the current pixel intensities within the current ROI and assists in creating a quantitative interpretation of the data and provides a means of interactively and accurately evaluating MRI images.



Figure 2: Tumor segmentation assisted by image parameters and statistical analysis of regions of interest.

V. SEMI-AUTOMATIC SEGMENTATION WITH CANNY AND GAUSSIAN FILTERING

Semi-automatic segmentation isolates a region of interest (ROI) on an MRI image by combining user interaction with automated image processing. The process implemented in the application involves several complementary steps, each playing a specific role in obtaining sharp and reliable contours.

5-1 selecting the region of interest

In the beginning, the user creates a formed area on the picture through the draw facility provided with the image. This process produces a binary mask indicating the pixels of the created region and will be analyzed to produce an outcome. Pixels outside of this formed area will be excluded, thereby allowing the user to concentrate their analysis only on the areas that are important for their analysis to reduce any unnecessary noise.

```
mask = np.zeros_like(img_gray, dtype=np.uint8)
cv2.fillPoly(mask, [polygon], 255) # polygon = coordonnées du tracé utilisateur
roi = img_gray.copy()
roi[mask == 0] = 0 # isolation de la ROI
```

5-2 image smoothing using gaussian filtering

Once the ROI is done, the image is smoothed with a Gaussian filter (`gaussian_filter`) to reduce noise while preserving important structures.

Low sigma (0.5–1): preserves fine details but introduces more noise.

High sigma (>2): significantly smooths, reduces noise, but may blur small edges.

The user can adjust this parameter in real time using a slider.

```
from scipy.ndimage import gaussian_filter
if sigma > 0:
    roi = gaussian_filter(roi, sigma=sigma)
```

5-3 edge detection by canny

Edge detection is performed using Canny's algorithm, which uses two thresholds to identify edges:

Low threshold: pixels considered weak edges.

High threshold: pixels considered strong edges.

If the thresholds are too low → many false edges (noise).

If they are too high → incomplete or lost edges.

```
import cv2
low, high = int(canny_thresholds[0]*255), int(canny_thresholds[1]*255)
edges = cv2.Canny(roi_clahe, low, high)
```

5-4 morphological edge cleaning

To obtain continuous and sharp edges, morphological closure is applied. This operation combines dilation and erosion to fill small discontinuities in the edges.

```
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
```

```
edges_clean = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, kernel)
```

The result is a clean segmentation, where the tumor boundaries appear sharp and well-defined.

5-5 Interaction and dynamic visualization

This entire process is integrated into the Dash application:

The user can draw and modify the ROI directly on the image. The Canny and sigma threshold sliders allow for instant visualization of the effect of adjustments. A dynamic histogram of intensities in the ROI is updated in real time, providing quantitative feedback for analysis.

```
intensities = img_gray[mask != 0] * 255
counts, bins = np.histogram(intensities, bins=50, range=(0, 255))
```

5-6 example result

The user surrounds the tumor with a polygon to extract a Region of Interest, apply a smoothing operation (sigma = 0.8) to reduce noise and edge detection using Canny Edge Detection algorithms (threshold values from 0.51 to 0.71), which creates well-defined and accurate edges. The resulting image has red outlines around a tumor area, which makes it easier to see and prepare for further analyses like measuring surface area or intensity-related studies.

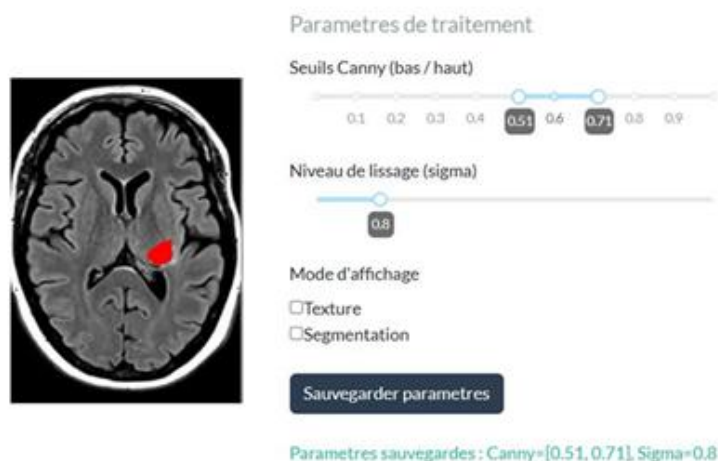


Figure 3 : Segmentation IRM par contours Canny Détection précise de la tumeur par lissage ($\sigma = 0.8$) et seuils Canny (0.51–0.71).

VI. DISPLAY MODES: SEGMENTATION AND TEXTURE

In this application, I have added two essential display modes for analyzing brain lesions: Segmentation and Texture. These modes allow you to explore the image from two different perspectives: pixel intensity and local tissue structure.

