Medical Imaging Course Final Assignment: DICOM Images Coregistration

Refo Ilmiya Akbar

MUSI - Master Universitario en Sistemas Inteligentes

Outline

- Introduction
- Data Description
- Implementation
- Results
- Conclusions

Introduction

Universitat de les Illes Balears

Objective 1 of the project

- Load, visualize, and perform coregistration of DICOM images
- Focus on the HCC-TACE-Seg dataset of a certain assigned patient
- Use PyDicom to load DICOM images and rearrange pixel arrays based on relevant headers
- Create an animation showcasing a rotating Maximum Intensity Projection on the coronal-sagittal planes with segmentation shown

Objective 2 of the project

- Implement coregistration of images using landmarks or a custom similarity measure without external libraries using a brain CT-scan dataset
- Visualize the Thalamus region in the input image space
- Contribute to efficient loading, visualization, and coregistration of DICOM images for enhanced understanding and analysis of medical imaging data

Data Description

The Dataset

- Objective 1:
  - HCC-TACE-Seg is a dataset of hepatocellular carcinoma (HCC) cases.
  - It contains multimodality annotated cases with and without advanced imaging segmentation.
- Objective 2:
  - Reference image: icbm avg 152 t1 tal nlin symmetric VI (a standardized template image of a brain)
  - Input image: RM Brain 3D-SPGR (3D spoiled gradient-recalled image)
  - Mask image: AAL3_1mm (Automated Anatomical Labelling atlas)

Implementation

Objective 1: The Implementation

Downloading and Visualization	Data analysis	Loading and Examining DCM files	Segmented Image Creation	Generating GIF file

Downloading and Visualization

- Download the assigned HCC 001 dataset using NBIA Data Retriever.

- Visualize the dataset using 3D Slicer to gain a better understanding of the data.

Data analysis

- Identify the specific acquisition for valid segmentation data.

- Determine the four segmentation sequences: liver, tumor, vessels, and aorta.

Loading and Examining DCM files

- Load the DCM files corresponding to the second acquisition along with their segmentation data.

[More on the next slide]

Segmented Image Creation

- Perform alpha-fusion iteratively along the first axis using an alpha value of 0.2 for the mask and 0.8 for the image to create segmented images.

[More on the next slide]

Generating GIF file

- Apply axial rotation to the segmented image. Use 48 projections with a 30-degree interval for rotation.

[More on the next slide]

Universitat de les Illes Balears

Loading and Examining DCM files

- Load the three-dimensional array of the DCM files.
- Examine the information from the segmentation dataset's DCM files to identify the corresponding slices in the array for each segment.
- Flip the segmented image on the second axis for correct orientation.
- Analyze the array indices to identify the tumor slices (range: 37-73).

Segmented Image Creation

- Perform alpha-fusion iteratively along the first axis using an alpha value of 0.2 for the mask and 0.8 for the image to create segmented images.
- Visualize the segmented images by plotting both the image and segmented image.
- Apply alpha value of 0.8 to the image for overlay effect on the CT scan.
- Conduct multiple visualizations using Median, Maximum Intensity Projection (MIP), and Average Intensity Projection (AIP) across the sagittal and coronal planes

Generating GIF file

- Apply axial rotation to the segmented image.
- Use 48 projections with a 30-degree interval for rotation.
- For each projection, rotate the image and segmented image by a certain degree and plot the MIP sagittal plane.
- Implement a loop over the projections and combine the resulting frames using ArtistAnimation.
- Generate a GIF file.

Objective 2: The Implementation

Loading and Pixel Array Extraction → **Visualizations and Preprocessing** → **Coregistration** → **Thalamus Region Masking**

Loading and Pixel Array Extraction

- Load the three images:
  - reference image
  - input image
  - segmentation image

Visualizations and Preprocessing

- Conduct horizontal, sagittal, and coronal visualizations to assess the effects of preprocessing

[More on the next slide]

Coregistration

- Implement the coregister landmarks() function to coregister the images through a rigid transformation using landmark points.

