Lung CT Volume Registration: Medical Image Registration and Applications project

Muhammad Kabir Hamzah
MSc. in Medical Imaging and Applications
University of Girona
Girona, Spain
u1992383@campus.udg.edu

Faisal Farhan
MSc. in Medical Imaging and Applications
University of Girona
Girona, Spain
u1992382@campus.udg.edu

Abstract

In this project, we carried out lung computed tomography (CT) image registration on a provided dataset utilizing both intensity-based and deep learning-based techniques. The training dataset comprised 4 pairs of volumetric scans, each accompanied by corresponding annotations (landmark points). The effectiveness of the registration algorithms was initially assessed using the Target Registration Error (TRE), defined as the 3D Euclidean distance between the transformed landmarks. This project demonstrated the impact of parameter optimization and segmentation masks in improving the accuracy of lung CT scan registration. By fine-tuning registration parameters and incorporating lung segmentation methods, a notable reduction in the mean TRE was achieved. The final optimized model achieved a mean TRE of 1.44 mm with a mean STD TRE of 1.77 mm and mean execution time of 250 seconds across all cases.

Keywords**:** *Image Registration, Lung CT, Elastix, Target Registration Error*

1. INTRODUCTION

Chest computed tomography (CT) is a diagnostic imaging method that uses X-rays to produce detailed visuals of the chest's internal organs and structures, including the lungs, heart, and blood vessels. CT scans are frequently utilized to detect and track various medical conditions, such as cancer, heart disorders, lung infections, and chest injuries. Image registration is a method used to achieve spatial alignment between two or more images on a point-by-point basis [1] [2]. It plays a critical role in medical image analysis and the creation of Computer-Aided Diagnosis systems. Over recent decades, extensive research efforts have focused on image registration, particularly within the realm of medical imaging [3] [4]. Researchers have introduced various techniques and customized approaches to address the unique challenges associated with this process. Chest CT registration is the process of aligning or “registering” two or more chest CT images taken at different times or under different conditions. These chest CT images may be taken on separate occasions or during different phases of treatment, such as prior to and following chemotherapy or radiation therapy. Through image registration, medical professionals can align these images to identify changes in the size, shape, or position of chest organs and structures for comparison. Registering chest CT images serves various purposes in medical practice, including identifying changes in the lungs (such as tumor growth or infections), tracking disease progression, assessing the effectiveness of treatments, and assisting with the precise placement of needles or medical instruments during biopsies or surgeries.

Various methods can be employed for chest CT registration, including manual and automated approaches using dedicated software. Manual registration relies on visual alignment of the images, whereas automated registration utilizes algorithms to analyze and determine the optimal alignment. Automated methods are often quicker and more precise compared to manual techniques. In this project, we implemented the lung computed tomography (CT) image registration for a given dataset using intensity and deep learning-based approaches. The dataset includes 4 pairs of volumetric scans from patients taken at two distinct phases (inhale and exhale), along with their corresponding annotations (landmark points). We evaluated the registration performance using Target Registration Error (TRE), which measures the 3D Euclidean distance between the aligned landmarks. The next sections of this report will explain the methodology and the analysis of the results obtained of the lung CT image registration project.

2. MATERIALS AND METHODS

2.1. Dataset description

The train set provided consisted of 4 patients' lung CT scans, taken from the Chronic obstructive pulmonary disease (COPD) dataset. Each patient's file consists of:

1) An Inhale chest CT scan in raw binary format.

2) An Exhale chest CT scan in raw binary format.

3) A text file containing 300 Lung landmarks of the inhale CT scan.

4) A text file containing 300 Lung landmarks of the exhale CT scan.

Additionally, we were provided with a dataset description with information such as image dimensions, voxel size (mm), features, mean and standard deviation of the displacement (mm), repeats and observers for our reference. Amongst these descriptors, we noted the most important ones which we call "descriptors for image handling" that are useful for the task and are image dimensions and voxel size (mm). We also considered the given mean and standard deviation of the displacement for the initial verification of the preprocessing step.

The test set provided consists of 3 patient's lung CT scans with each file consisting of 3 out 4 of the information as given in the train set with the exception of the 300 exhale landmarks.

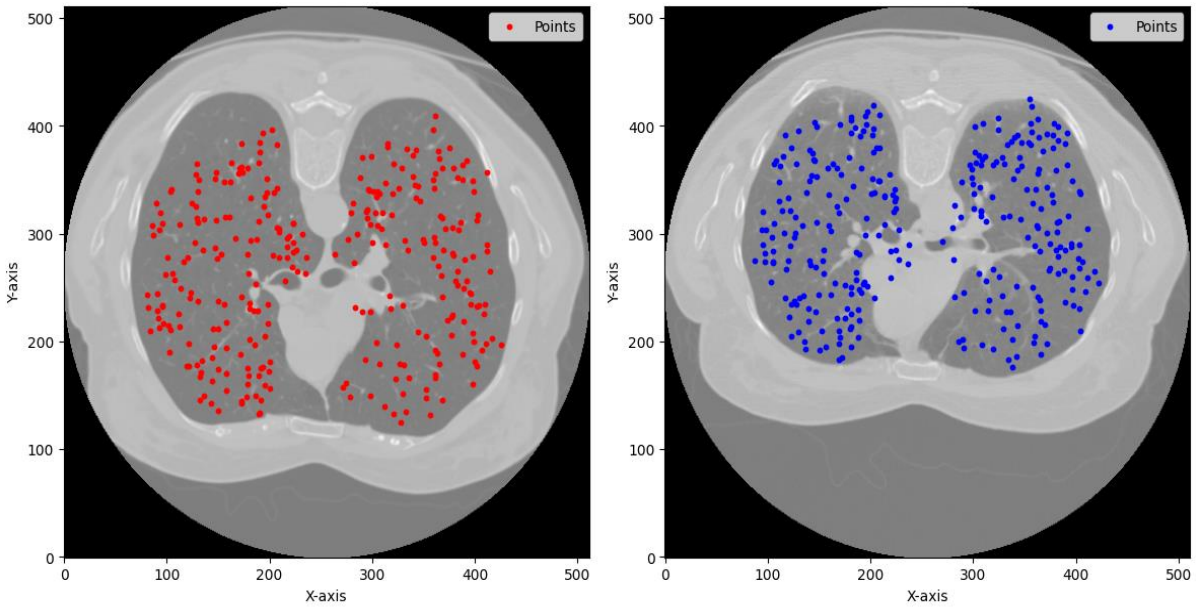


Fig 1. The inhale iBH-CT (left) slice and the exhale eBH-CT (right) slice of COPD1 image with corresponding landmark points.

