SCC0251/0651/5830 + MAI5020 — Prof. Moacir Ponti

PAEs: Ana Letícia Vicente, Márcus Vinícius Costa, Melissa Motoki and Sherlon Almeida

Assignment 3:

Color & Segmentation & Morphology

Code the assignment by yourself. Ask if you need help. Plagiarism is not tolerated.

1 Introduction

In this assignment, you have to implement functions in order to process images to identify regions of interest (ROIs) using a heatmap. Read the instructions for each step. Use python with the numpy and imageio libraries.

Your program must have the following steps:

1. **Read** the parameters:
   a) Filename for the input image (I).
   b) Filename for the reference image (H).
   c) Technique indexes $i \in M = \{1, 2\}$;
2. **Implement** the techniques below:
   a) **index i = 1** - Erosion.
   b) **index i = 2** - Dilation.
   c) Adaptive Limiarization (Otsu);
3. **Process** the input images:
   a) Convert the input image in Grayscale.
   b) Apply Adaptive Limiarization (Otsu) to binarize the image.
   c) Create the binary mask by applying the sequence of Erosions and Dilations.
   d) Create the heatmap using the given implementation.
   e) Create the colored mask as a combination of the heatmap and the binary mask.
   f) Create the image (G) combining the image (I) with the colored mask.
   g) Compare the created image (G) with the reference image (H);

2 Step-by-step overview applied to this assignment

In this assignment, you will work with Chest X-Ray and Skin lesion images. The databases are the Kaggle's Community award-winning COVID-19 Radiography Database[1], which contains chest X-ray images from several public repositories. Besides, the Skin Cancer Lesion Database[2], a large collection of multi-source dermatoscopic of pigmented lesions.

The first step of this assignment involves converting the input RGB images into grayscale. Figure 1 presents the given images of both datasets, while Figure 2 contains the grayscale transformation.

After that, you can apply Otsu's adaptive limiarization to binarize the image; note that all images present the segmentation perspective provided by this method. In Figure 3, the useful information of the images is represented as 0's (black) in a binary mask. Besides, this method shows the region of interest (ROI) in each type of medical image (e.g., lungs and skin lesions).

Afterward, you must apply mathematical morphological operators (e.g., erosion and dilation) to mitigate the ROI's pixels with noise or edge information, as shown in Figure 4.

[1] `https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database`

[2] `https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000?select=HAM10000_images_part_1`

Figure 1: All images from the COVID-19 X-ray and Skin Lesion datasets in RGB mode.

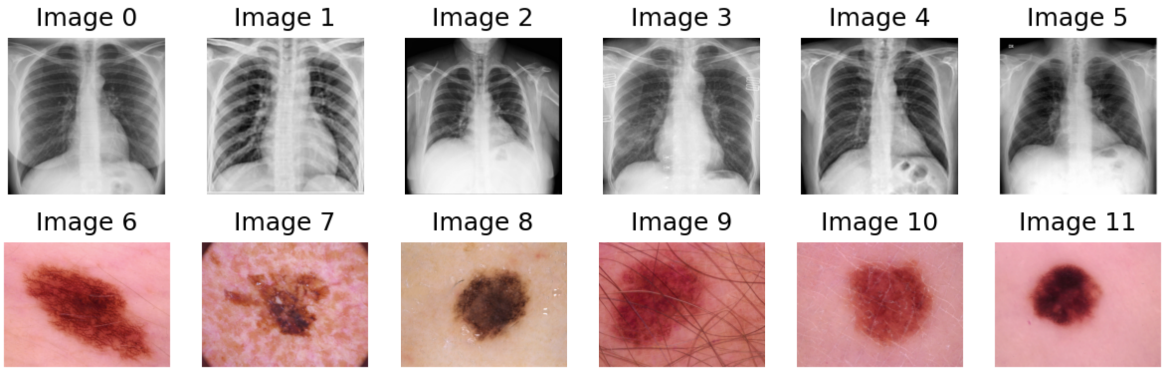


Figure 2: Applying a grayscale transformation to all images.

