Chest CT Scan Registration: COPD Challenge

Medical Image Registration and Applications
Joaquin Oscar Seia Kaouther Mouheb Marawan Elbatel

I. Abstract

With the advancements in the computer vision and image analysis fields, the research community has proposed multiple image registration techniques that proved to be very useful in the case of medical image analysis. In this work, we used and compared some of the main proposed methods for medical image registration applied to the use case of lung computed tomography (CT) registration. We compared two classical approaches, namely Elastix and ANTsPy with a novel deep learning approach, VoxelMorph. The three techniques were used to register inhaling and exhaling CT scans coming from the same patient. Besides, we investigated the effect of using different preprocessing techniques on the registration performance for each technique. Our work proved that Elastix outperformed the other methods. The results of our work were submitted to the Medical Image Registration and Applications course challenge and achieved the best performance in terms of target registration error.

Keywords: medical image registration, Elastix, lung computed tomography

II. Introduction

When diagnosing pulmonary diseases, computed tomography (CT) plays a fundamental role. This image modality provides a non-invasive technique to study the anatomy and in some cases even the physiology (4D images) of the lungs. However, when analyzing sequential data from the same patient or comparing images coming from different subjects, registration techniques become mandatory. An accurate registration of the thoracic CT is extremely useful to clinically assess the condition of the patient, but at the same time it represents a challenging task due to the elastic nature of the lung tissue deformation. When acquiring a temporal series of scans from the same patient, in addition to the potential movement of the subject, the physiological respiratory movements of the lungs during the inhaling and exhaling phases, generate an elastic deformation over the lung tissue. In some cases, these movements need to be compensated in order to properly assess the affected structures.

A. Objective

The objective of this work was to put our theoretical knowledge acquired during the Medical Image Registration and Applications (MIRA) course into practice to solve the task presented by the CODP DIR-Lab challenge held out by the Emory University School of Medicine [1].

The problem implied thoracic CT registration between the inhaling and exhaling phases of the same patient. The aim was to develop a pipeline that accurately registers two chest CT scans in a minimal computation time. The two CT scans came from the same patient, one belonged to the exhaling phase and the other to the inhaling one. Trying to mimic the usual structure of medical imaging challenges, we were provided a training set and later a hidden test set that was used to evaluate the performance of the developed algorithms.

III. Materials and methods

In the following subsections, we cover the data we used in our work and the methods we implemented in order to register the images. We first discuss the parsing of the raw images into NIFTY ones and the common preprocessing steps across all registration methods. Then we present a lungs and body segmentation pipeline used to extract their corresponding masks. Lastly, we present the three different methods we implemented for registering the volumes: Elastix-based, ANTsPy-based and a Deep Learning-based method using VoxelMorph. The code of this project is done in Python language and is publicly available in https://github.com/joaco18/mira_final_project

A. Data

Each case provided in the CODP DIR-Lab challenge included the thoracic CT image pairs for the inhaling and exhaling phases and the coordinates of a set of 300 landmarks manually located by medical professionals in both volumes. The CT scans had an average size of 512x512x120.3 pixels with an average pixel spacing of 0.641x0.641x2.5. The data was provided in raw format.

The organizers divided the original dataset of 10 cases in two independent sets, a training set consisting of 4 cases and a hidden test set which included only 3 cases. For the latter, during the evaluation time, for each case the two volumes were provided but only the inhaling landmarks coordinates and not the exhaling ones.

B. Evaluation metrics

The results of the challenge were evaluated according to the registration accuracy measured by the target registration error and by the computational time required to register the images.

Target registration error (TRE) is given by the Euclidean distance between the fixed landmarks (exhale) and the transformed moving landmarks (inhale).

$$TRE = \sum_{n=1}^{N} \sqrt{(x_i^n - x_e^n)^2 + (y_i^n - y_e^n)^2 + (z_i^n - z_e^n)^2}, \quad (1)$$

where N represents the total number of landmarks (300) and (x_i^n, y_i^n, z_i^n) represents the coordinates in the physical space of the n^{th} landmark in the inhaling volume, and (x_e^n, y_e^n, z_e^n) in the exhaling one.

C. Data Preparation

In this section, the data preparation process is described.

1) Parsing the data

The CT volumes were originally provided in raw format. We converted them to NIFTY format programmatically, in general terms, the implemented function defined a header file to adequately read the image with simpleITK's *ReadImage* function. Among the data provided in the header file, the data type (Int16) and some of the metadata (volume size and the pixel spacing) provided by the challenge organizers were utilized. When providing the voxel spacing, we applied the negative of the z spacing, in this way, we modified the orientation of the image and the volume appeared in superior-inferior orientation across the Z axis.

