Detecting contours of human organs in CT images using the Canny edge detector

Abstract

This project explores the application of the Canny edge detector for detecting contours of human organs in CT images.

1 Introduction

Medical imaging plays a crucial role in diagnosing and understanding various conditions. The study focuses on enhancing organ boundary detection in CT images for improved medical analysis.

2 Methods

2.1 Data

The dataset used was obtained from the Computed Tomography-Magnetic Resonance Imaging Database (CTMRI DB). The algorithm was tested on images provided, additionally, two patients from the CTMRI DB were selected, each with approximately five records, including multiple images per record.

2.2 Gaussian Smoothing

To reduce noise and highlight important features, Gaussian smoothing was applied to the CT images.

The `gauss` function performs Gaussian smoothing on an input image using a 2D convolution with a Gaussian kernel. This process helps reduce noise and blur the image, making it smoother (2b).

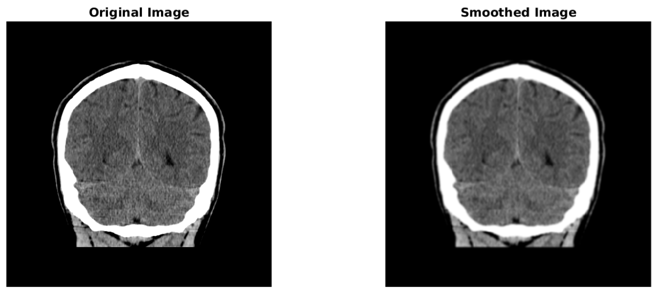
The Gaussian kernel is generated based on the specified standard deviation (σ) and the size of the kernel (s).

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

The kernel is then normalized by dividing it by the sum of all its elements to ensure that the weights add up to 1. The convolution is performed with the original image using this Gaussian kernel, and the result is a smoothed image.

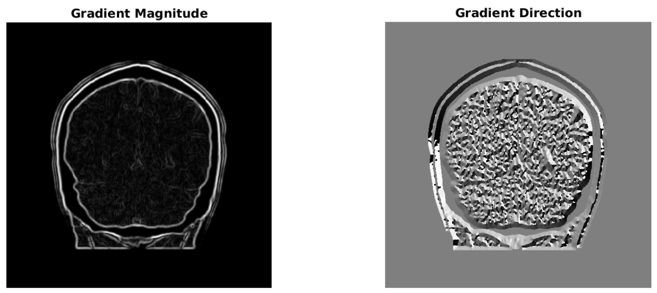
2.3 Gradient Magnitude and Direction

The gradient magnitude and direction of the smoothed images were calculated using various filters, including the Prewitt, Sobel, and Roberts operators, to identify potential edges.



(a) (b)

Figure 1: Original and Smoothed Image



(a) (b)

Figure 2: Gradient magnitude and direction

The magnitude (mag) of the gradient at each pixel was then computed using the Euclidean norm:

$$mag = \sqrt{G_x^2 + G_y^2}$$

The direction (dir) of the gradient was determined using the arctangent function:

$$dir = \arctan\left(\frac{G_y}{G_x}\right)$$

These calculations help highlight areas with significant changes in intensity, which are indicative of potential edges (2). The resulting gradient magnitude and direction maps were used in subsequent steps of the Canny edge detection algorithm.

2.4 Non-maximum Suppression

To enhance the precision of detected edges, a non-maximum suppression technique was applied. The purpose of this step is to thin out the detected edges, retaining only the most prominent features.

The algorithm operates by examining each pixel's gradient magnitude and its orientation. For each pixel, the method compares its magnitude with the neighboring pixels along the gradient direction. The pixel retains its magnitude only if it is greater than its neighbors along the gradient direction, otherwise, it is suppressed to zero.

