Brain MRI Analysis
a Signal, Image and Video Project

Pietro Bologna, Christian Sassi

University of Trento
Department of Information Engineering and Computer Science

Abstract

Aside from the segmentation aspect, the algorithm employed in this project is designed to generalize well even when applied to images not included in the dataset. However, its flexibility is somewhat limited compared to algorithms that utilize more advanced techniques, such as machine learning algorithms. Nonetheless, it can closely resemble real medical segmentations.

I. Introduction

The goal of our project is to create a program that can analyze images from an MRI dataset. Given the nature of the project, it was decided to use only image processing algorithms without using more advanced and complex algorithms such as those that make use of machine learning. On these MRI images, the so-called segmentation is applied, the general goal of which is to identify and separate the different portions present in an image, such as objects, contours or regions of interest. This process is critical in many applications, including object recognition, medical image analysis, computer vision, and other areas where it is necessary to understand the structure and composition of an image.

The image processing techniques used in this project are the following:

1. **Image adjustment**
   The initial brain image undergoes adjustment, including denoising, histogram equalization, and setting the background to black.
2. **Skull removal**
   In this phase, the skull is removed and this is achieved by analyzing each area of the images and retaining the larger ones.
3. **Brain adjustment**
   In this phase, all objects remaining in the image are examined to determine if they could be part of the brain.
4. **Segmentation and Classification**
   In this phase, the final image undergoes segmentation using the K-means algorithm. Subsequently, a very basic classification approach is employed: each segment's area (colored pixels over black pixels) is utilized to differentiate regions.

After this first part, the various processed images are used to create 3D models of the brain. At this stage, the individual segmented images are overlaid to make a **3D graph** that is then saved as an HTML file. The entire project was developed using the Python programming language, mainly using the libraries: OpenCV, Scikit, NumPy and Plotly. The entire software can be accessed in the specific GitHub repository.

The following sections will delve deeper into the basis of this project, followed by a step-by-step analysis of the source code to clarify the approach used.

II. Segmentation

What is segmentation

Image segmentation is a technique that partitions a digital image into multiple segments where each of them shares common characteristics or computed properties. The primary aim of segmentation is to extract meaningful information from the image, facilitating more focused analysis and interpretation. [1]

Segmentation for medicine

Segmentation, a fundamental technique in medical image processing, plays a pivotal role in various aspects of healthcare, ranging from diagnosis to treatment planning. By partitioning medical images into distinct regions or segments based on their characteristics such as intensity, texture, or shape, segmentation facilitates the extraction of crucial anatomical structures and pathological lesions.
Furthermore, segmentation serves as a cornerstone for advanced medical imaging modalities like magnetic resonance imaging (MRI). In the context of brain imaging, segmentation plays a crucial role in accurately identifying key regions of the brains, as demon-

strated in this project.

Segmentation algorithm: K-Means

K-means is a popular unsupervised machine learning algorithm for clustering data points. Specifically, it partitions n observations (or data points) into k clusters by assigning each point to the cluster with the nearest centroid (cluster center), by using different approaches such as Euclidean distance. After the initial assignment, the centroids are recalculated and the entire process is repeated until there is no significant change in the position of the centroids. This indicates that the elements within each cluster have stabilized. [2]

Algorithm 1 K-Means

```
choose k as the number of clusters
randomly choose k datapoints as centroids
repeat
    for each datapoint do
        assign point to closest centroid
        recalculate centroid
    end for
until convergence criteria is met
```

The pseudocode presented in the Algorithm 1 demonstrates a basic implementation of the K-Means algorithm.

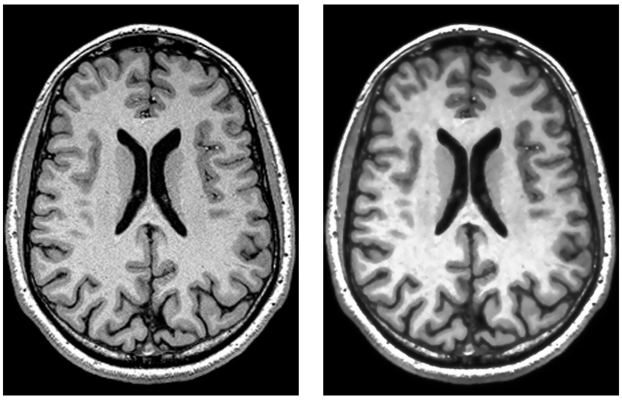
III. How it works

In this project, the dataset consists entirely of MRI brain scans, where the algorithm has been tailored specifically for axial images. This decision was made because segmentation performed on axial images tends to yield more accurate results and be more appreciable compared to segmentation performed on images captured from different perspectives of the brain.
However, not all of the axial images are used. Specifically, some of the topmost images were excluded because they only capture a small portion of the brain where segmentation would not yield meaningful results. Furthermore, certain lowermost images were excluded due to the presence of other organs besides the brain, such as eyes. Moreover, attempting segmentation on these images using traditional methods would be highly challenging and prone to significant errors.