Also during parsing, we generated a modified version of the landmark files, which included two extra rows containing "index" and "300" respectively. This was required for Elastix's Transformix interface to be able to read the points correctly (300 points in index format coordinates). Additionally, a landmarks mask was generated for visualization purposes of the target points.

During the parsing of the data, a comma separated value (CSV) file was generated including all the dataset information. Some of the fields included in this file were: partition, size_x, size_y, size_z, space_x, space_y, space_z, case, disp_mean, disp_std, observers_mean, observers_std, lowest_mean, lowest_std. As it can be seen, the expected values for initial TRE and the best registration results were included for fast comparisons.

2) Preprocessing

Across the different experiments we run, different preprocessing techniques were evaluated.

For Elastix-based registration, several preprocessing techniques were explored. Min-max normalization was developed in a way such that it could take into account only the voxels' intensities inside a mask -such as the lungs or body regions- to further focus on the desired region. The method took the minimum and maximum values in the desired region and linearly maps them into the range [0, 1]. The intensities outside the mask were clipped in order to force consistency. Another explored technique was lung windowing of the scan's intensities to highlight the contrast in the lung area. This method utilized a window of [-1024, 600], as suggested by [2]. Lastly, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to perform a locally adaptive histogram equalization that better highlighted the contrast of the image.

For ANTsPy, a particular preprocessing function was implemented using the methods provided by the iMath module of ANTsPy. This function normalized and truncated the intensities of an image stored in an ANTsImage object.

For the deep learning method, the previously mentioned normalization function was enriched to support z-score normalization (also known as standardization). Both min-max and z-score normalization techniques were explored.

3) Segmentation

We developed a method for performing lung and body segmentation in chest CT scans. The pipeline takes advantage of three main elements: The high contrast between the intensities of lung parenchyma, body tissues and background; the concentric nature of these three regions and the continuous shape of the lungs across the different slices in z-axis. The general pipeline can be presented as:

The segmentation of the lungs is performed one axial slice at the time. The segmentation of the lungs is not a difficult task since the intensity profile of the structure allows a fast threshold based segmentation. However, the trachea (which is not part of the lungs and did not contain any landmark in our problem) has almost the same intensity properties than the lungs and in the apical slices it can have almost the same size as the lungs regions.

To get rid of the trachea, we designed a sequential segmentation procedure which constrained the segmented areas in one slice to be overlapped to the lung regions in the previous one. The process starts by identifying the middle slice across the z axis of the volume and getting the lungs' segmentation on it. In the middle region, there is zero probability that the lungs regions are not the two biggest lung-like regions segmented, therefore keeping the two biggest connected components represents a safe choice. As we go upwards or downwards, the trachea or other abdominal structures, can be confused with the lungs, but imposing the overlap with the previous slice's lungs' mask, lets us keep only the actual lung region and avoid undesired structures.

In each slice, the segmentation is performed with the following steps

1) A global thresholding was applied with an intensity threshold of 600.
2) The connected components of the obtained binary mask were extracted.
3) The background region was identified based on location properties, a unified mask of the whole body and CT scanner's table was obtained.
4) The body was separated from the CT scanner's table region based on area size properties, then the latter was removed.
5) The body mask was used to select the components inside the patient's body.
6) If the mask of the previous slice was provided, it was used to constraint the segmentation in the current slice.
7) The lung masks were obtained by keeping only the two largest components inside the body. (Overlapping the previous slide lung mask)
8) Hole filling was used to fill in the gaps inside the lungs masks corresponding to bronchus and vessels.
9) The function returns both the body and the lung masks, which were stored in one single array with two different categorical labels.

4) Dataset

In order to standardize the process of retrieving and preprocessing the data between the different methods and experiments, we created a dataset class in a PyTorch-compatible fashion that provided a set of functionalities and options to prepare the data for registration. The database's CSV file generated during the parsing stage was used by this class to easily read the images and useful metadata for each case.

To control the process, the class contained multiple attributes such as the data path, the partitions to extract (training, test, validation), the new shape if resizing is required, normalization parameters and multiple flags to control the preprocessing

Fig. 1. Example of the lungs' segmentation in a slice. Top left, result after thresholding. Top right, result after identifying the structures inside the body. Bottom left, result after trachea removal with previous slice. Bottom right, final result with body and lungs mask.

(CLAHE, histogram matching, windowing, etc.) and what was returned for each sample (images, lung masks, body masks). This dataset class was shared across all the different registration methods and gave us the necessary structure to work simultaneously and uniformly on different methods.