[More on the next slide]

Thalamus Region Masking

- Implement the 'getthalamusmask()' function to generate a mask by selecting the thalamus indexes.

[More on the next slide]

Visualizations and preprocesing

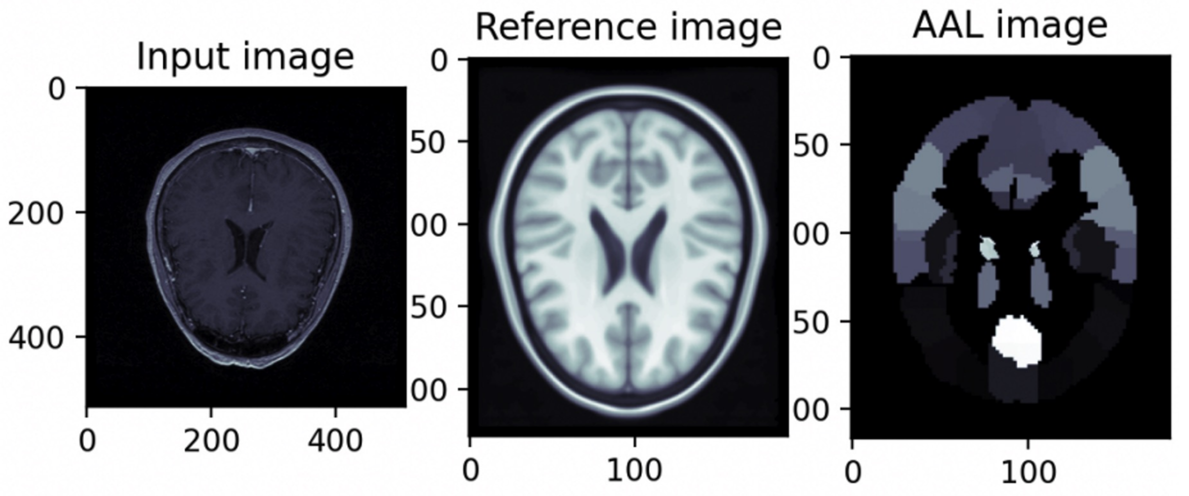


Figure 1: Initial visualization of the images

Universitat de les Illes Balears

Visualizations and preprocesing

- Preprocess the images to address differences:
  - Crop the reference image and segmentation image to match the size of (181, 216, 181).
  - Transform the input image through cropping, zooming, rotation, and intensity normalization to align it with the other images and achieve consistent intensity levels.
  - Normalize the input image to have a similar range of values as the reference image.