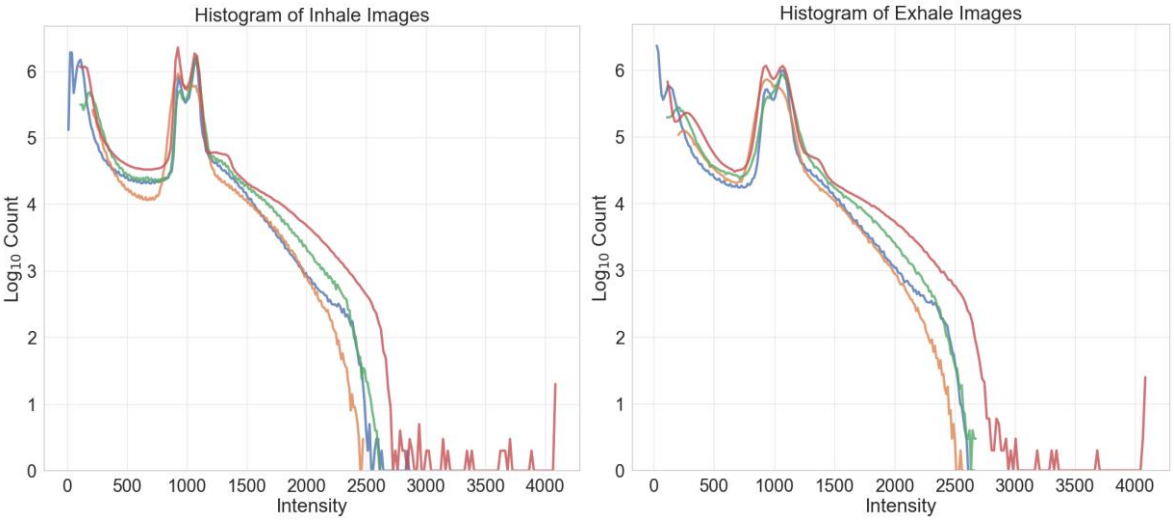


Fig 2. Histogram of voxel intensity values of the inhale iBH-CT (left) images and exhale eBH-CT (right)) images of COPD1-COPD4.

2.2. Preprocessing

The first step was to load and visualize the given images, but since we already identified that they weren't in the usual NIFTI format we needed to convert them accordingly and this was done with ITK-Snap. The following describes how we achieved this:

1. We load the images into ITK-snap by selecting **Raw Binary** as the file format instead of the usual **Analyze.**

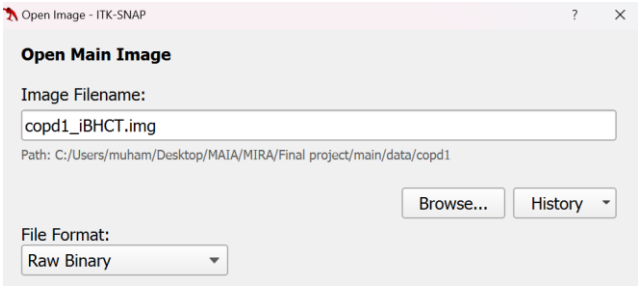


Fig 3. Loading image in ITK-SNAP

2. Next we input the image dimensions, voxel spacing and voxel type (int16) with the information provided in the dataset description.

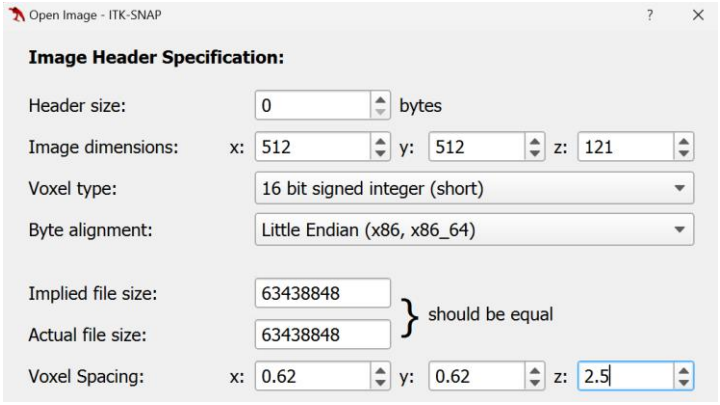


Fig 4. Inputting image dimensions in ITK-SNAP

3. Since the landmarks were given in the RAS Orientation format, we needed to re-orient the images from RAI to RAS. We did this by usual the Reorient Image function given in the tool section of ITK-Snap.

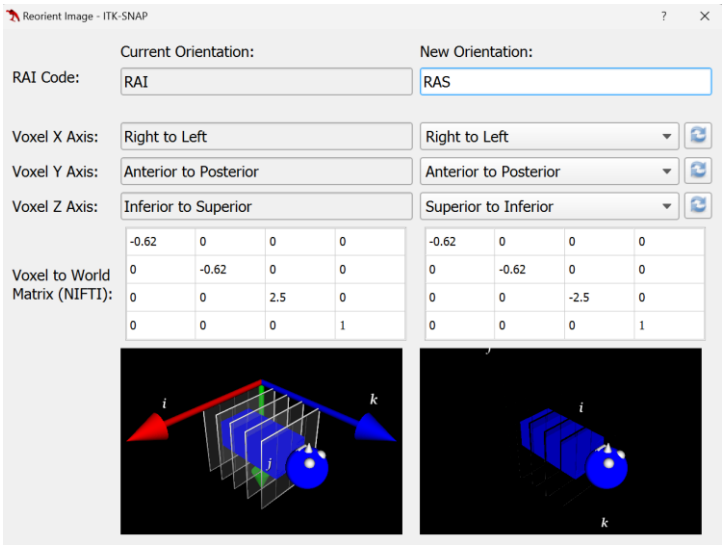


Fig 5. Re-orienting images in ITK-SNAP

4. Finally, save the re-oriented images as NIFTI files and proceed with the next steps.

Regarding the landmarks, they needed to be converted from text files to .pts format to work with SimpleITK. Thus, we created a function that converts the .txt files to .pts using the number of points as the header of the file.

2.3. Registration

A fundamental image registration framework comprises four primary components:

1. **Transform**: Applies a transformation to the moving image to align it with the fixed image.
2. **Metric**: Measures the similarity between the moving and fixed images.
3. **Interpolator**: Assigns intensity values to pixels outside the original grid after applying the transformation.
4. **Optimizer**: Identifies the minimum or maximum of the metric using an optimization algorithm.