D. Registration

In a registration process, two images are involved, the moving one $I_M(x)$ and the fixed one $I_F(x)$. The basic idea is that we want to deform the first one so that it matches the second one. Registration involves finding a transformation $T(x) = x + u(x)$ that makes the image $I_M(T(x))$ spatially aligned to $I_F(x)$. The quality of the alignment is defined by a distance or similarity measure (e.g. mutual information). Commonly, the registration problem is formulated as an optimization one, in which a cost function (opposite to the similarity measure) is minimized with respect to T. In deep learning approaches, this general idea is more or less preserved, but another step is introduced in the pipeline. The transformation map is generated by a deep neural network, whose weights are optimized instead of directly optimizing the transformation parameters.

However, it is important to notice that the transformation $T(x)$ is most commonly defined as a coordinate mapping from the fixed image domain to the moving image domain. If the transformation is defined from moving to fixed image (as the names suggest), not all voxels in the fixed image domain would be mapped to the moving one and holes would occur in the deformed moving image. With the transformation defined from fixed to moving domains, the resampling stage becomes a loop over all voxels x in the fixed image domain, compute for each of them its mapped position $y = T(x)$, interpolate the moving image at y and fill the value at x in the output image.

As stated in previous sections, during the test stage of the challenge, only the inhale landmarks were going to be provided. Due to the fact that registration is done mapping coordinates from fixed to moving in the used algorithms, the choice of how to treat the inhaling and exhaling images was straightforward. Inhaling images were treated as fixed and exhaling ones as moving, for VoxelMorph and Elastix. However, ANTsPy assumes this behavior by default, so the exhaling images were used as fixed and the inhaling ones as moving.

E. Elastix

Elastix [3] is an open-source software package for image registration, consisting of a collection of algorithms that are commonly used to solve medical image registration problems. A large part of the code is based on the Insight Toolkit (ITK). Elastix has an add-on version over SimpleITK library for python. After trying the SimpleITK version of Elastix, we finally decided to keep using the executable binaries provided by Elastix, mainly due to the automatic handling of deprecated parameter map arguments available in the second and not in the first and some stability problems faced with the former. In order to run experiments efficiently, and include Elastix in a broader python pipeline, we programmed a set of wrappers that converted the Elastix shell commands into python functions. Basically three functions were made, one to modify parameter maps files, another one to run Elastix (to run the registration between the desired images) and a last one to run Transformix using the previously obtained transformation map.

The experiments run with Elastix, were conceived in a hierarchical fashion. Across a first round of experiments, we found the best parameter map we could find among the ones provided in the Elastix model zoo. Once we had chosen one, we run a second round of experiments varying some of the most significant hyperparameters in the parameters map file, trying to find a more suitable combination for our problem. Finally, in a third round, with the best parameter map already selected, we decided to explore different pre-processing strategies in order to further improve the registration performance. In the following paragraphs, we describe in detail each step.

1) First Round

On the official Elastix webpage, the developers of this tool provide a collection of parameter maps which were previously found suitable for some specific registration problems of certain body structures. From those, we discarded all the ones not meant for Chest/Lung 3D/4D CT image registration. After reading in detail the description of each parameter map, and in order to reduce the number of experiments, we decided to try three different options identified as Par0008, Par0007 and Par0003.

Par0008 consisted in an intra-patient B-spline transformation using mutual information as cost function. In the model zoo, two parameter maps were provided, an affine registration one and an elastic deformation one. We tried several comparisons, but the method only worked when providing Elastix with the body mask of the patient. We run just registrations using affine and affine+elastic parameter maps.

Par0007 consisted in an intra-patient B-spline transformation using mutual information as cost function. In this parameter map collection, three files were shared corresponding to a *coarse, fine and local rigidity penalty* transformations. Since

the local rigid penalization was not working due to some deprecation over Elastix software, we ended up modifying the metric from MattesMutualInformationWithRigidityPenalty to AdvancedMattesMutualInformation. We run several combinations: Coarse, Coarse+Fine, Coarse+Fine+RP (all of them with no mask); Coarse, Coarse+Fine (using the body mask) and finally Coarse, Coarse+Fine (using the lungs mask).