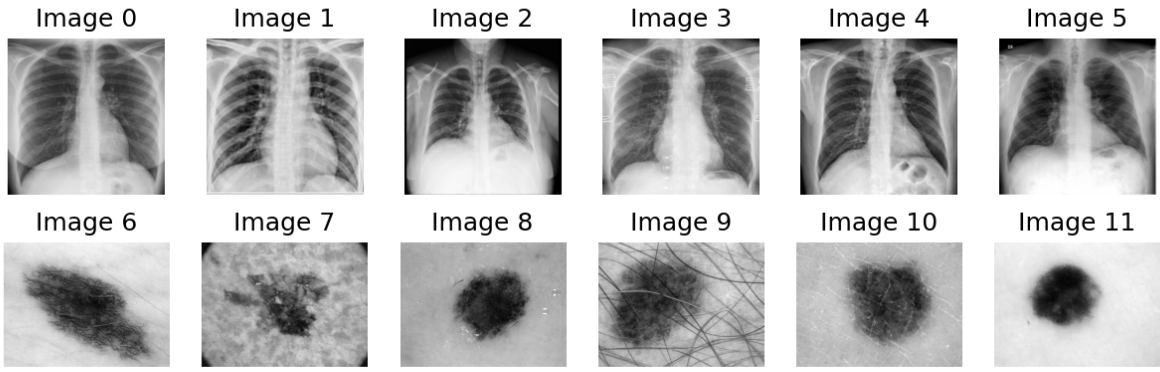


Figure 3: Applying OTSU threshold method.

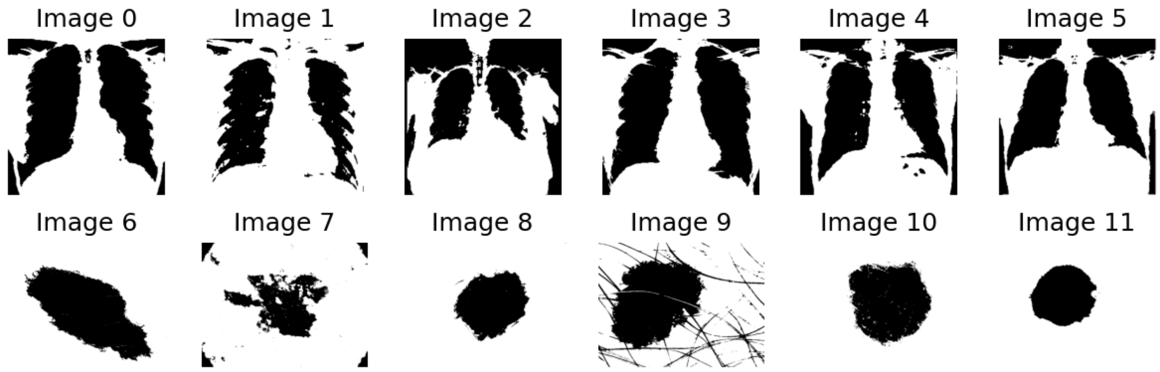
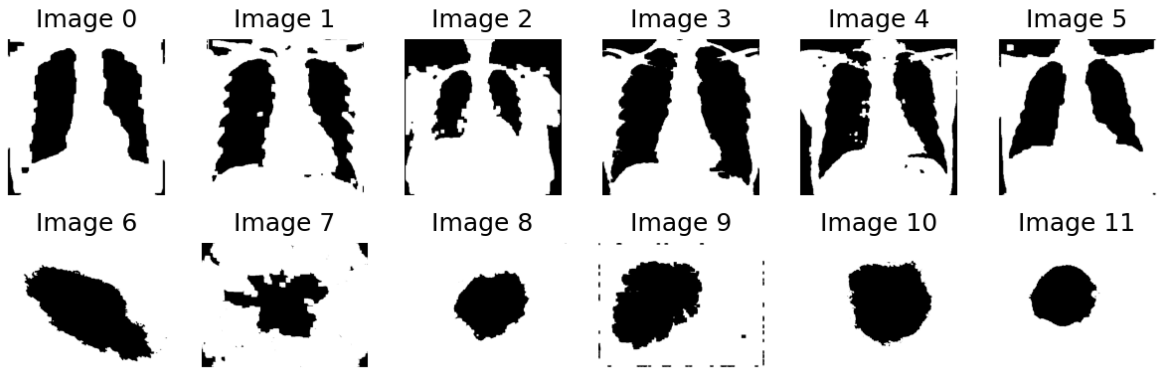
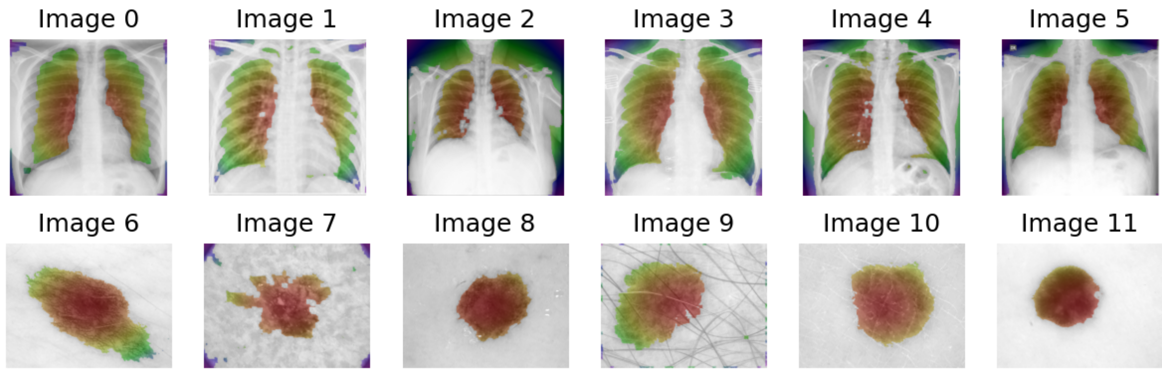