Developing an effective registration method involves selecting optimal parameters for these components. For this task, we conducted multiple experiments by tuning these parameters. The parameters used for each experiment and the results obtained are discussed in the next sections.

2.4. Performance metrics

Recall that we were provided in the dataset description the mean and standard deviation of a displacement. Well, these values are called the target registration error (TRE) and represent the displacement between each inhale and exhale landmarks before registration. For this task, the performance of our algorithm is evaluated using the TRE, and for each patient, we can compute the mean TRE and its standard deviation (STD). Recalling that we also validated our initial preprocessing steps using this displacement, we provided the original displacement as well as our obtained result in the tables below.

Image	Displacement means and std (mm)
COPD1	25.90 (11.57)
COPD2	21.77 (6.46)
COPD3	12.29 (6.39)
COPD4	30.90 (13.49)

Table 1. Original Values

Image	Displacement means and std (mm)
COPD1	26.14 (11.33)
COPD2	21.64 (6.41)
COPD3	12.62 (6.38)
COPD4	29.58 (12.92)

Table 2. Validated Values

To facilitate comparison across multiple registration algorithms, a single representative value was derived by averaging two metrics; TRE (mean and STD), and execution time across all patients for each experiment. These values are presented in the next sections.

3. EXPERIMENTAL RESULTS & DISCUSSION

SimpleElastix Registration

The selection of the registration parameters is a critical aspect of our work, as these parameters determine the registration process and, consequently, the performance of our algorithm. To register our images, we adopt the *Elastix* library in python called *SimpleElastix* because it is simple to use and because it provides default parameters that we can use to kick off our experimentation. *SimpleElastix* also makes it easy to conduct multiple experiments using different parameters by simply setting up the parameters in a file called the parameter file. It also makes it easy to use multiple parameters as needed. Therefore, we define a function using *SimpleElastix* such that it takes as input, the images, landmarks and optionally a list of parameter files to use to register and transform the fixed landmarks.

3.1. Experiments

We explored both non-rigid and rigid transformation even though the patient movement between the inhale and exhale images is clearly a non-rigid movement. This allows us to compare speed, computational costs, and possible composition between transforms to improve the registration performance.

3.1.1. Default Affine

We began our experiment with simple parameters, meaning that we use the default *affine* transform provided by *SimpleElastix.* A code snippet is shown below, and we present the result of this experiment in Table 3.

```
if param == 'default_affine':
    parameter_map_affine = sitk.GetDefaultParameterMap("affine")
    parameterMapVector.append(parameter_map_affine)
```

Fig 6. Code snippet for Default Affine

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	25.60	11.22	22.35
COPD2	22.28	6.01	19.61
COPD3	7.58	3.36	23.25
COPD4	26.08	10.17	24.25
Mean Value	**20.385**	**7.69**	**89.46**

Table 3. Default Affine results

3.1.2. Default Bspline

Our second was conducted using the default *bspline* transform provided by *SimpleElastix.* We also present a table showing the result of the experiment.

```
elif param == 'default_bspline':
    parameter_map_bspline = sitk.GetDefaultParameterMap("bspline")
    parameterMapVector.append(parameter_map_bspline)
```

Fig 7. Code snippet for Default Bspline

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	23.71	11.34	118.19
COPD2	18.81	6.87	103.78
COPD3	10.49	5.60	131.03
COPD4	28.06	12.72	121.18
Mean Value	**20.27**	**9.13**	**118.55**

Table 4. Default Bspline results

As shown in both tables, the results weren't impressive as they were too close to the initial TRE values before registration and the *bspline* experiment took too much time.

We quickly realized that the default parameters weren't suitable for the given task, thus we looked at the defacto hub for elastix parameters, the *ElastixModelZoo* on Github! We identified the studies of Par0003, Par0004, Par0008, Par0011, and Par0015. They were performed on intrapatient 3d lung CT scan datasets. Because of the similarity between our projects, we tried the parameters of these studies.

For some reason which we couldn't figure out, all the parameter files we tried from Par0003, Par0004, Par0008, and Par0015 didn't run. Thus, we proceeded to with those of Par0011.

3.1.3 Par0011 Affine

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	25.14	10.19	28.88
COPD2	25.86	5.32	25.04
COPD3	7.45	3.34	30.26
COPD4	22.91	8.49	30.74
Mean Value	**20.34**	**6.84**	**28.73**

Table 5. Par0011 Affine results

Seeing the result of this experiment, one thing became clear, affine registration isn't the best solution for this task. This is understandable since affine registration is a global registration, but a patient's breathing rhythm wouldn't be uniform everywhere. Hence, we needed to stick to a local registration i.e. bspline.

3.1.4 Par0011 Bspline

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	8.22	5.58	103.52
COPD2	14.38	6.89	96.65
COPD3	5.01	3.14	130.73
COPD4	8.61	4.16	112.93
Mean Value	**9.06**	**4.94**	**110.96**

Table 6. Par0011 Bspline results

The result of this experiment clearly indicates that a bspline registration is most suitable for this task with the downside being the computation time. Next, we investigate the set of parameters that achieved this performance. From Par0011 Bspline file, we extracted the important parameters and values as shown in the table below. The parameters marked with an

* are those which we believe could be adjusted for improvement based on the variations in those of the other files.

Parameter	Value
FixedImagePyramid	FixedRecursiveImagePyramid
MovingImagePyramid	MovingRecursiveImagePyramid
Interpolator	BSplineInterpolator
ResampleInterpolator	FinalBSplineInterpolator
Resampler	DefaultResampler
Registration*	MultiMetricMultiResolutionRegistration
Metric*	AdvancedNormalizedCorrelation TransformBendingEnergyPenalty
Optimizer*	AdaptiveStochasticGradientDescent
NumberOfResolutions*	5
MaximumNumberOfIterations*	1000
ImagePyramidSchedule*	16 16 16 8 8 8 4 4 4 2 2 2 1 1 1
Transform	BSplineTransform
FinalGridSpacing*	10.0 10.0 10.0
NumberOfSpatialSamples*	2000

Table 7. Parameter and Values

Looking through the other parameter files, we found that while some parameters are constant across all the files, there were a few discrepancies that lie around these parameters. The values with an * appear more frequently than others.