Visualizations and preprocesing

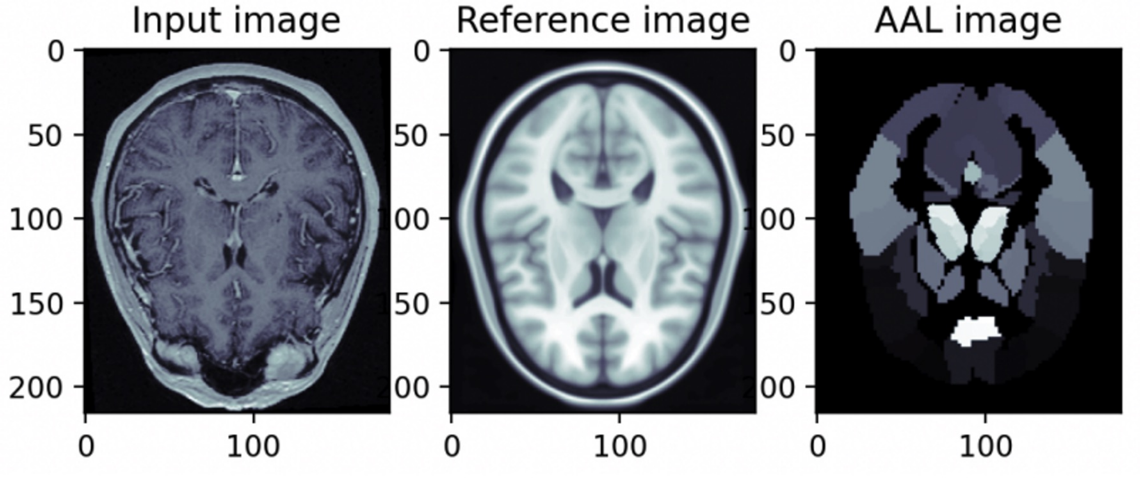


Figure 2: Preprocessed images

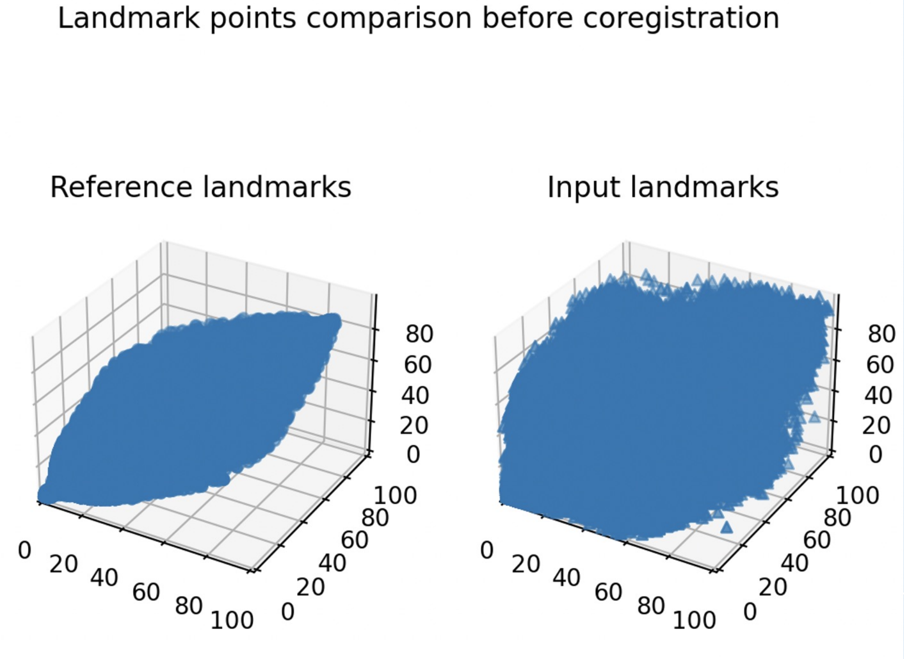
Coregistration

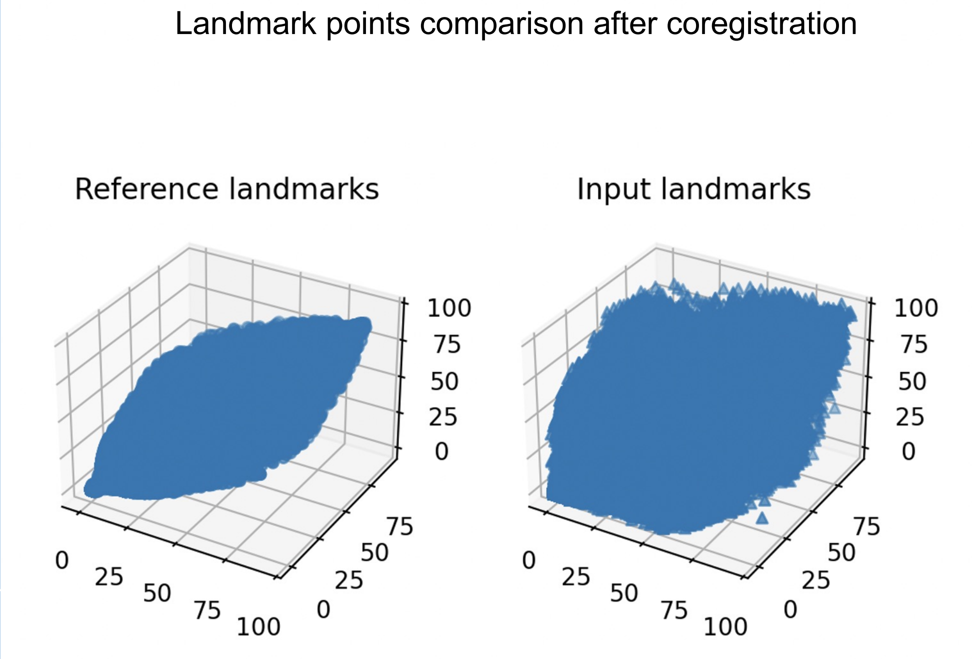
- Transform the reference and input images into 3-dimensional arrays using the reshape(-1,3) function
- Examine the residual vector between each point to observe disparities
- Perform a 3D visualization on each point of the 3D arrays for comparison
- Implement the coregister landmarks() function to coregister the images through a rigid transformation using landmark points
- Update initial parameters, compute centroids, and define a function to minimize residuals
- Optimize the transformation parameters using the least squares function to minimize residuals