6-1 Segmentation Mode

Using the Segmentation function will apply a Color Map Filter to the MRI, where the Content of the MRI is based on intensity value and presented with Colors to show where potential areas of contrast. The areas of High Intensity are considered to be associate with abnormalities and can be displayed by warm tones (reds/yellows).

Areas of least intensity are displayed by cool tones. This function allows for the rapid identification of potential areas of Abnormality and is helpful for Manual/Semi-Automatic Segmentation. The Segmentation function is most valuable when attempting to identify Hyper-Intensity on FLAIR or T2 MRI images.

This mode facilitates the rapid localization of abnormalities and aids in manual or semi-automatic segmentation. It is particularly useful for identifying hyperintense areas in FLAIR or T2 MRI sequences.

Code snippet:

```
if "segment" in view_mode:
    img_display = cv2.applyColorMap(
        cv2.cvtColor((img_gray * 255).astype(np.uint8), cv2.COLOR_GRAY2BGR),
        cv2.COLORMAP_JET)
    fig = px.imshow(img_display)
    fig.update_layout(coloraxis_showscale=False, dragmode='drawclosedpath',
        xaxis_showticklabels=False, yaxis_showticklabels=False)
    return fig, hist_fig, ""
```

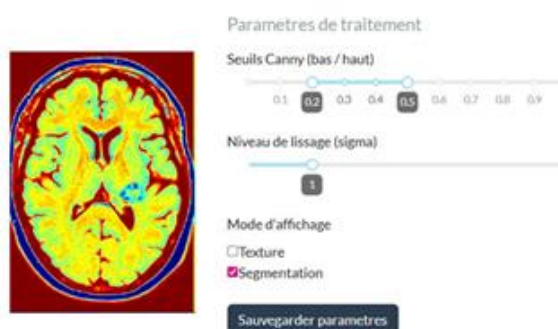


Figure 4: Segmentation Mode: Color-coded MRI display highlighting intensity variations. The tumor appears in blue, aiding localization.

As illustrated in the figure above, the tumor appears blue, clearly distinguishing it from the rest of the brain.

6-2 Texture Mode

This mode is useful for detecting structural abnormalities invisible to the naked eye because it is based not only on intensity but also on local variation. This allows for the differentiation of tissues with similar intensities but different textures.

The Texture mode creates a texture map for the image by calculating the variance of the intensity values found within an area of an image. I chose this mode because it highlights areas of irregular texture, which is particularly relevant given that tumors are typically more heterogeneous than surrounding normal tissue.

Moreover, it can also be a solution to identify any structural problems that cannot be visually detected, because of their intensity levels and the local variations within the image.

Code snippet:

```
if "texture" in view_mode:
```

```
    texture_map = local_texture_variance(img_gray, kernel_size=9)
```

```
    fig = px.imshow(texture_map, color_continuous_scale='Viridis',  
                    title="Texture map (local variance)")
```

```
    fig.update_layout(coloraxis_showscale=True,
```

```
                        xaxis_showticklabels=False,
```

```
                        yaxis_showticklabels=False)
```

```
    return fig, hist_fig, ""
```

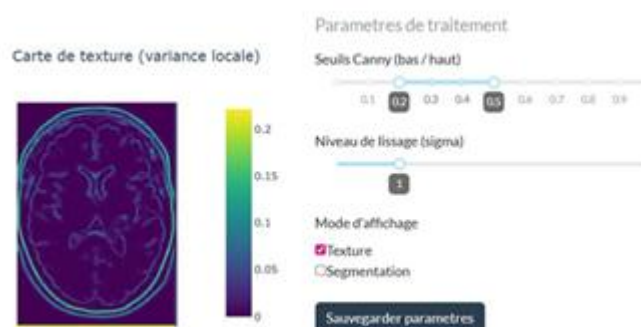


Figure5: Texture Mode Local variance map highlighting microstructural heterogeneity. Useful for detecting subtle tissue abnormalities.

The texture map highlights contours and heterogeneous areas in shades of blue and purple, with high yellow/green values indicating potentially pathological regions.

VII. HISTOGRAM ANALYSIS AND PEAK DETECTION

A histogram is a graphical representation of the distribution of the pixel intensity values in an image or an area of interest (ROI). The histogram contains one bar for every possible intensity value between 0 and 255 for a grayscale image that indicates the number of pixels within that intensity range. This representation is a key indicator of the make-up of the tissue in the area of interest.

The detection of peaks in the histogram allows us to know the main intensity values, or those that are represented by the highest bar counts, within the histogram. These peaks typically represent the relative composition of tissue types (gray matter, white matter, cerebrospinal fluid) including the presence of abnormal tissue types for instance, tumors.

Interpreting the histogram and the location of detected peaks

The histogram shown is a representation of the intensity distribution of the pixels associated with the selected area of the MRI image shown in the previous figure. Three distinct peaks can be found on that histogram, which are located approximately at 127, 168, and 188, and correspond to the dominant grey levels present in the area of interest.