Before proceeding, it is essential to preprocess the image to ensure that the segmentation function receives the correct input.

Image adjustment

The first step involves properly adjusting the image by denoising it, setting the background to black, and performing histogram equalization.
The background is set to black because some images may have a dark background that is not entirely black. This could negatively impact the next steps.
By denoising the image and applying histogram equalization, the colors within the image can be made more uniform. This aids in the segmentation process by facilitating better clustering of similar shades of color or distinguishing between two similar colors in separate segments.
These minor adjustments can ultimately have an overall positive impact on the segmentation process. Therefore, it is imperative to work with an image that is as clear as possible, minimizing imperfections or impurities.



(a) *Original image (input).* **(b)** *Adjusted image (output).*

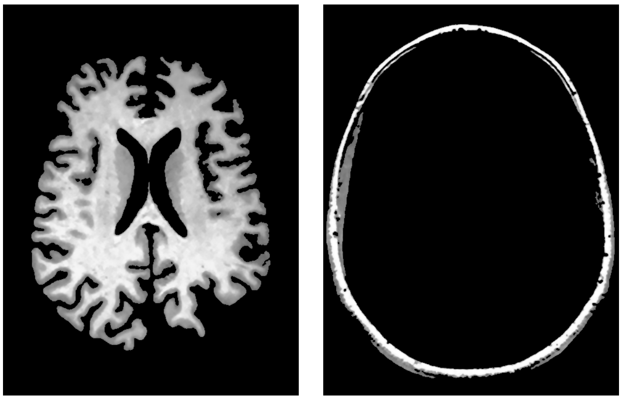
Figure 1: *Result obtained by using the **adjust_image** function of the **process** module.*

As showed in Figure 1b, the input image (Figure 1a) undergoes significant enhancement in quality, allowing a better and smoother segmentation.

Skull removal

In this phase, the primary objective is to eliminate the skull (if present) by delineating all the contours in the image. Initially, an appropriate threshold is applied to the image to facilitate contour extraction and, subsequently, each of them is extracted individually. These contours are then arranged in a list according to their respective areas. The larger contour, which is more likely to represent the skull, is discarded at this stage. Following this, the remaining contours undergo another process, whereby only those with areas exceeding a certain threshold are retained. This ensures the exclusion of smaller objects and the preservation of larger ones, which are presumed to represent the brain tissue.
This approach involves starting with a basic extraction of the brain, even if it misses some details.

As shown in Figure 2, the brain (Figure 2a) is sep-



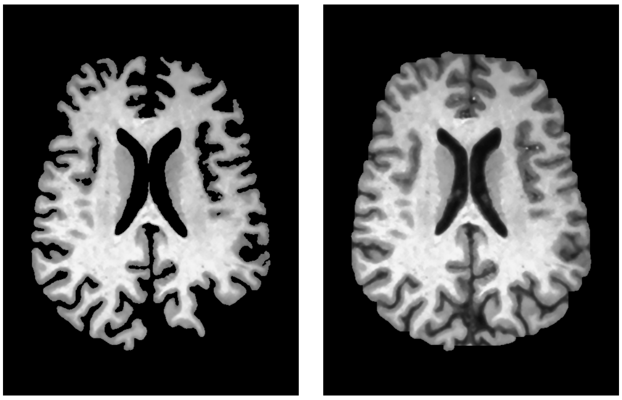
(a) *Extracted brain.* **(b)** *Extracted skull.*

Figure 2: *Result obtained by using the* ***remove_skull*** *function of the* ***process*** *module.*

arated from the skull (Figure 2b). However, it is evident that the extracted brain lacks many details, essential for a comprehensive and accurate segmentation. Therefore, it becomes imperative to refine the brain extraction for improved accuracy of the segmentation.

Brain adjustment

In this phase, following the processing of the larger brain components, attention turns to the smaller ones. These smaller segments may hold crucial details previously overlooked. Thus, the primary focus in this phase is on identifying and capturing those missed details to reconstruct the brain accurately. It is important to note that during this phase, not all remaining objects necessarily belong to the brain tissue. Some may be associated with other regions of the image that should not be considered for future segmentation.