Parameter	Value
Registration	MultiMetricMultiResolutionRegistration MultiResolutionRegistration*
Metric	AdvancedNormalizedCorrelation TransformBendingEnergyPenalty AdvancedMattesMutualInformation*
Optimizer	AdaptiveStochasticGradientDescent* StandardGradientDescent
NumberOfResolutions	4 5*
MaximumNumberOfIterations	1000* 2000
ImagePyramidSchedule	16 16 16 8 8 8 4 4 4 2 2 2 1 1 1 16 16 8 8 8 4 4 4 2 2 2 1 1 1 1 32 32 8 16 16 4 8 8 2 1 1 1 1 1 1
NumberOfSpatialSamples	2000* 10000
SP_A	100

Table 8. Parameter and Values

Based on this analysis, we came up with several combinations to modify the Param 011 file.

3.1.5 Modification 1

Parameter	Value
Registration	MultiResolutionRegistration
Metric	AdvancedMattesMutualInformation
NumberOfResolutions	5
MaximumNumberOfIterations	1000
ImagePyramidSchedule	16 16 16 8 8 8 4 4 4 2 2 2 1 1 1
SP_A	100
NumberOfSpatialSamples	2000

Table 9. Parameter and Values for Modification 1

Image	**Mean TRE (mm)**	**STD TRE (mm)**	**Time (sec)**
COPD1	6.55	5.28	94.27
COPD2	13.10	7.42	104.38
COPD3	4.54	3.60	95.96
COPD4	8.24	4.68	91.58
Mean Value	**8.11**	**5.75**	**96.55**

Table 10. Results for Modification 1

3.1.6 Modification 2

Parameter	Value
Registration	MultiResolutionRegistration
Metric	AdvancedMattesMutualInformation
NumberOfResolutions	5
MaximumNumberOfIterations	1000
ImagePyramidSchedule	16 16 8 8 8 4 4 4 2 2 2 1 1 1 1
SP_A	100
NumberOfSpatialSamples	2000

Table 11. Parameter and Values for Modification 2

Image	**Mean TRE (mm)**	**STD TRE (mm)**	**Time (sec)**
COPD1	6.10	5.12	94.38
COPD2	10.92	6.80	82.84
COPD3	4.24	3.46	101.74
COPD4	8.83	5.27	96.65
Mean Value	**7.02**	**5.16**	**93.90**

Table 12. Results for Modification 2

3.1.7 Modification 3

Parameter	Value
Registration	MultiResolutionRegistration
Metric	AdvancedMattesMutualInformation
NumberOfResolutions	5
MaximumNumberOfIterations	1000
ImagePyramidSchedule	32 32 8 16 16 4 8 8 2 1 1 1 1 1 1
SP_A	100
NumberOfSpatialSamples	2000

Table 13. Parameter and Values for Modification 3

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	6.43	5.42	96.75
COPD2	11.31	6.56	84.52
COPD3	4.94	3.70	106.69
COPD4	10.07	5.39	104.81
Mean Value	**8.19**	**5.27**	**98.69**

Table 14. Results for Modification 3

While there is an overall performance boost from the results of param 011, it was observed that the ***ImagePyramidSchedule*** plays an important role in registration, and that which we used in modification 2 performed better. We also decided to increase the number of iterations in the next experiments to observe its effect as well as the Number of spatial samples.

3.1.8 Modification 4

We tried different values for ***NumberOfSpatialSamples*** and ***MaximumNumberOfIterations*** and the best combination and results are:

Parameter	Value
Registration	MultiResolutionRegistration
Metric	AdvancedMattesMutualInformation
NumberOfResolutions	5
MaximumNumberOfIterations	3000
ImagePyramidSchedule	16 16 8 8 8 4 4 4 2 2 2 1 1 1 1
SP_A	100
NumberOfSpatialSamples	10000

Table 15. Parameter and Values for Modification 4

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	4.46	4.78	236.49
COPD2	7.89	6.73	276.09
COPD3	3.01	3.05	269.07
COPD4	11.16	7.53	242.39
Mean Value	**6.13**	**5.52**	**256.51**

Table 16. Results for Modification 4

While the results obtained were satisfactory, we sought to enhance them by incorporating a segmentation mask into the registration framework. To achieve this, we explored several methods to generate a lung mask for each patient, with three primary approaches standing out:

1. **Segmentation using Foundation Models (e.g., SAM2, BiomedParse)**
2. **Segmentation using the Lung Mask Library.**
3. **Segmentation with a pretrained UNET model.**

3.1.9 Segmentation Using Foundation Models

For the foundation models, we experimented with **BiomedParse**. However, these models are designed to handle 2D slices rather than 3D volumes. To address this, we developed a preprocessing script to slice the 3D images into individual 2D slices, apply segmentation, and subsequently reconstruct the 3D volume from the 2D segmentations. This workflow allowed us to utilize the capabilities of these foundation models effectively for the 3D segmentation,

however it's accuracy also depends on the text prompt used. This challenge as well the loss of spatial information (3D) and the prediction time (usually 512 predictions per image) made us prefer the segmentation using the lung mask library. Another observation was that BiomedParse was designed to segment specific diseases like COVID-19 infection, nodule or tumor and not an entire organ, hence the under segmentations.

3.1.10 Segmentation Using Lung Mask Library

For the lung mask library, we employed the **R231CovidWeb pretrained model**. This model requires input images in Hounsfield Units (HU), necessitating preprocessing of the raw intensities. Thus, we used an approximation to convert raw pixel intensities and HU values. Specifically, we transformed the pixel intensities using the function:

$f(x) = x - 1024$

This approximation allowed us to generate acceptable HU values for the input images, enabling the use of the R231CovidWeb model for lung mask segmentation.

3.1.11 Segmentation with a pretrained UNET model.

We found from a repository a model called "3D Patch-Based Lung Segmentation on CT" and experimented with the pretrained model on our dataset. But the result we obtained wasn't as expected or claimed in the repository, therefore we decided to proceed with our experiments with the results obtained from the lung mask library.

We present sample segmentation masks from the models as well as the registration results using the best obtained mask (the lung mask library) and our best parameters (modification 4).