Coregistration

- Create downsampled arrays of the input and reference images to reduce computation time
- Perform coregistration between the downscaled arrays to obtain optimal coregistration parameters
- Adjust the input image with the reference image using the translation then axialrotation() function
- Visualize the effects of the coregistration process (next slide)

Coregistration (Visualizations)

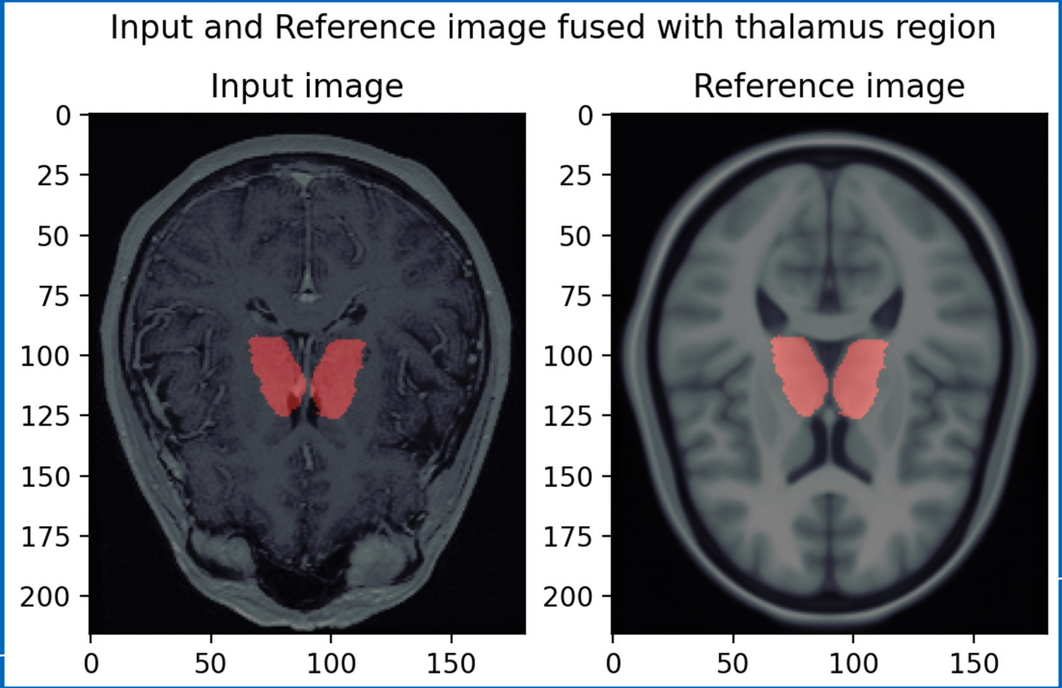




Thalamus Region Masking

- Implement the 'getthalamusmask()' function to generate a mask by selecting the thalamus indexes.
- Utilize the 'visualaxialslice()' function to perform alpha fusion between the thalamus region mask and the input image.
- Apply the same fusion to the reference image for verification.