Par0003 consisted in a non-rigid B-spline transformation. An affine registration was first proposed as an initialization. Then the developers shared the parameter maps for performing the final registration with a Gaussian image pyramid (without downsampling) using resolution levels (R) from 1 to 8. In addition, two alternatives were shared for each value of R. In the first one, the resolution of the B-spline control point grid was kept at a constant value of 12 mm (isotropic) in all resolutions. In the second one (identified as *ug*), the grid was refined after each resolution, such that at the final resolution, the control points were spaced 12 mm apart again. In order to reduce the number of experiments, the following variations were studied. First we run the registration using Affine+B-spline for resolutions in the set $\{1, 2, 4, 6, 8\}$ for the fixed B-spline grid. Then, from the results obtained in those runs, we run Affine+B-spline for resolutions in the set $\{2, 4, 6\}$ with the multi-resolution B-spline grid refinement (ug). Lastly, based on the latest result, we run just B-spline for resolutions in the set $\{4, 5, 6\}$ with ug-grid. All the registrations were run providing Elastix the lungs masks.

2) Second Round

With Par0003 B-Splines R6 ug, being selected as the best parameter one from the model zoo, we explored varying different hyperparameters inside the parameter map file. We did this sequentially, meaning that we tried varying a single parameter at a time, and if it improved the performance we kept it and kept on modifying another one. If the performance got worse, we kept the "default" value of the parameter. In the following paragraph, some words are dedicated to each experiment and the name we gave it can be found in parentheses.

First we modified the default resolution pyramid schedule from smoothing both fixed and moving images by a factor of 2 in each dimension, at each resolution for the following values *ImagePyramidSchedule* = (16 16 4, 16 16 4, 8 8 3, 4 4 2, 2 2 1, 1 1 1) (pyram). This choice was made to take into account the anisotropic nature of the voxels of the provided data. Then, we replaced the Optimizer from *StandardGradientDescent* to *AdaptiveStochasticGradientDescent* (Optim), which automatizes some hyperparameter choices for the SGD method. After this, also to take into account the anisotropy of the voxels, we modified the *FinalGridSpacingInPhysicalUnits* from (12 12 12) to (12 12 4) (Grid) and later to (16, 16, 4) (Grid2). Next, we changed the *MaximumNumberOfIterations* from 1000 to 2000 (n_samples_1) and the *NumberOfSpatialSamples* from 2000 to 10000 (n_samples_2) and later to 15000 (n_samples_3). Finally, we modified the Image Pyramid method (*Moving(Fixed)ImagePyramid*), from "SmoothingImagePyramid" to "RecursiveImagePyramid" (Recursive).

3) Third Round

In the last round of experiments, after finding the best hyperparameters setting, we explored different preprocessing techniques. We evaluated three main options, applying CLAHE, applying windowing as it has already been explained before and no preprocessing done. For each of those three cases, we evaluated 4 normalization options: no normalization (None), min-max normalization using full scan (MM), min-max normalization using lungs mask (MMLM) and min-max normalization using body mask (MMBM).

F. ANTsPy

ANTsPy[1] is a Python wrapper for Advanced Normalization Tools (ANTs)[4] which is a C++ tool, originally built for neuroimaging, that can be used to perform image registration, segmentation and more image processing techniques. In terms of image registration, the package offers multiple transformation models (elastic, diffeomorphic, unbiased) and metrics (cross-correlation, mean squares errors, mutual information...). We implemented two wrappers for using this tool.

The first one was mainly used to register two given volumes (fixed and moving) using ANTsPy with given parameters. The parameters were altered sequentially in order to find the best combination. First, the effect of preprocessing was examined with different settings. Then the best metric to be optimized was found. The metrics we tried were mean squares error (MSE), demons, global cross correlation (GCC), and mutual information (MI). Next, the optimal transformation type was selected from SyN, SyNOnly, Elastic and ElasticSyN. In ANTs, SyN refers to Symmetric normalization which combines an affine and a deformable transformation, SyNOnly performs only a deformable transformation assuming that the fixed and moving volumes are already aligned. Elastic performs an elastic transformation that combines an affine and a deformable transformation. ElasticSyN is the same as SyN with an elastic regularization. Next, the number of neighboring pixels (samples) to be considered was explored. We tested four different values (32, 16, 8 and 4). Finally, we altered the number of iterations and the number of resolution levels of the image pyramid.

The second wrapper function applies a given list of transforms to the moving points given in the physical space according to the LPS orientation (the default for ITK). The corresponding physical space coordinated of the given landmark for each volume were obtained using ANTs' LabelStats command.

Recall that ANTs applies the interpolation method previously explained in section III-D. by default. Therefore, when using ANTs, the images and the points are transformed in inverse directions. So, the forward transforms are used to transform images, whereas inverse transforms are applied to transform landmarks.