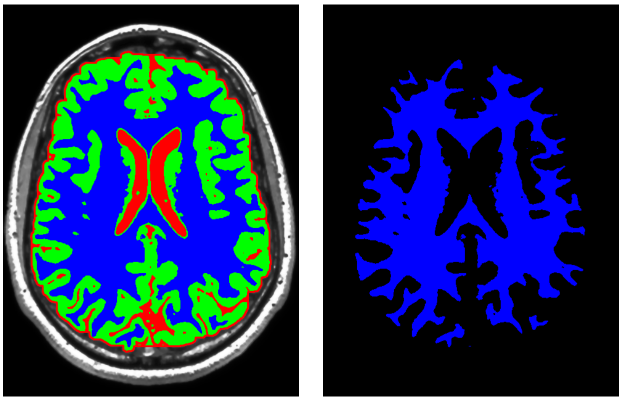
(a) *Previously extracted brain.* **(b)** *Adjusted brain.*

Figure 3: *Result obtained by using the* ***adjust_brain*** *function of the* ***process*** *module.*

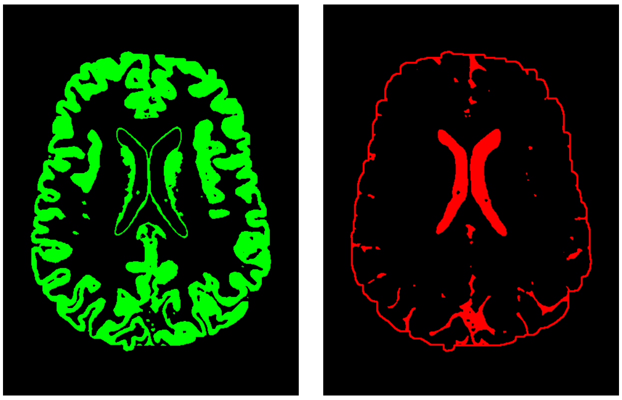
As depicted in Figure 3, significant improvements are observed in the brain image (Figure 3b) compared to Figure 3a. The process begins by subtracting the raw brain (Figure 2a) and the skull (Figure 2b) from the original image (Figure 1b), leaving only the interior of the head. Next, it is crucial to filter out objects belonging to the brain and exclude those from other regions unaffected by segmentation. Initially, the area of the raw brain is expanded, encompassing all its extreme pixels to create a more squared contour, contrasting with the sharper one generated by the OpenCV library function. Subsequently, objects are deemed to belong to the brain only if they intersect with the brain's contour or, alternatively, if they are inside it. Finally, the last step involves tracing the final contour of the brain and use it as a mask on the original image, resulting in the adjusted brain image shown in Figure 3b.
However, it is important to highlight that some images of the dataset are more challenging to be processed.

Segmentation and Classification

In this final phase, the objective is to efficiently segment and classify the processed image.



(a) *Final segmented image.* **(b)** *White Matter.*



(c) *Grey Matter.* **(d)** *Cerebrospinal fluid*

Figure 4: *Segmentation results.*

As mentioned earlier, the entire segmentation process relies on the K-Means algorithm (Algorithm 1). During segmentation, the output from K-Means consists of the k extracted colors. Subsequently, each

color is used as a mask on the original image to extract different segments of the image.
Regarding the classification aspect, the approach is straightforward. To efficiently classify the segments, the algorithm examines the number of colored pixels in each segment. Typically, the three different segments will have a relatively consistent number of colored pixels. Based on this observation, it is reasonable to associate the lowest value with the cerebrospinal fluid (Figure 4d), followed by the grey matter (Figure 4c), and, finally, the white matter (Figure 4b).

As demonstrated in Figure 4a, the selected algorithm successfully segments the provided image. It is worth noting that the displayed figure is just one example, but similar results were observed for all images analyzed in the dataset. Furthermore, promising results were also achieved on images that were not part of the dataset.

Plot 3d graph

After the various image processing steps, the resulting images are used to create 3D graphs that allow visualization of the reconstructed 360-degree brains. For example, from the skull removal stage one can reconstruct the complete brain by superimposing the various slices, as it can be seen in Figure 5. Otherwise, thanks to the classification stage, it is possible to reconstruct the segmented brain in which one can distinguish the three areas that make up the brain (cerebrospinal fluid, gray matter and white matter) as showed in Figure 6.

