# MIRA Final Report on : Image Registration of Chest CT Volumes using Elastix and VoxelMorph

**Anwai Archit and Sheikh Adilina**
University of Girona

## 1 Abstract

The Registration of Medical Images is one of the most important preliminary step which helps with straight-forward diagnosis for clinicians from the registered images, also the quality of registration has strong correlation with the algorithms further developed on top of the registered images to support Computer Aided Diagnosis in Medical Imaging and Applications. The inhale and exhale pair of CT Images and respective landmark points have been provided from the National Heart Lung Blood Institute COPDgene study. We have built two pipelines from scratch. In the first pipeline, we have used Affine, B-Spline and Combined Affine and B-Spline Registration Parameters on both segmented and original pairs of Lung CTs using Elastix. Our best results were achieved using B-Spline Registration on segmented lung CT masks using Elastix. Initially, we begun with a novel Deep Learning Approach, named VoxelMorph. It was applied on the entire 8 samples of the provided dataset. By using VoxelMorph, we could somehow better the TRE although with slight betterment in the visual registration. We however tried another pipeline using segmentations, and using preregistered images through the deep learning pipeline to avoid the misaligned registrations of lung nodules in the volumes. The best results with VoxelMorph were achieved using B-Spline on segmented masks. We definitely achieved robust results and encouraging enough to endorse Deep Learning, considering if we had more amount of data.

## 2 Introduction

Image Registration (in general words) is stated as the methodology to align the images in reference to a fixed frame of reference by using global transformations to the pixels in the volumes. Here, if the method is applied to all the pixels of the volumes together, i.e, Rigid Transformation or local transformations applied to pixel-wise or region-wise, i.e, Non Rigid Transformation. For instance, usually the Rigid Transformations are the baseline strategies before Non Rigid Transformations to verify using the global alignment before implementing the local-level pixel-wise transformations. This clearly avoids using heavy deformations and eliminating the unreasonable transformations cause by non rigid transformations.

The aim of our project was to :

1. Register the 3D CT Lung Volumes

2. Transform the Landmark Points, Minimize the Mean and Standard Deviation between the Inhale and Exhale Landmarks and Evaluate using TRE

## 3 Materials and Methods

### 3.1 Dataset description

We were provided with 8 cases of COPDgene dataset from the DIR-Lab Challenge. Each case contained two 3D CT Lung volumes of exhale and inhale. Each cases also contained 2 text files containing coordinate information for the reference landmarks on the inhalation (iBH-CT) and exhalation (eBH-CT) breath-hold CT images. Detailed information about each of the cases are given in Table : 1.

Table 1: The COPD dataset

| Label | Image Dimension | Voxels (mm) | Displacement (mm) : Mean, Std |
|---|---|---|---|
| COPD1 | 512 x 512 x 121 | 0.625 x 0.625 x 2.5 | 25.90 (11.57) |
| COPD2 | 512 x 512 x 102 | 0.645 x 0.645 x 2.5 | 21.77 (6.46) |
| COPD3 | 512 x 512 x 126 | 0.652 x 0.652 x 2.5 | 12.29 (6.39) |
| COPD4 | 512 x 512 x 126 | 0.590 x 0.590 x 2.5 | 30.90 (13.49) |
| COPD7 | 512 x 512 x 112 | 0.625 x 0.625 x 2.5 | 21.66 (7.66) |
| COPD8 | 512 x 512 x 115 | 0.586 x 0.586 x 2.5 | 25.57 (13.61) |
| COPD9 | 512 x 512 x 116 | 0.664 x 0.664 x 2.5 | 14.84 (10.01) |
| COPD10 | 512 x 512 x 135 | 0.742 x 0.742 x 2.5 | 22.48 (10.64) |

### 3.2 Preprocessing

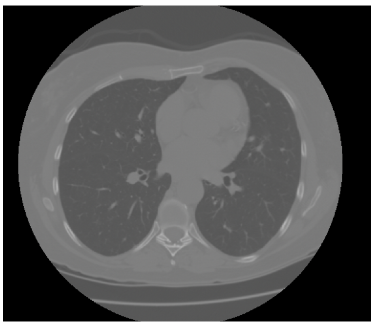
The COPD images given to us were in raw img format. We were provided with "MatlabUtilityPack1 - v1.0" which contained Matlab files to view the raw images and the landmark points superimposed on them.