Figure 4: Applying mathematical morphological operators (e.g., Erosion and Dilation).



Finally, using the binarized image (e.g., mask), you will create a colored mask using a transformation of a Gaussian filter in an RGB color distribution that corresponds to the heatmap. After combining it with the input images, the output images shown in Figure 5 are obtained (more details in section 5).

Figure 5: Applying Color to Regions of Interest (ROI).



3 OTSU

Otsu's thresholding or Otsu's binarization, is a technique used in image processing for automatic image thresholding. The goal of thresholding is to separate objects or regions of interest from the background in an image by selecting an optimal threshold value. Otsu's method computes the optimal thresholding by maximizing the between-class variance of pixel intensities and minimizing the intra-class variance.

The intra-class variance with a threshold in L can be described as $\sigma^2 = W_a\sigma_a^2 + W_b\sigma_b^2$. The W_x represents the weight related to the class x, calculated as $W_x = \frac{number\_pixels\_class\_x}{number\_pixels}$. The method evaluates which threshold L is best for binarization by computing all possible thresholds for a specific image.

It's important to note that these thresholds span a closed interval from the minimum to the maximum pixel value in the original image. Additionally, the threshold L divides the image into two classes: one containing pixels with values less than L, and the other containing pixels with values greater than or equal to L[3].

4 MORPHOLOGY

In the field of digital image processing, morphological operations are crucial for shaping and refining structural features within an image. These operations are particularly valuable when preparing images for segmentation and advanced analysis, such as combining with heatmaps for medical imaging applications. This assignment focuses on two primary morphological operations: **erosion** and **dilation**. Both are based on the application of a structuring element to the binary image, which influences the output based on its shape and size.

Erosion and dilation are fundamental morphological operations used to manipulate the shapes within a binary image. Erosion reduces object sizes by sliding a structuring element over the image and setting the pixel to the foreground only if the structuring element completely fits within the foreground, otherwise, it sets it to the background. This operation effectively shrinks white regions and expands black ones by taking the minimum value of pixels beneath the structuring element.

Conversely, dilation expands object sizes by using the same structuring element to increase the size of the foreground regions. It sets the pixel to the foreground if any part of the structuring

[3]Reference: Otsu, N. (1975). A threshold selection method from gray-level histograms. Automatica, 11(285-296), 23-27.

element overlaps with the foreground, thus adopting the maximum value under the structuring element. This enlarges objects and fills in small gaps and holes in the foreground, enhancing feature visibility.

For complex image processing tasks such as the segmentation of medical images or detailed analysis of regions within an image, these morphological operations are often employed sequentially. The sequences typically involve:

- **Opening** is **an erosion followed by a dilation**. This sequence is useful for removing small objects from the foreground (typically noise), as the erosion removes the small objects and the dilation restores the shape of the larger objects.

- **Closing**, on the other hand, is **a dilation followed by an erosion**. This is useful for closing small holes inside the foreground objects, or small black points on the object.

In practice, each image undergoes a series of transformations dictated by its unique morphological cycle. For instance, an image may undergo multiple dilations followed by erosions, enhancing the mask's quality by reducing noise and defining clearer regions of interest (ROIs). This process is implemented in a loop where each image is processed according to its specific needs, ultimately leading to a set of images ready for further analysis and visualization.