To make these 3D plots, a special function that uses the Plotly library was developed to incorporate the various images into a 3D plot.
First, image resizing is done to achieve a smooth and detailed 3D plot, although this resulted in a slight loss of quality.
After that, the four corners of the image in three-dimensional space are defined, creating a rectangular shape to represent the image in the graph.
Next, interpolation is used to obtain a smooth and detailed representation of the image in three-dimensional space. This helps to create a visually accurate 3D representation of the input image in the context of the three-dimensional graph.
Depending on the value of the *color* flag, the function adds a simple surface of a standard color, otherwise it extracts RGB channels from the image and adds a scatter diagram with colored dots.
Finally, the 3D graph is displayed on a web page where one can interact with it and have an all-round view of the brain. The generated 3D plot is also saved in an HTML file that can be opened at any time without having to re-run the entire code again.

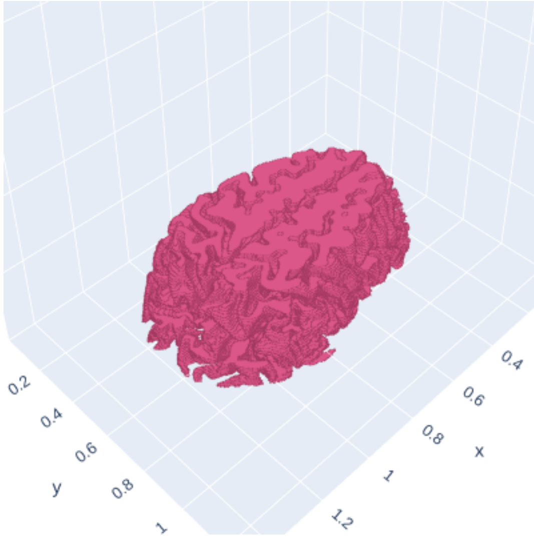


Figure 5: *3D removed skull.*

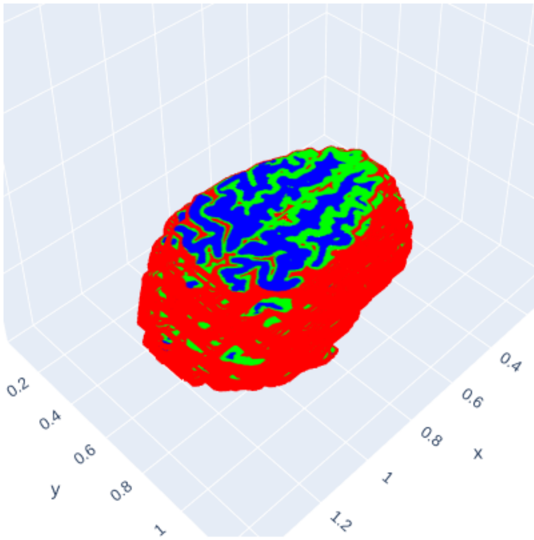


Figure 6: *3D segmented brain.*

IV. Results

As discussed previously, segmentation has emerged as a crucial process in the field of medicine.
The approach highlighted in this project relies solely on image processing algorithms, with no usage of advanced techniques such as machine learning, except for a simple application of K-Means in the segmentation process.
However, it is important to note that due to the absence of machine learning algorithms capable of generalizing across varying inputs, the performance of the proposed solution may vary when presented with dissimilar images. Nonetheless, efforts have been made to enhance generalizability, albeit it may not match the efficacy of previously discussed algorithms.

Nevertheless, the results obtained demonstrate promising outcomes, similar to real medical results.

IV. Future work

In the context of MRI image segmentation, a more complex but at the same time more accurate approach might be to use machine learning algorithms. For example, one possible work could be to train a model so as to automate the identification of different anatomical structures, such as brain fluid, gray matter, white matter, etc. This could be useful to automate the segmentation process and have greater accuracy in classification.
Also in the area of machine learning and deep learning, it would be very useful to implement an algorithm that can recognize even tumor areas. This would be very useful and, in the medical field, would allow one to prevent the continued enlargement of the tumor mass and intervene as early as possible.
Additionally, integrating multi-modal information could enhance segmentation accuracy. In fact, by incorporating data from different MRI sequences, such as T1-weighted, T2-weighted, and diffusion-weighted imaging, the model could gain a more comprehensive understanding of the brain's anatomical structures and pathological features.

So, by exploring these advanced techniques and incorporating them into the MRI image segmentation pipeline, one can strive for more accurate, efficient, and clinically relevant segmentation solutions for various neurological conditions.

References

[1] *Image segmentation*. `https://en.wikipedia.org/wiki/Image_segmentation`.

[2] *K-Means clustering*. `https://en.wikipedia.org/wiki/K-means_clustering`.