#### 3.2.1 Conversion to NIFTI Format

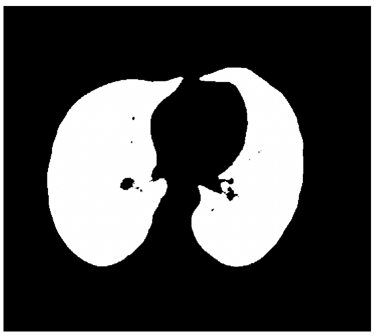
For our convenience, we wanted to work with NIFTI format so we used ITK-Snap [8] to manually convert the images with the help of the COPD dataset information the aforementioned table. While conversion, we were careful with the orientations. For each of the lung volume we had to change the orientation of the Z-axis from "Inferior to Superior" to "Superior to Inferior".

#### 3.2.2 Segmentation

Amidst trying our methods, we felt the need of the registration process to be localised on the lung nodules of different lobes and oblique fissures within the lung volumes. There is a chance where the segmented lungs primarily on the lung nodules within the CT Volumes. The Lung Segmentation Pipeline [6] is built using thresholding, region growing after removing the borders to help seed placement and further postprocessing using morphological operations to obtain the desired mask.



(a) Original Image



(b) Segmented Image

Figure 1: Segmentation sample of COPD4 instance

### 3.3 Traditional Method - Elastix-Based Registration

For the traditional method, we have used elastix [5] [7] to register the images and then used transformix to transform the moving images as well as the provided landmark points. We used the parameters files from Par0049 - elastix as a baseline for the parameters we have chosen to perform the registration. We relied on this particular model because it was specifically applied on the datasets from the DIR-Lab. This model had multiple registration settings. We kept the same values as the components that were common in all the parameter files. We mostly explored the components which were different in their different parameter files eventually selecting the parameters that worked best for our data.

#### 3.3.1 Registration

For our baseline transformation, we explored three different transformations, affine, bspline and the combination of both.

1. **Affine** can be termed as non rigid transformation. It used a combination of rotation, translation, reflection and shear mapping to perform global registration.
2. **B-Spline** Registration can also be termed as a method of deformable registration that uses B-Spline curves to define a continuous deformation field that maps each and every voxel in a moving image to a corresponding voxel within the reference image.
3. **Combined** - We got the idea of combining Affine and B-Spline from the Par0003 - elastix from the Model Zoo of elastix. Even though on a different dataset but this too was applied on interpatient 3D Lung CT scan just like our dataset. We kept all the parameters same as the B-Spline parameter and changed only the transform to Affine.

We used the **Multi Resolution Registration**. We tested with numerous grid starting with the default value 16.0 16.0 16.0 because the **Final Grid Spacing** in the parameter file was not in voxel. Based on our experiments and rationally since the voxel spacing in our dataset has a ratio of 1:1:4, we chose 16.0 16.0 4.0 as our final setting (shown in Fig: 2). We also experimented with downsampling patterns, ultimately deciding to keep the **Number Of Resolutions** 5 with **Image Pyramid Schedule** [16 16 4, 8 8 3, 4 4 2, 2 2 1, 1 1 1]. We chose the **Recursive Image Pyramid** option since it is quick and smooths out the image while downsampling.

#### 3.3.2 Optimizer

**Adaptive Stochastic Gradient Descent** was used as optimizer as it does the optimization automatically. However, we tuned the SP_a parameter which has a default parameter of 20 but from our experiments we found out that keeping the SP_a at 50 gave us lower mean TRE (shown in Fig: 3) for all the instances so we kept the value at 50. We also increased the **Maximum Number Of Iterations** to 2000 to allow the algorithm more iterations for optimization. The **Adaptive Step Sizes** was set to "true" as it makes the registration more robust.

#### 3.3.3 Metrics

Advanced Mattes Mutual Information and Advanced Normalized Correlation are the two most often used metrics for Lung CT scans, thus we explored both metrics and found that **Advanced Mattes Mutual Information** performed better. Our experimental results are shown in Fig: 4.

#### 3.3.4 Sampler and Interpolator

The order of B-Spline interpolation used in each resolution level was set to 1 and the Order of B-Spline interpolation used for applying the final deformation was set to 3.

For the Image Sampling, we carried out experiments using different values for **Number Of Spatial Samples** (shown in Fig: 5) and decided to keep 10000 as value as it gave the best Mean TRE with the least standard deviation.We kept the **Use Direction Cosines** as true as setting this setting to true means the important information from the image relating to voxel coordinates to world coordinates are taken into account. We also set the **Erode Mask** setting to false so that information from the border/edge of the mask is taken into account.