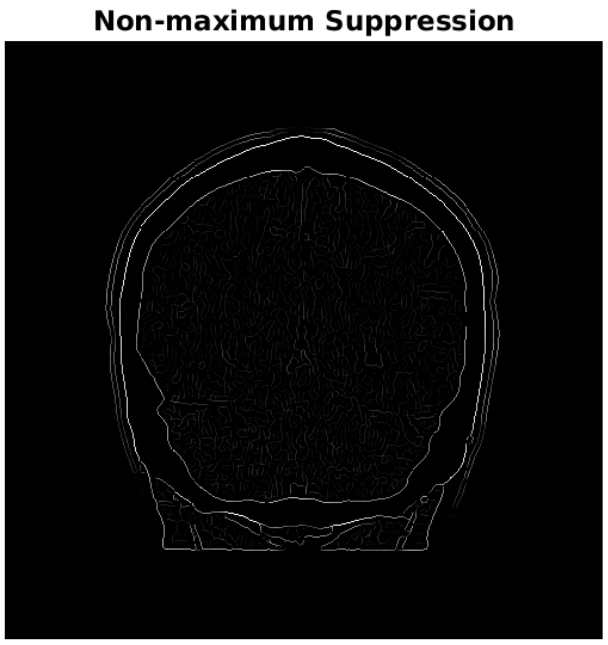


Figure 3

This process ensures that only local maxima in the gradient direction are preserved, contributing to a more refined edge map (3).

2.5 Edge Tracking by Hysteresis

Hysteresis-based edge tracking was implemented to connect edge segments into complete contours, improving overall detection accuracy.

This function ensures that strong edge candidates are connected, while weaker ones are suppressed, contributing to the final edge detection (4).

2.6 Parameter optimization

The parameter optimization process aims to identify the optimal set of parameters for the Canny edge detection algorithm by automatically exploring different combinations. The following parameters were considered:

1. **Sigma (σ):**
   - **Range:** [0.5, 1, 1.5]

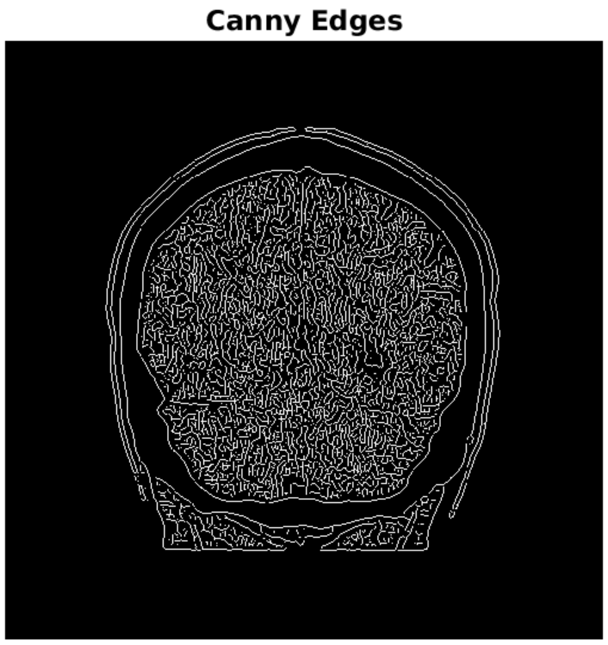
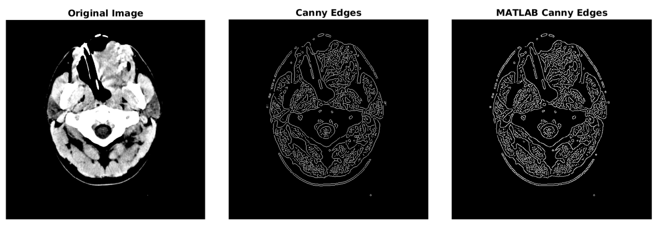