G. Deep Learning: VoxelMorph

VoxelMorph [5] is a deep learning deformable-based registration method. Instead of optimizing an objective function between fixed and moving images for each pair of images like Elastix, VoxelMorph aligns the image pair by mapping the input image pair to a deformation field. VoxelMorph tries to optimize the unsupervised objective function:

$$\mathcal{L}_{reg}(f, m, \phi) = \mathcal{L}_{sim}(f, m \circ \phi) + \lambda_1 \mathcal{L}_{smooth}(\phi), \quad (2)$$

[1]https://antspy.readthedocs.io/en/latest/

where ϕ is the registration field, $m \circ \phi$ is the moving image warped by the registration field ϕ. $\mathcal{L}_{sim}$ is a similarity-based function such as Normalized Cross-Correlation (NCC) or Mean Squared Error (MSE), and $\mathcal{L}_{smooth}$ is a regularization over the displacement vector to enforce a spatially smooth deformation.

We used the VoxelMorph with a UNet-based architecture with its default UNet convolutional features [6]. We train our model with varying $\mathcal{L}sim$. We try the MSE and NCC as our similarity loss function $\mathcal{L}sim$. Moreover, we try to use the segmentation mask described in III-C3 to incorporate non-image appearance features. We could then reformulate the unsupervised objective function from Voxel Morph into a semi-supervised loss $\mathcal{L}_{semi}$ as

$$\mathcal{L}_{semi} = \mathcal{L}_{reg}(f, m, \phi) + \lambda_2 \mathcal{L}_{seg}(s_f, s_m \circ \phi), \qquad (3)$$

where $\mathcal{L}_{reg}(f, m, \phi)$ is the unsupervised registration loss described in Eq. 2, and $\mathcal{L}_{seg}(s_f, s_m \circ \phi)$ is the segmentation loss. We chose the dice loss as our segmentation loss $\mathcal{L}_{seg}(s_f, s_m \circ \phi)$. First, we resampled all the input 3D volumes to the same size (256, 256, 128) using spline interpolation of the third order. Then, we normalized each 3D volume between 0 and 1 with the min-max function. We trained each experiment with and without the segmentation mask with different similarity functions for 200 epochs. We used Adam optimizer with a starting learning rate of $1e-3$. We decayed the learning rate by 0.1 factor if the validation loss was not changing with patience of 20 epochs.

We trained the model using the original unregistered dataset, but we also tried to use the deep learning network as a second-stage method after registration with Elastix. Thus, we tried to train VoxelMorph with an Elastix pre-registered dataset. Our motivation for this was based on the fact that deep learning requires large amounts of data to be trained, so prior registered information may easier the task to learn and therefore be helpful for the network to converge faster.

IV. Results

A. Elastix

In figure 2 we can see the results for the first round of experiments. It can be easily seen that parameter map 0003 with the multi resolution B-spline grid refinement (ug) outperforms all the other parameter maps. Even without doing the affine registration stage, the results were even better and obtained in less time than other methods with similar performance.

From figure 3, we can see that the modifications of the parameter map corresponding to experiment *n_samples_2*, led to the best performance. It should be noticed that a small improvement in the performance implied an extra computational cost. Nevertheless, we decided to go for it since the time required was not excessive. The parameter map differed from the "default" one in the following fields:

- *ImagePyramidSchedule* = (16 16 4, 16 16 4, 8 8 3, 4 4 2, 2 2 1, 1 1 1)
- *Optimizer* = AdaptiveStochasticGradientDescent
- *MaximumNumberOfIterations* = 2000
- *NumberOfSpatialSamples* = 10000

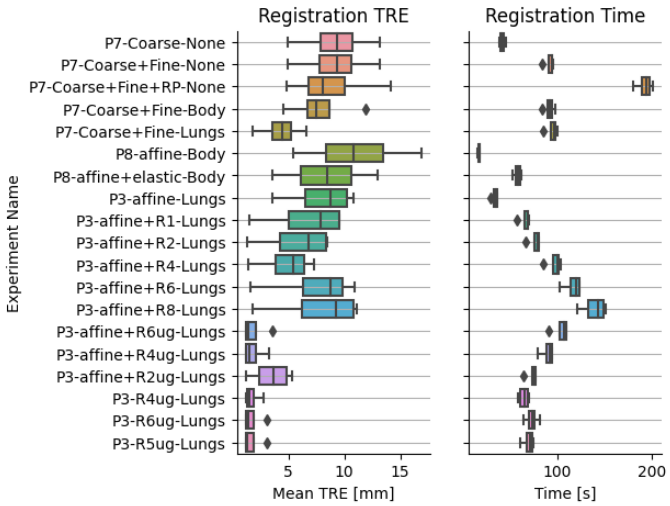

Fig. 2. Boxplots of mean TRE and Time across all training subjects for each Elastix experiment from the first round. In all of them no preprocessing was applied, the experiment names should be interpreted as: Parameter_map-Transformation_kind-Mask_used