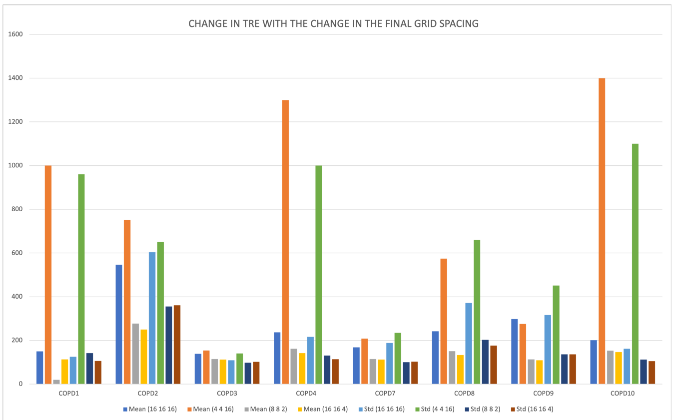



Figure 2: The change in the value of TRE with the change in the value of Final Grid Spacing

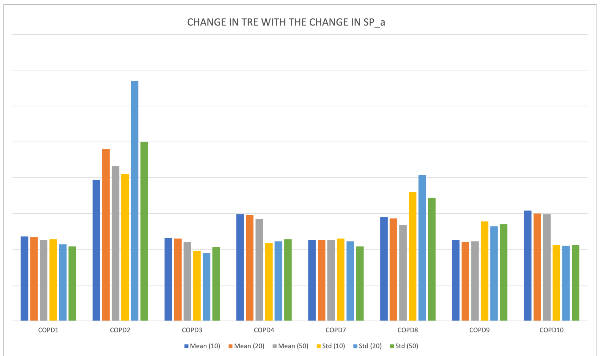



Figure 3: The change in the value of TRE with the change in the value of SP_a

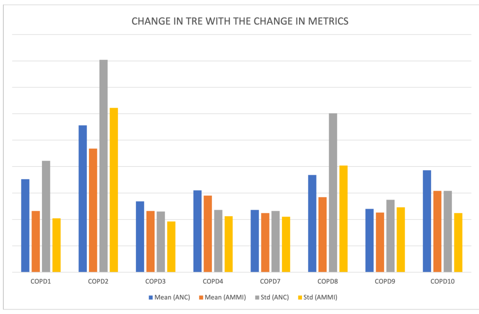



Figure 4: The change in the value of TRE with the change in Metrics

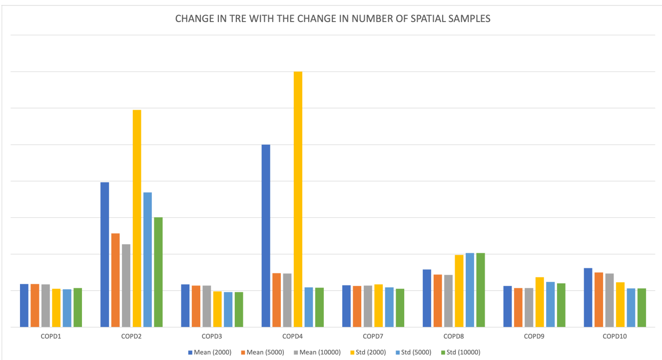



Figure 5: The change in the value of TRE when the Number Of Spatial Samples is changed

### 3.4 VoxelMorph

VoxelMorph [2] is a learning based method which uses Convolutional Neural Networks (CNN) to perform registration. Given two images fixed ($f$) and moving ($m$), the deep learning model ($g(m, f)$) learns the deformation field ($\phi$), shown in Fig: 7. It would have been extremely easy for the model to calculate the loss function if we had the ground truth of the deformation field. But it is not possible for a person to manually assign the x and y deformation of every single pixel. So, VoxelMorph optimizes its network by generating deformation field that maximises the matching of the images and at the same time making sure the field is as smooth as possible. The equation is given in Fig: 6

$$\mathcal{L} = \underbrace{\|m \circ \phi - f\|}_{\text{images match}} + \lambda \underbrace{\mathrm{Reg}(\phi)}_{\text{smooth field}}$$