Thalamus Region Masking (Visualization)

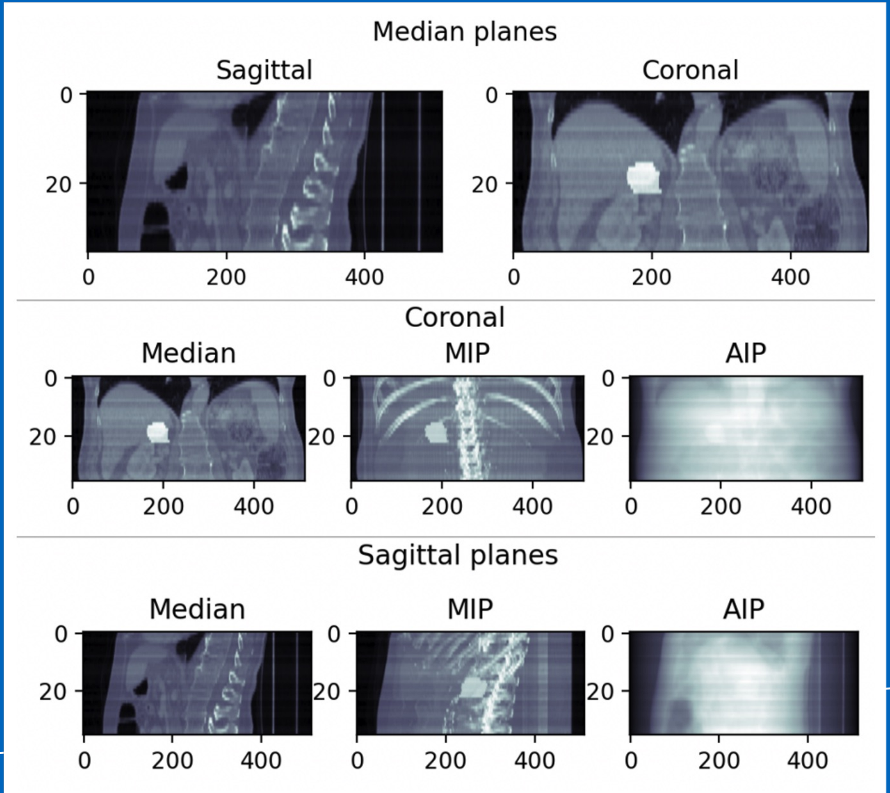


Thalamus Region Masking

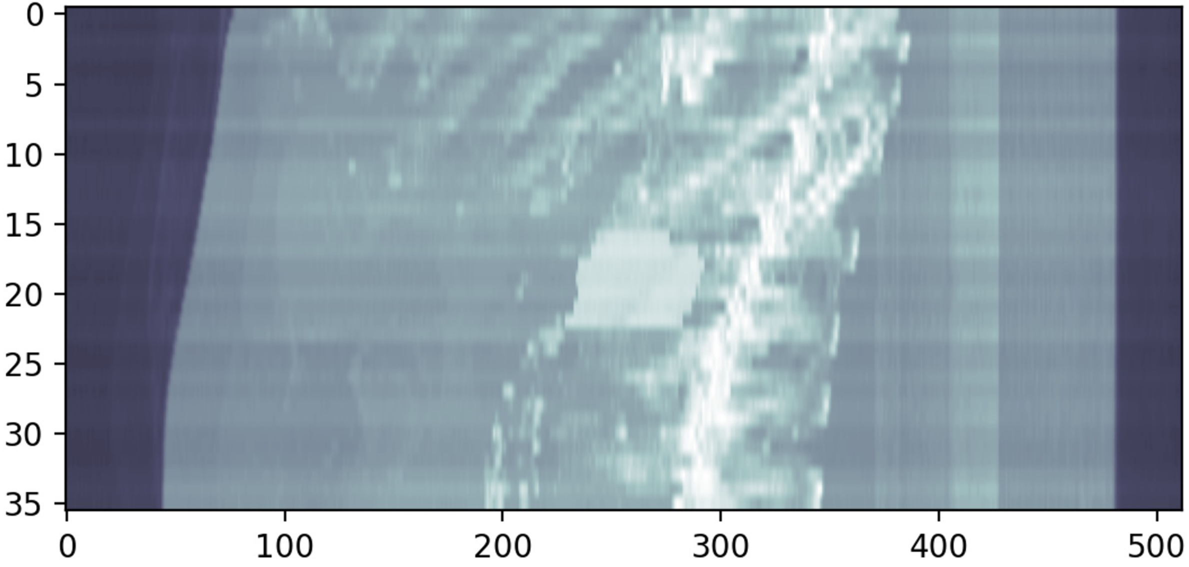
- Perform a reverse transformation to observe the masking in the input space.
- Slice the fused image to obtain a 2-dimensional array and match its intensity with the original image intensity after coregistration.
- Repeat the coregistration process with inverted values of the best parameters.
- Repeat the preprocessing steps with inverted values, including rotation, resizing, and zooming out, and add padding to restore the original size.
- Generate a visualization of the mask in the input space.