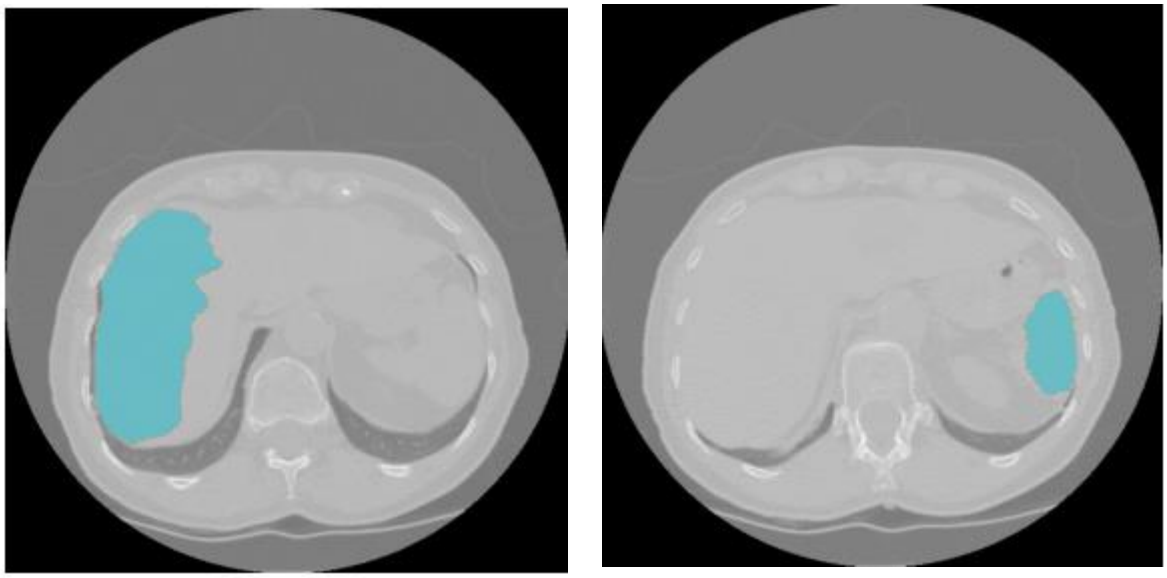


Fig 8. Biomedparse segmentation

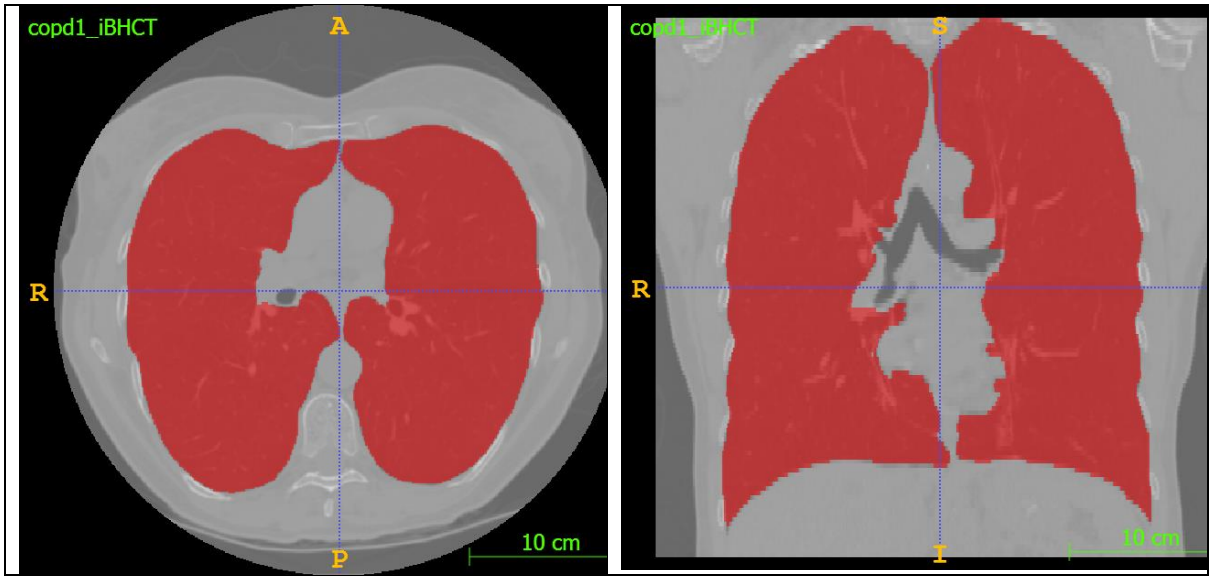


Fig 9. Lung Mask segmentation

3.2 Results

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	1.14	0.74	326.13
COPD2	2.11	2.83	350.70
COPD3	1.15	0.75	380.7
COPD4	1.35	0.75	349.42
Mean Value	**1.44**	**1.77**	**351.24**

Table 18. Registration Results obtained with Lung Mask Segmentation

Our last sets of experiments involve different preprocessing steps in an attempt to improve our overall performance. We started with a simple intensity normalization, followed by contrast enhancements and lastly a denoising process. Below we provide the results achieved using the different preprocessing steps with our best parameter file and the lung mask segmentation.

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	1.15	0.75	241.78
COPD2	2.08	2.81	242.69
COPD3	1.16	0.78	263.30
COPD4	1.35	0.75	285.56
Mean Value	**1.44**	**1.77**	**258.33**

Table 19. Results for Intensity Normalization

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	1.10	0.60	239.30
COPD2	3.03	4.91	244.17
COPD3	1.10	0.62	262.68
COPD4	1.34	1.06	250.53
Mean Value	**1.64**	**1.80**	**249.67**

Table 20. Results for Contrast Enhancement

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	1.10	0.59	243.21
COPD2	2.99	4.86	248.67
COPD3	1.10	0.63	261.94
COPD4	1.34	1.02	254.00
Mean Value	**1.63**	**1.78**	**251.96**

Table 21. Results for Intensity Normalization + Contrast Enhancement

Image	Mean TRE (mm)	STD TRE (mm)	Time (sec)
COPD1	1.46	1.26	241.82
COPD2	2.93	3.46	253.88
COPD3	1.33	0.95	261.62
COPD4	2.87	4.15	250.66
Mean Value	**2.15**	**2.46**	**251.50**