Figure 6: VoxelMorph Equation

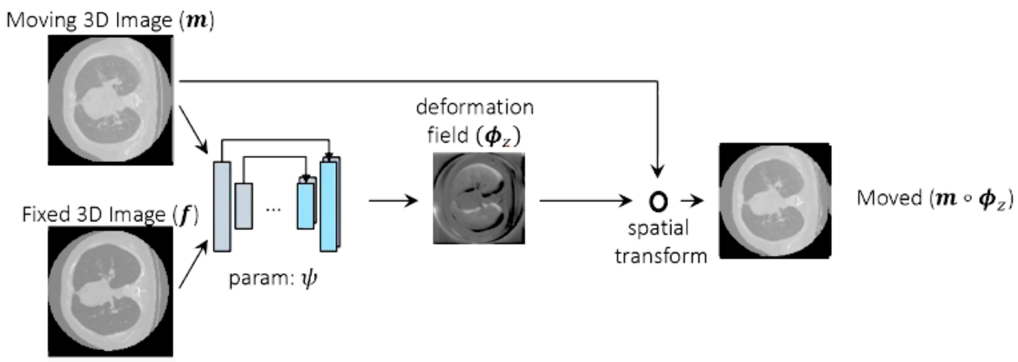



Figure 7: VoxelMorph Architecture

VoxelMorph currently has a U-Net like architecture but researchers are currently working on changing it to transformer architecture. The exact architecture that we used in shown in Fig: 8. Our model has four layers of encoder and four layers of decoder with several 32-filter convolutions. The skip connections are shown using arrows.

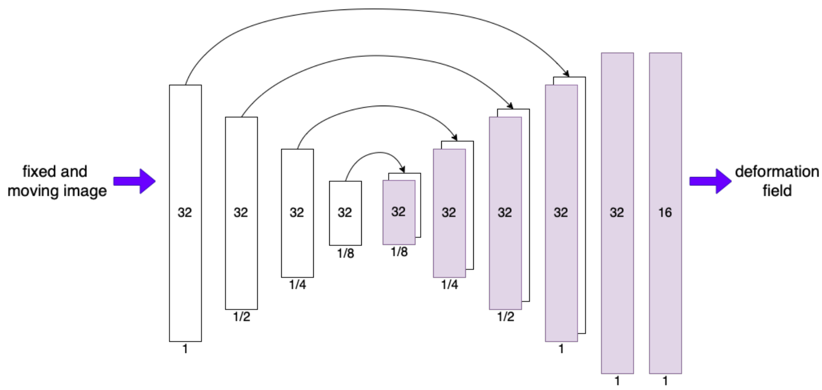



Figure 8: Architecture of VoxelMorph

## 4 Results and discussion

In this section, we will describe the details of the experiments done in this paper.

All of the VoxelMorph variations were ran using Colab (Pro). Colab Pro has the Intel(R) Xeon(R) CPU @ 2.30GHz and GPU of Tesla P100-PCIE-16GB. Colab Pro also has two RAM options with the standard option of 12.80GB and High RAM option which goes upto 26.30GB. We ran all our experiments using GPU with the High RAM option.

The traditional registration (elastix and transformix) were done using Apple M1 Pro chip which contains a 10-core CPU with 16GB RAM.

Lastly, **Target Registration Error** (TRE) was the metric that we used to evaluate the performance of the registration.

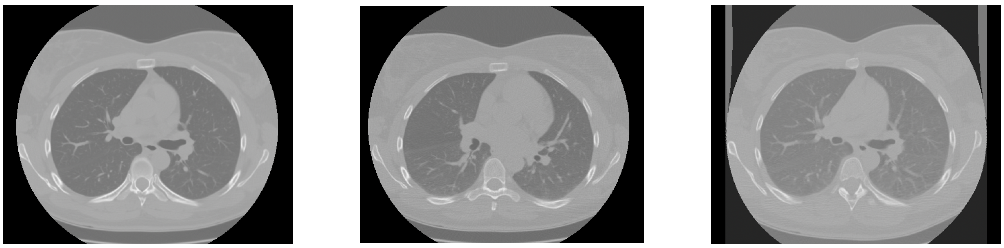
### 4.1 Building our Pipeline

We were told that we would have just 1 hour to submit the transformed landmark points on the Challenge Day, so we automated a lot of our tasks by building python scripts to perform the repetitive tasks that we would have to do manually otherwise. To generate the folders, we used separate python programs. We also wrote python scripts to move and rename files to the relevant folder so that the files are ready for the VoxelMorph code. Moreover, to avoid any errors and to save the unnecessary process of putting the commands in the command line one after another, we generated separate text files containing all the elastix and transformix commands for all the COPD cases. This proved to be very efficient on the Challenge Day because we were able to make our submission in approximately 30 minutes.

### 4.2 Traditional Method

For all the traditional methods, we have carried out separate experiments with and without masks. Table: 2 and Table: 3 show the results in mean TRE and also the standard deviation in brackets. Each row represents a single COPD case from the dataset. The second column is the TRE value before registering the images. The results of using combination of Affine and B-Spline are in the third column. The results of Affine and B-Spline are in the last two columns.