Figure 6: Intensity Histogram and ROI Grayscale MRI with highlighted lesion (red) and corresponding intensity histogram. Peaks indicate tissue contrast.

The histogram shows the following peaks: the first peak at approximately 127 is indicative of high concentrations of gray matter, the second peak at about 168, and the third peak at 188, are representative of higher-intensity pixels typically seen in hyper intense regions found on FLAIR or T2-weighted imaging, and these hyper intense areas are associated with pathological conditions, for example a tumor, edema.

Furthermore, multiple peaks suggest that the selected area is heterogeneous, it contains both normal and abnormal. Knowledge of this will aid in determining thresholds for segmentation, thus allowing isolation of the lesion. For example, a threshold of 160-190 could be used to extract the tumor from surrounding tissue.

Thus, evaluating the histogram can provide more than a visual representation; it can serve as a quantitative reference of tissue composition and the complexity of the lesion(s). With this data, the following objectives may be achieved:

Adjust segmentation parameters (e.g., 'Canny' thresholds or binarization).

Characterize the tumor based on its intensity.

Use intensity distributions to develop automated classification methods.

```
from scipy.signal import find_peaks
```

```
import numpy as np
```

```
import plotly.graph_objects as go
```

```
# Intensities of the selected area
```

```
intensities = king[mask == 255].flatten()
```

```
# Histogram calculation
```

```
counts, bins = np.histogram(intensities, bins=50)
```

```
hist_fig = go.Figure()
```

```
hist_fig.add_trace(go.Histogram(x=intensities, nbinsx=50, marker_color='blue', name="Intensities"))
```

```
# Peak detection
```

```
peaks, _ = find_peaks(counts, prominence=10) # 'prominence' to avoid false peaks
```

```
# Peak annotation
```

```
for p in peaks:
```

```
    hist_fig.add_trace(go.Scatter( x=[bins[p]], y=[counts[p]], mode='markers+text', text=[f'pic: {bins[p]}'], textposition="top center", marker=dict(color='red', size=10), name=f'trace {p} ))
```

```
    hist_fig.update_layout(title="Intensity Histogram with Peak Detection",
```

```
    xaxis_title="Intensity", yaxis_title="Frequency")
```

VIII. LIMITATIONS AND DIFFICULTIES

Although each method has its strengths, they also present limitations. Histogram Analysis provides information about pixel intensity distribution but ignores spatial relationships, which can lead to misleading interpretations. Semi-Automatic Segmentation depends on user input, reducing reproducibility between users. Gaussian Smoothing effectively reduces noise but may blur fine structures and decrease contour detection accuracy, particularly for small lesions. Similarly, texture- and segmentation-based analyses are sensitive to parameter selection and image artefacts. Overall, the need for careful adjustment of parameters: Canny thresholds, sigma, and window size, limits automation and clinical applicability. Future improvements should focus on automatic segmentation, Deep Learning approaches, and validation on larger datasets to ensure reliability and clinical use.

IX. CONCLUSION AND PERSPECTIVES

The approach I took in this work demonstrated the need of employing a variety of methods for analyzing brain MRI data, ranging from semi-automatic segmentation to histogram analysis with peak detection. We were able to better classify lesions and more because to the intuitive visualization and quantitative data it provided. Additionally, texture maps enhance the research by highlighting structural anomalies that are unseen to the unaided eye, while the histogram and peak detection offer us a helpful analytical dimension for modifying thresholds and recognizing various tissues.

REFERENCES

- [1] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J.S. Kirby, et al., "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Nature Scientific Data*, 4:170117 (2017) DOI: 10.1038/sdata.2017.117
- [2] Togaçar et al : "BrainMRNet : Brain tumor detection using magnetic resonance images with a novel convolutional neural network model" 2019.
- [3] Fachrurrozi1, M et al "Improving the performance for automated brain tumor classification on magnetic resonance imaging deep learningbased", *IAES International Journal of Artificial Intelligence (IJ-AI)* Vol. 13, No. 2, June 2024, pp. 1679~1686 ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i2.pp1679-1686
- [4] Sun, Y., Wang, L., Li, G. et al. A foundation model for enhancing magnetic resonance images and downstream segmentation, registration and diagnostic tasks. *Nat. Biomed. Eng* (2024). <https://doi.org/10.1038/s41551-024-01283-7>

AUTHORS' INFORMATION



GHITA GHOULI, *Polytech Lyon engineering school, specializing in Biomedical Engineering.*

ghita.ghouli15@gmail.com

Fourth-year student at Polytech Lyon, specializing in Biomedical Engineering at Claude Bernard University Lyon 1. My interest includes medical imaging, image processing, and the development of innovative biomedical technologies to treat patients. She is particularly interested in applying engineering and digital tools to improve healthcare, diagnosis, and medical research.