Table 22. Results for Denoising

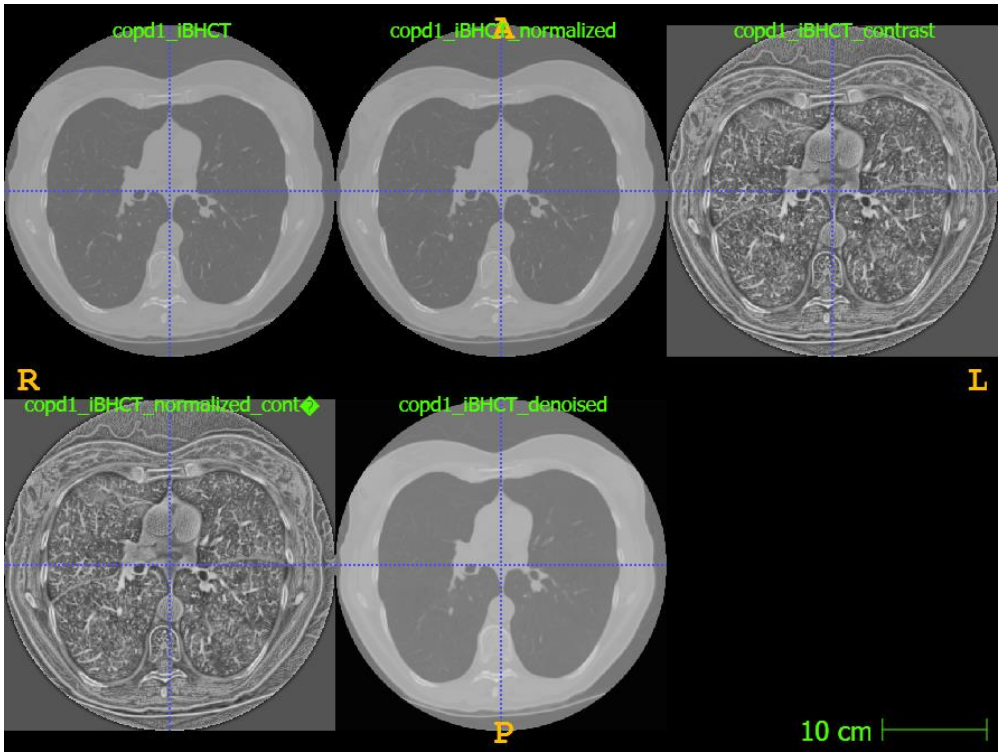


Fig 10. Visualization of Image preprocessing steps.

3.3 Result Discussion

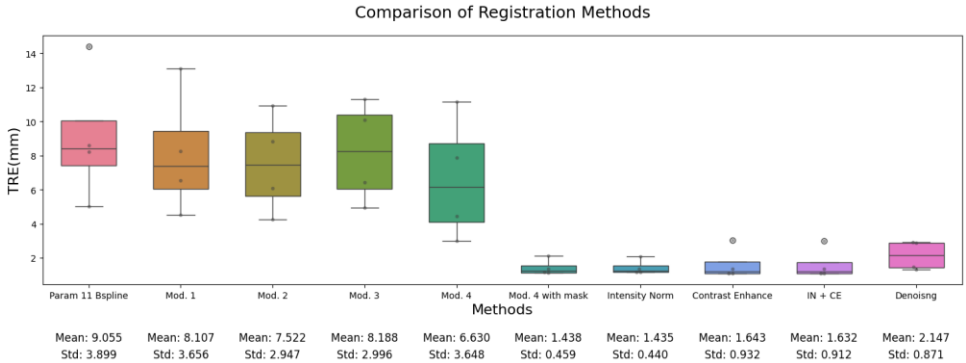


Fig 11. Comparison of Registration Methods

To summarize the experiments conducted, the results have been presented in box plots illustrated in Figure 8. and the mean values of each case have been calculated to evaluate performance improvements. The experiments demonstrated the influence of key parameters on the final registration outcomes. Key observations include:

3.3.1. Parameter Optimization Impact

Changing the NumberOfSpatialSamples, MaximumNumberOfIterations, and ImagePyramidSchedule yielded significant performance improvements. By systematically selecting parameter values from prior studies, the final results demonstrated marked reductions in mean TRE and STD TRE.

Specifically, modifications to MaximumNumberOfIterations and ImagePyramidSchedule optimized the balance between computational cost and accuracy, showing that increasing the iteration count improves alignment precision but also adds to processing time.

3.3.2. Segmentation Contribution

Introducing segmentation using the Lung Mask Library provided a major breakthrough. The segmentation masks successfully isolated lung regions, allowing registration to focus on critical areas and thus reducing errors significantly.

The mean TRE was reduced by more than 500%, confirming the importance of segmentation in medical image registration tasks. Notably, the segmentation masks preserved spatial information while reducing computational complexity.

3.3.3. Preprocessing Contribution

A simple preprocessing like intensity normalization boosts the performance of our registration algorithm. Although the same cannot be said for denoising as it had negative effects.

3.3.4. Outlier Case (Patient COPD2)

One interesting anomaly was Patient COPD2, whose mean TRE consistently exceeded 2 mm across methods. This indicates the presence of unique anatomical or scanning variations that posed challenges to all registration techniques attempted.

Despite parameter tuning and segmentation refinements, this outlier highlights potential limitations of the registration framework in handling extreme variations.

3.3.5. Computational Costs and Trade-offs

Before preprocessing steps were implemented, the final optimized model achieved a mean TRE of 1.44 mm with a mean STD TRE of 1.77 mm and mean execution time of 351.24 seconds across all cases. One observation aside the performance boost was that the preprocessing step optimized the computational cost of our registration algorithm . For instance, the mean execution time after intensity normalization reduced to about 250 seconds.

Incremental adjustments to the parameters (e.g., increasing spatial samples and iterations) progressively improved alignment but demanded additional computation time.

B. VoxelMorph Registration

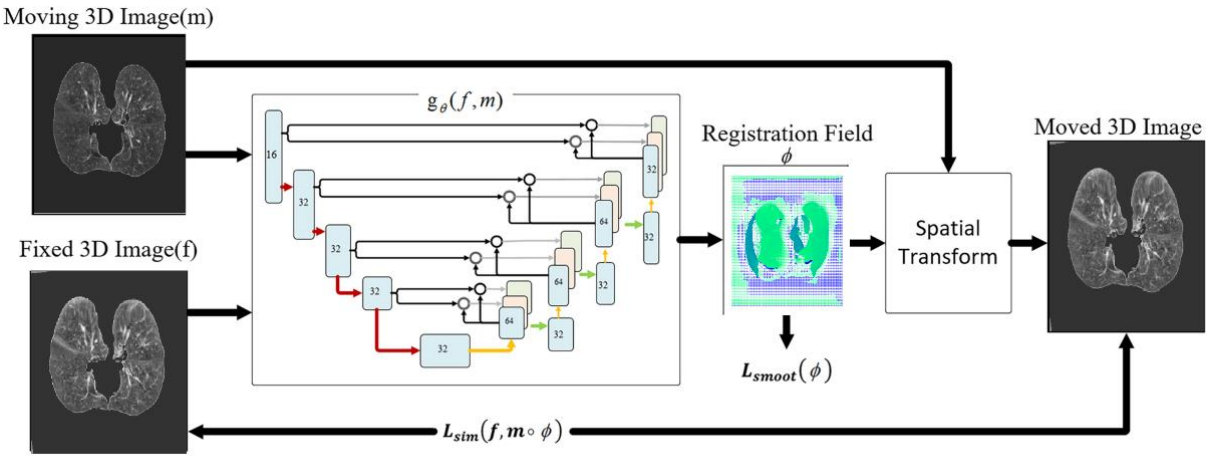


Fig 12. Pipeline for deep-learning based registration using VoxelMorph.