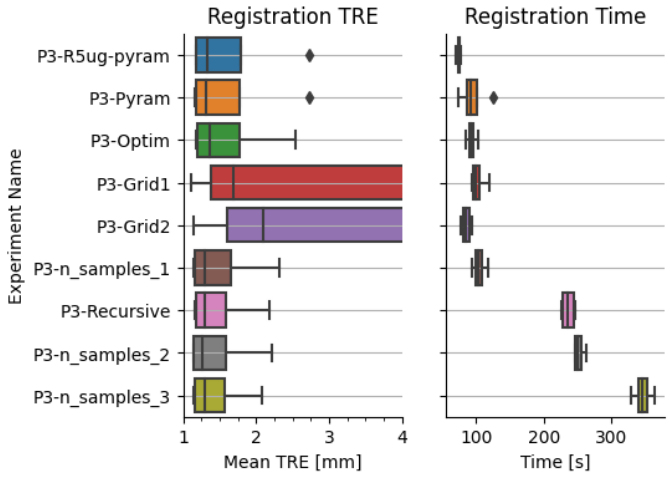

Fig. 3. Boxplots of mean TRE and Time across all training subjects for each Elastix experiment from the second round. In all of them no preprocessing was applied and always the lungs mask was used, the experiment names should be interpreted as: Parameter_map-experiment_name

Finally, as it is shown in figure 4, no improvement was achieved by using any of the pre-processing techniques explored in the experiments. Therefore, the "raw" images are used.

B. ANTsPy

The results of the experiments conducted to find the best combination of parameters for ANTsPy are summarized in figure 5.

Based on our experimental results, we saw that using an affine transformation followed by a deformable one (SyN) achieved the best results in terms of transformation type, mutual information outperformed the other metrics, and the optimal number of samples is 16. The best result was obtained with 5 levels of resolution, with the lowest 2 not being used (0 iterations). That is to say, the image is down sampled twice before applying the registration. We have also noticed that applying preprocessing, by normalizing the intensities and truncating the highest and

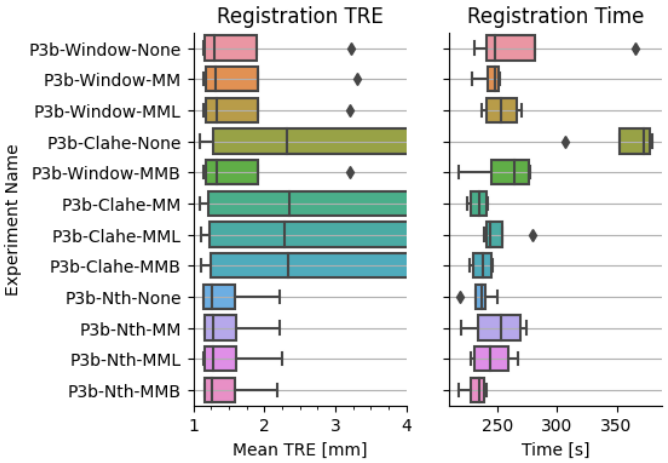

Fig. 4. Boxplots of mean TRE and Time across all training subjects for each Elastix experiment from the third round. In all of them no preprocessing was applied, the experiment names should be interpreted as: Parameter_map-Preprocessing-Normalization.

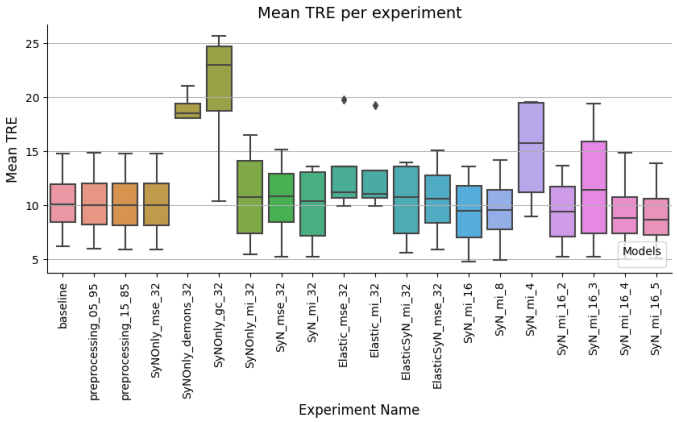

Fig. 5. Boxplots of mean TRE across all training subjects for each ANTsPy experiment.

lowest ones, helped to increase the performance. The best result achieved using this approach had a mean TRE of 9.0825 in a mean time of 197.956 seconds. However, even this result was not comparable to the ones achieved using Elastix; therefore, we did not use it for the challenge.