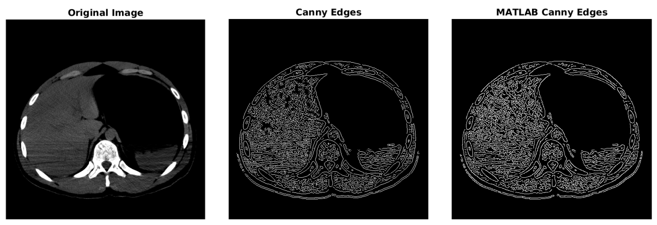
Figure 4: Hysteresis

   - Smaller sigma values result in less smoothing, while larger sigma values lead to increased smoothing. The optimal sigma depends on the characteristics of the input image.
2. **Filter Size:**
   - **Range:** [3, 5, 7]
   - The filter size determines the size of the convolution kernel used for Gaussian smoothing. Larger filter sizes result in stronger smoothing. The optimal filter size depends on the scale of the structures present in the image.
3. **Hysteresis Thresholds:**
   - **Low Threshold (*lowThresholdRatio*):** [0.1, 0.2, 0.3]
   - **High Threshold (*highThresholdRatio*):** [0.3, 0.35, 0.4]
   - These two thresholds to distinguish between strong and weak edges. The low threshold captures potential weak edges, while the high threshold is used to identify strong edges.
4. **Neighborhood Size:**
   - **Range:** [3, 5, 7]
   - The neighborhood size is for determining the connectivity of edges. A larger neighborhood considers more pixels in deciding whether to connect an edge. The optimal size depends on the expected width and continuity of edges in the image.
5. **Kernels:**
   - **Options:** Roberts, Prewitt, Sobel
   - Different convolution kernels were considered for calculating image gradients.



(a) (b) (c)

Figure 5



(a) (b) (c)

Figure 6

The optimization process involves iterating over all possible combinations of these parameters and selecting the set that maximizes the edge quality metric. The edge quality is evaluated using the F1 score, which considers precision and recall. The optimization aims to find parameters that result in edge maps similar to the edges detected by the MATLAB Canny edge detection function, as that function was set as the ground truth of the evaluation. The best parameter set is chosen based on the highest F1 score.

After a parameter search, the optimal set of parameters were chosen empirically for the Canny edge detection algorithm:

- **Sigma (σ):** 1.5
- **Filter Size:** 7
- **Kernel:** Sobel
- **Hysteresis Thresholds:** 0.2 (Low), 0.35 (High)
- **Neighborhood Size:** 5

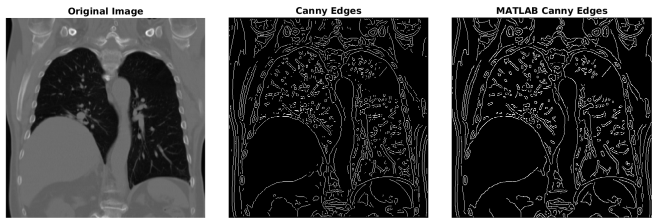
3 Results

The algorithm successfully detected organ contours in CT images. The results are shown in order: original image, implemented canny edge detection algorithm and MATLAB canny edge detection algorithm (for comparing).

4 Discussion

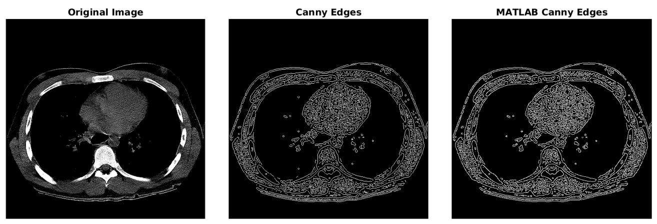
The implemented Canny edge detection algorithm demonstrated performance comparable to the MATLAB Canny edge detection function. The similarity in results indicates the effectiveness of the implemented algorithm in detecting organ contours in CT images.

Further improvements could involve exploring advanced techniques for gradient calculation or considering machine



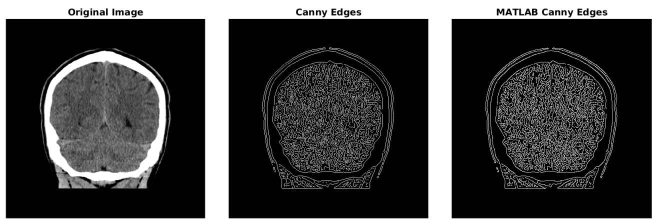
(a) (b) (c)

Figure 7



(a) (b) (c)

Figure 8



(a) (b) (c)

Figure 9

learning approaches to adaptively tune parameters based on specific image characteristics, as that was the only concern I had, that I had to adapt the parameters to each image. The goal would be to develop a more adaptive system that can intelligently tune parameters based on specific image characteristics, reducing the need for manual adjustments.

In conclusion, the current implementation provides a solid foundation, and future work could focus on making the algorithm more automated and adaptable to diverse datasets, thereby improving its usability in various medical imaging contexts.