As a comparison to SimpleElastix registration, we explored the use of VoxelMorph, a deep learning-based approach for image registration. VoxelMorph utilizes a convolutional neural network (CNN) to learn an optimal transformation between images for non-rigid registration tasks. We applied this method to our lung CT scan dataset and compared the results to the previously tested algorithms.

1) VoxelMorph Architecture: The VoxelMorph framework consists of a 3D U-Net architecture designed to estimate dense deformation fields between input images. The network takes two images as input: a fixed image and a moving image, and outputs the deformation field that aligns the moving image to the fixed image. This approach is particularly advantageous for medical image registration due to its ability to handle complex, non-linear deformations.
2) Registration Pipeline with VoxelMorph: For this experiment, we integrated the VoxelMorph framework into our registration pipeline. The input to the model was a pair of lung CT images, with the moving image being deformed to align with the fixed image. Fig. 10 shows the proposed voxel morph pipeline.

3.4. Experiments

Since we are working with very limited data, the VoxelMorph model was trained using 2 cross validation strategies, with 100 epochs for each training strategy. The cross-validation strategies are as follows.

3.4.1. 2-Fold Cross-Validation:

We manually split the data into two parts, using one set for training, and the other for validation. i.e. First fold: Train images: 1&2, Validation images 3&4. Second fold: Train images: 3&4, Validation images 1&2.

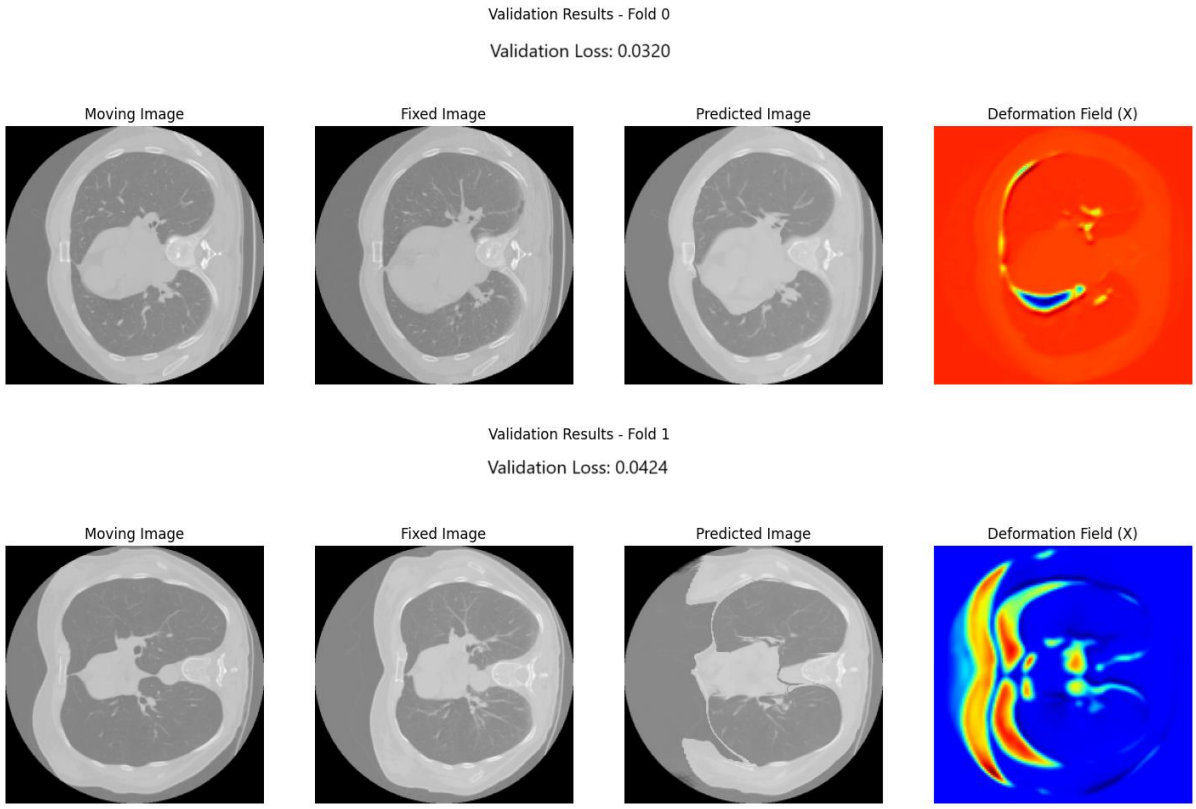
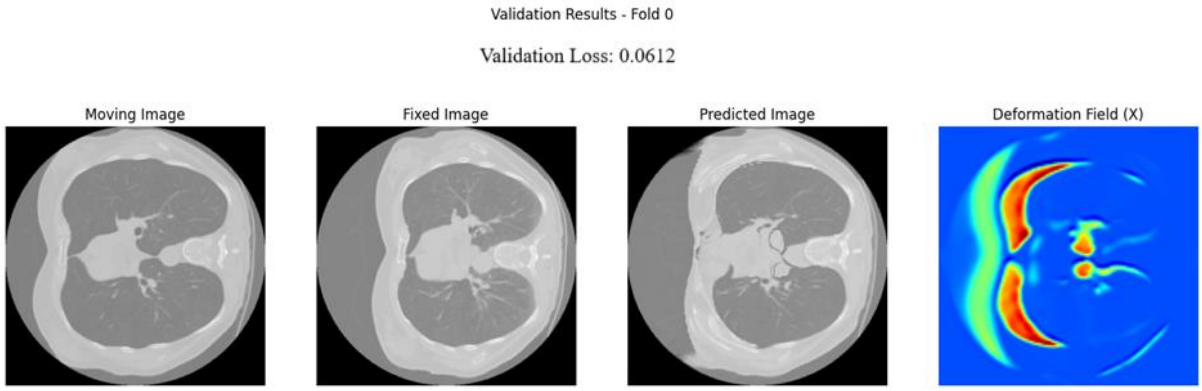


Fig 13. Results obtained using 2-Fold Cross-Validation

3.4.2. Leave-One-Out Cross-Validation (LOO-CV):

For LOO-CV, we trained on three images and validate on the one left out. Since we have 4 images, we repeat the training process four times with a different validation image during each training. We evaluated each of the training experiments using the TRE.