Results

Objective 1 results

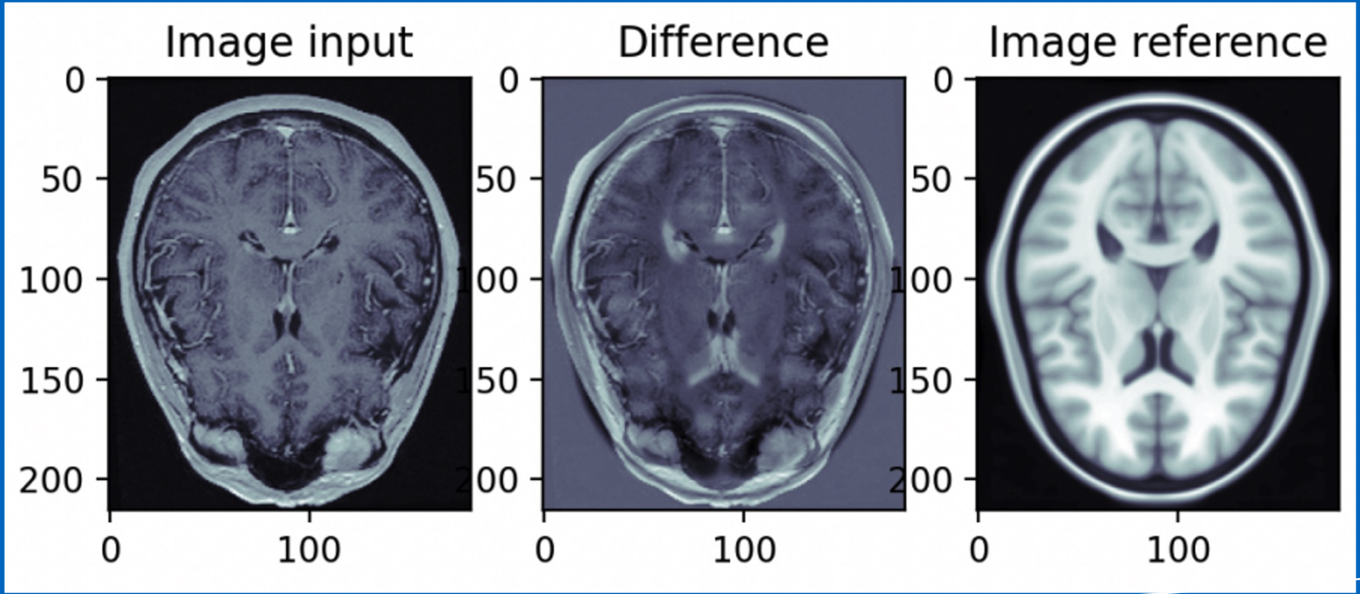


Objective 1 results



48 projections of sagittal-coronal planes with 30ms interval

Objective 2 results



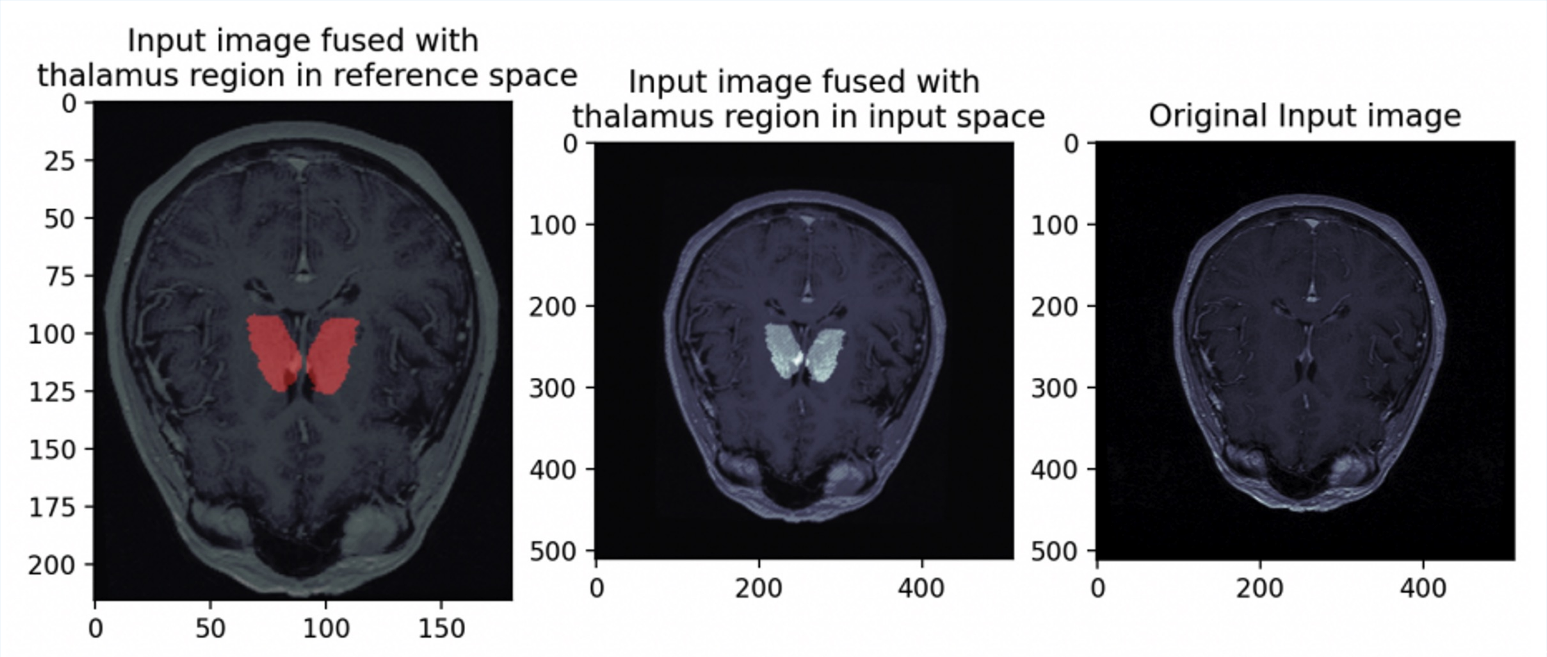
Universitat de les Illes Balears

Objective 2 results

Performance Measure	Value
MAE	13.31 HU
MSE	364.03 HU^2
MI	0.7 bits

Table 1: Performance of the coregistration algorithm

Objective 2 results



Universitat de les Illes Balears

Conclusions

Highlights of the Project

- **Valuable insights**: Gained insights into the complexities associated with working with DICOM data
- **Contribution to technical skills**: Enhanced technical skills in medical imaging techniques
- **Challenges**: Understanding DCM files, long computation time for coregistration, and initially unclear project objectives
- **Overcame challenges**: Through persistent efforts and learning from previous lab activities
- The project was **enjoyable** and engaging

Source code:

https://github.com/realr3fo/medical_imaging

Bibliography

- https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230229
- https://pydicom.github.io/pydicom/stable/auto examples/index.html
- https://mscipio.github.io/post/read dicom files in python/