C. Deep learning

Although the data size was limited, we tried to explore deep learning techniques for CT scan registration. Figure 6 shows the boxplot of the initial displacement (provided TRE) across the 4 trainin subject of the challenge, as well as a LOOCV (Leave One Out Cross-Validation) optimizing $\mathcal{L}_{semi}$. It can be noticed that deep learning succeeded to decrease the TRE, but not in a way that its results could be comparable to the other registration approaches explored. We believe that was caused by the limited amount of data available to optimize the deep learning network parameters. To tackle this issue, we proposed to use the pre-registered images with Elastix for training the deep learning network to output a fine-grained deformation field that could better increase the registration performance.

We show in Figure 7 the boxplot of the LOOCV TRE of different experiments using the above-mentioned approach.

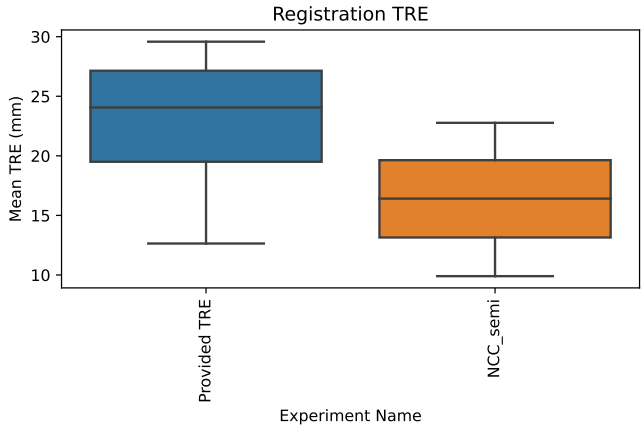

Fig. 6. Boxplot of provided TRE (no registration) and Semi-Supervised VoxelMorph with NCC loss function.

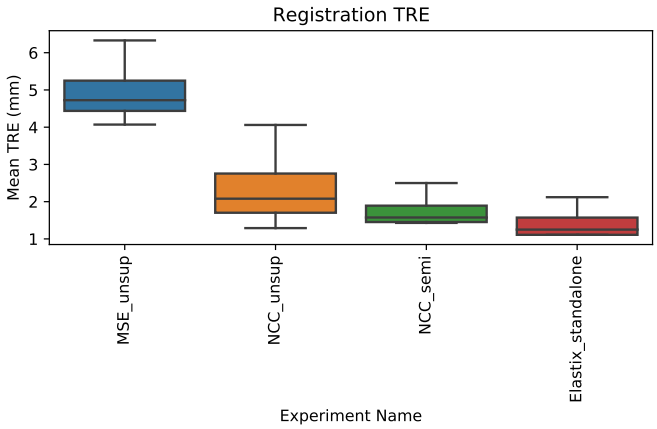

Fig. 7. Boxplots of mean TRE across all training subjects for each of VoxelMorph experiment with Elastix Pre-Registration compared to Elastix stand-alone.

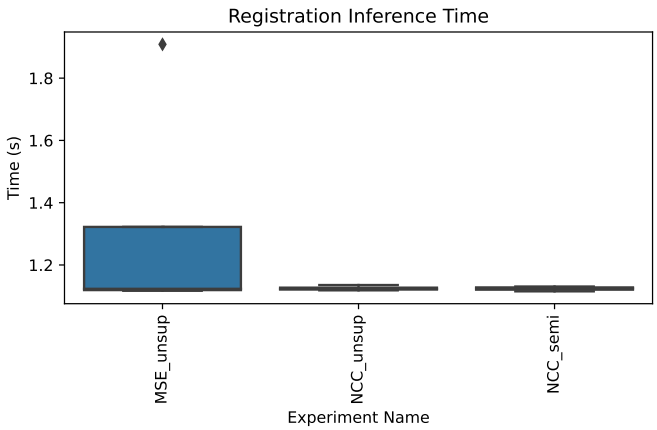

Fig. 8. Boxplots of mean inference time across all training subjects of VoxelMorph.