In this assignment, you are tasked with applying these operations to refine the segmentation of images. After converting images to grayscale and applying a binarization technique (Otsu's method), you will employ a sequence of erosions and dilations (as shown in Figure 4). This sequence is designed to enhance the binary mask, which plays a critical role in creating a heatmap for visualizing regions of interest.

The choice and sequence of morphological operations can significantly impact the effectiveness of the heatmap overlay and the final visual output, which combines this heatmap with the original image to highlight areas of interest effectively.

5 COLOR

Color is a powerful descriptor that simplifies object detection and extraction of elements in a scene. Considering the color information present in our images in this assignment, you will implement the following step-by-step shown in Figure 6.

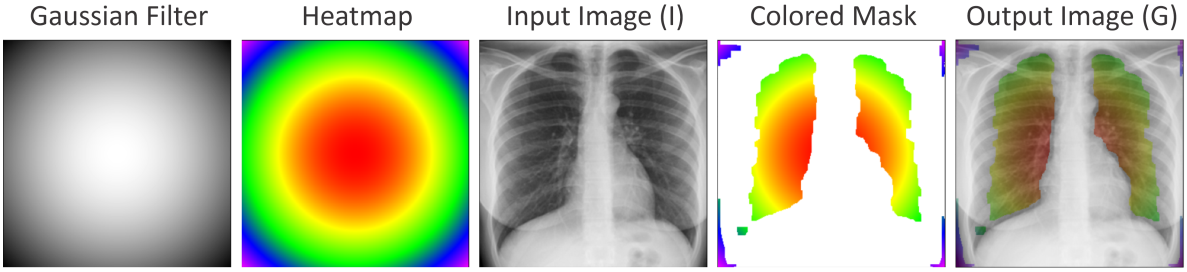
Firstly, you will create the Gaussian low-pass filter. The equations 1 and 2 represent this filter. The standard deviation, σ, determines the width of the Gaussian curve, with a smaller value of σ resulting in a sharper filter. For this assignment, you can consider the sigmas as the dimension $M \times N$ of the image.

$$x = (\frac{(u - \frac{P}{2})^2}{2\sigma_r^2} + \frac{(v - \frac{Q}{2})^2}{2\sigma_c^2}) \tag{1}$$

$$H(u, v) = e^{-x} \tag{2}$$

The filter is defined by the function $H(u, v)$, where u and v are the spatial frequency variables, and P and Q are the dimensions $M \times N$ of the image.

Figure 6: Step-by-Step to implement heatmap in this assignment.



The heatmap is defined by a transformation of each position of the Gaussian filter into an RGB value. You can use the code provided in the **Code Reference.ipynb** file to calculate the heatmap. After that, you can use the binary mask to obtain the ROI's positions and create the colored mask using the colors from the heatmap.

To create the output image (G), which combines the input image (I) and the generated colored mask, consider the Equation 3 with alpha equal to 0.30. In other words, you will consider 70% of the input image and 30% of the colored mask to create the output image (G).

$$img\_final = (1 - \alpha) \times img\_gray + \alpha \times mask\_color \tag{3}$$

6 Input and Output

The following parameters: Input image (I), reference image (H), and the sequence of (1) Erosions and (2) Dilations to be performed.

input	output

`chest_xray.png`
`chest_xray_ref.png`
`1 1 1 2 2 2 2`
	0.7079

7 Comparing against expected

Your program must compare the created image G against reference image H. This comparison must use the root mean squared error (RMSE).

$$RMSE = \sqrt{\frac{1}{m \times n} \sum_i \sum_j (H(i,j) - G(i,j))^2}$$

where $m \times n$ is the size of the image.

Considering that images are composed of 3 channels (R, G, B), you must compute the RMSE for each channel separately (as presented in the equation above) and calculate the average among them. For example:

$$RMSE_{color} = \frac{(RMSE_{red} + RMSE_{green} + RMSE_{blue})}{3}$$

8 Submission

Submit your source code to `https://runcodes.icmc.usp.br` (only the `.py` file).

1. **Use your USP number as the filename for your code.**
2. **Include a header**. Use a header with name, USP number, course code, year/semester and the title of the assignment. A penalty on the evaluation will be applied if your code is missing the header.
3. **Comment your code**. For any computation that is not obvious from function names and variables, add a comment explaining.
4. **Organize your code in programming functions**. Use one function for each technique.