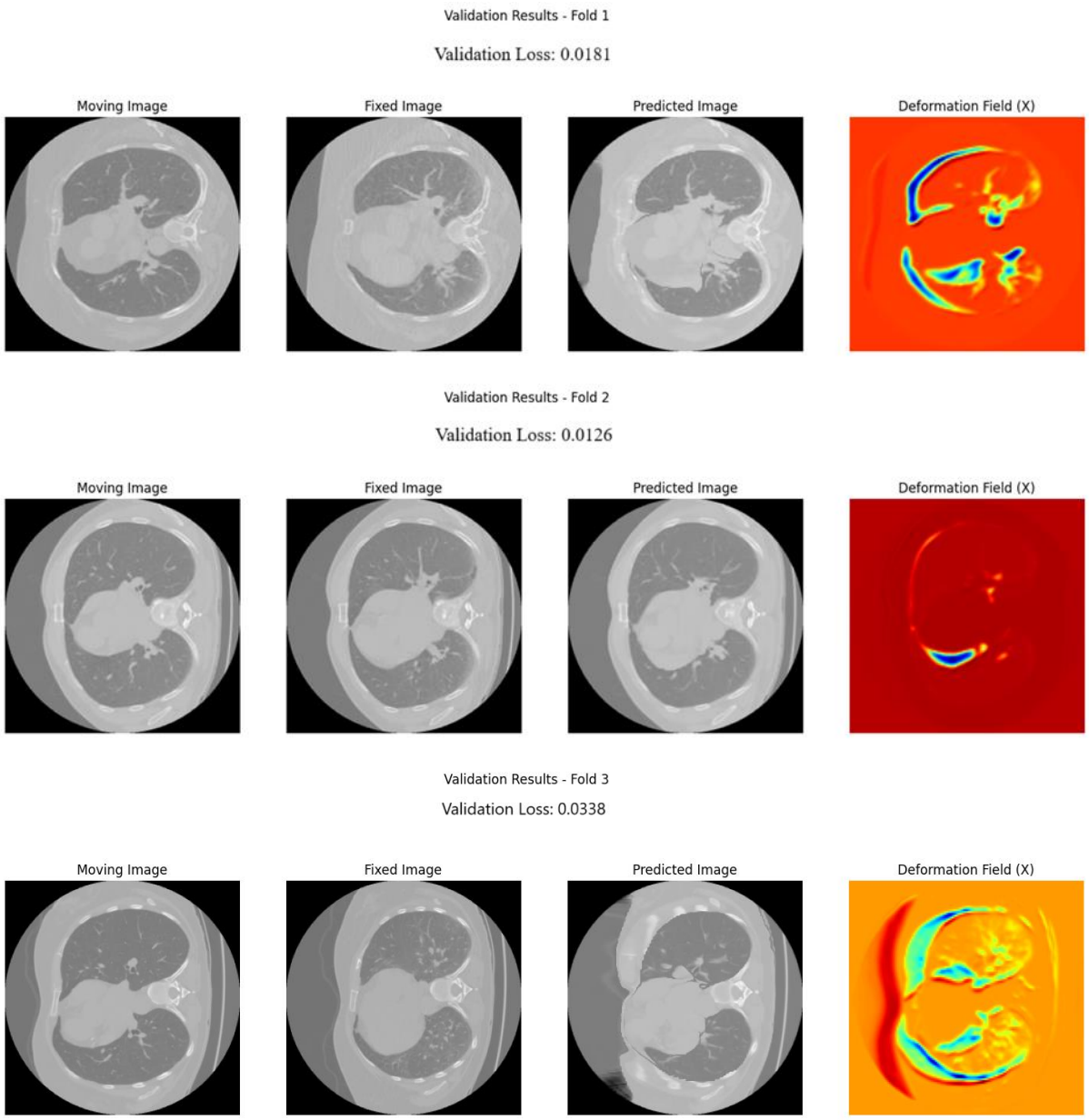


Fig 14. Results obtained using Leave-One-Out Cross-Validation

It is obvious that the performance of the voxelmorph registration isn't comparable to that obtained with SimpleElastix. This poor performance can be attributed to several factors but most importantly the shortage of training data.

4. CONCLUSION

In conclusion, this study demonstrated the effectiveness of parameter optimization and segmentation masks in enhancing lung CT scan registration accuracy. By carefully tuning registration parameters and integrating lung segmentation techniques, the mean TRE was significantly reduced. The use of the Lung Mask Library emerged as a pivotal enhancement, improving alignment precision while balancing computational costs. Despite the promising outcomes, challenges remain, particularly for outlier cases like COPD2, which highlight the need for further refinement and specialized approaches. Future work should explore deep

learning models, GPU-accelerated algorithms, and adaptive preprocessing strategies to address these challenges. Overall, this project provides a robust foundation for future advancements in medical image registration, paving the way for improved diagnostic and analytical applications in clinical settings.

References

[1] A. Hering, S. Häger, J. Moltz, N. Lessmann, S. Heldmann, and B. van Ginneken, "CNN-based lung CT registration with multiple anatomical constraints," *Med. Image Anal.*, vol. 72, p. 102139, Aug. 2021, doi: 10.1016/j.media.2021.102139.

[2] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim, "elastix: a toolbox for intensity-based medical image registration," *IEEE Trans. Med. Imaging*, vol. 29, no. 1, pp. 196–205, Jan. 2010, doi: 10.1109/TMI.2009.2035616.

[3] J. B. Maintz and M. A. Viergever, "A survey of medical image registration," *Med. Image Anal.*, vol. 2, no. 1, pp. 1–36, Mar. 1998, doi: 10.1016/s1361-8415(01)80026-8.

[4] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: a survey," *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1153–1190, Jul. 2013, doi: 10.1109/TMI.2013.2265603.