We can notice that the NCC performs better than MSE, although the MSE has more stable learning as shown in Figure 9. We can notice that using NCC as a similarity-based function performs better than using MSE, this might be because the for-

TABLE I
COMPARISON OF THE BEST THREE EXPERIMENTS OF THE THREE ALGORITHMS EXPLORED.

Method	TRE	Time (seconds)	Training Time (minutes)
LungSegmentation	-	2.94±0.29	0
ANTsPy	9.08±3.18	197.96±34.88	0
VoxelMorph	1.77±0.433	(252.95±7.14) + 1.124±0.006	64.13±8.42
Elastix	1.432±0.413	252.95±7.14	0

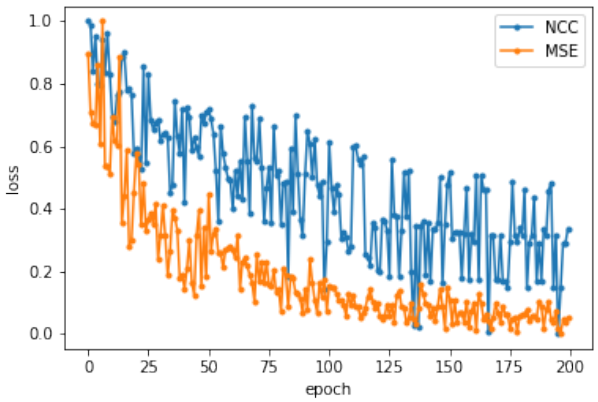

Fig. 9. VoxelMorph normalized training loss over training epoch for different similarity objective functions

mer appeared to suffer from a vanishing gradient problem, with the loss being minimal due to min-max normalization. Besides, NCC seemed to be more robust to overfitting, as its objective function was more complex than the one using MSE. However, NCC required more training rounds to converge. We could also notice that when optimizing $\mathcal{L}_{semi}$, NCC performs better when incorporating prior information (lungs segmentation). To this end, the results of deep learning were far from being comparable to either ANTsPy or Elastix with a large training time as shown in Table II; therefore, we did not use it for the challenge.

TABLE II
TRAINING TIME FOR VOXELMORPH FOR 200 EPOCHS.

Method	Similarity Function	Time (minutes)
Unsupervised	MSE	55.78±9.49
Semi-Supervised	MSE	56.46±7.06
Unsupervised	NCC	62.14±5.98
Semi-Supervised	NCC	64.13±8.42

V. DISCUSSION

Using ANTsPy resulted in a significant reduction of the TRE compared to the initial displacement; however, it remains far from the results achieved using Elastix. Moreover, this method suffered from multiple computational problems, mainly memory errors even with a 16 GB RAM. Running this method on a more powerful device with more iterations and using the higher resolutions of the pyramid together with more developed metrics such as local cross-correlation could further improve the results.

Deep learning failed due to the limited size of the dataset to optimize the deep learning network parameters. For future work, deep learning can benefit from a progressive training scheduling technique for training deformable image registration with small datasets. Either by pre-registering the images with another simple method or by varying data augmentation complexity linearly through training rounds. This scheduling technique may be similar to that of curriculum learning.

VI. CONCLUSION

In this work, we used and compared some of the main proposed methods for medical image registration applied to the use case of lung computed tomography (CT) registration. We compared two classical approaches, namely Elastix and ANTsPy with a novel deep learning approach, VoxelMorph. After exploring several configurations, variations and preprocessing techniques for each method, Elastix proved to outperform the other methods.

REFERENCES

[1] R. Castillo, E. Castillo, D. Fuentes, M. Ahmad, A. M. Wood, M. S. Ludwig, and T. Guerrero, "A reference dataset for deformable image registration spatial accuracy evaluation using the copdgene study archive," *Physics in Medicine Biology*, vol. 58, no. 9, p. 2861, apr 2013. [Online]. Available: https://dx.doi.org/10.1088/0031-9155/58/9/2861

[2] J. Hofmanninger, F. Prayer, J. Pan, S. Röhrich, H. Prosch, and G. Langs, "Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem," *European Radiology Experimental*, vol. 4, no. 1, 2020.

[3] S. Klein*, M. Staring*, K. Murphy, M. A. Viergever, and J. P. Pluim, "elastix: a toolbox for intensity-based medical image registration," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 196 – 205, January 2010.

[4] B. B. Avants, N. Tustison, G. Song *et al.*, "Advanced normalization tools (ants)," *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.

[5] G. Balakrishnan, A. Zhao, M. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," *CVPR: Computer Vision and Pattern Recognition*, pp. 9252–9260, 2018.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.