Combination of Affine and B-Spline registration took on an average **3 minutes and 19 seconds for each instance**. The B-Spline registration took around **2 minutes and 50 seconds** while the Affine registration took approximately **1 minute and 7 seconds**. For majority of the cases, B-Spline registration performs better than the other traditional methods. We can also see a significant drop in the Mean TRE when mask is added during the registration. Hence, we select **B-Spline with mask as our best traditional method**. Fig: 9 shows the registration of an axial slice taken from the COPD7 instance.



(a) Fixed Image (b) Moving Image (c) Moved Image

Figure 9: Registration of COPD7 using the best traditional method

### 4.3 VoxelMorph

VoxelMorph has been written in Keras [3] with a TensorFlow [1] back end. To implement VoxelMorph, we took the help of the code provided in the VoxelMorph tutorial from the VoxelMorph GitHub Repository. VoxelMorph takes around **37 minutes** to train using the Colab Pro GPU in High RAM. We trained the VoxelMorph with the first 7 training data and kept the last data (COPD10) as a test instance to evaluate the performance of the model.

Since VoxelMorph optimizes its network solely from the images and not from any ground truth, the authors of VoxelMorph showed that the algorithm performs better if it is forced to focus on a particular area. Professor Manolis Kellis mentioned in one of his lectures [4] that adding segmentation maps during the training process enhances the performance of VoxelMorph by almost 10% (shown in Fig: 10) which is a significant improvement in performance. Hence we decided to train VoxelMorph using segmented lung volume. Table: 4 shows the quantitative results of VoxelMorph on the COPD dataset with and without segmentation. However, in this case VoxelMorph with segmentation was not able to perform better than VoxelMorph without segmentation. A crucial reason could be the lack of data. VoxelMorph requires at least 10 volumes to be able to perform proper learning and we were training it with just 7 instances which is already below the required number of cases, adding segmentation further reduces its scope of learning the deformation field.

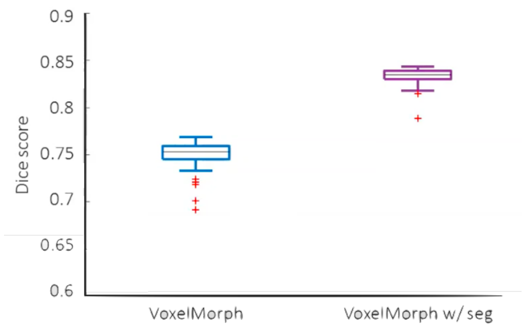
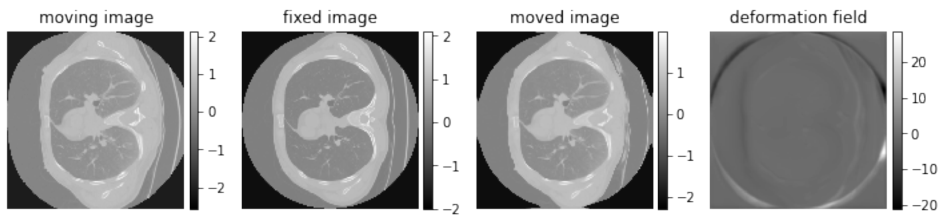
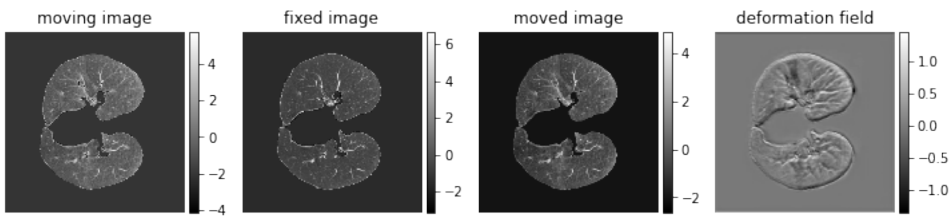



Figure 10: Performance of VoxelMorph with and without segmentation

We also wanted to compare how VoxelMorph performs when it is trained on already registered data. From Tables 5, 6 and 7, it can be seen that VoxelMorph was not able to outperform any of the traditional methods. Compared to all the results obtained, the best performing VoxelMorph model was the one where we combined our best traditional method (B-Spline) with VoxelMorph so we decided to also show some qualitative results of the best VoxelMorph model. The fixed, moving, moved images along with the deformation field are shown in Fig: 11. Fig: 12 shows the deformation field and the registered image side by side.

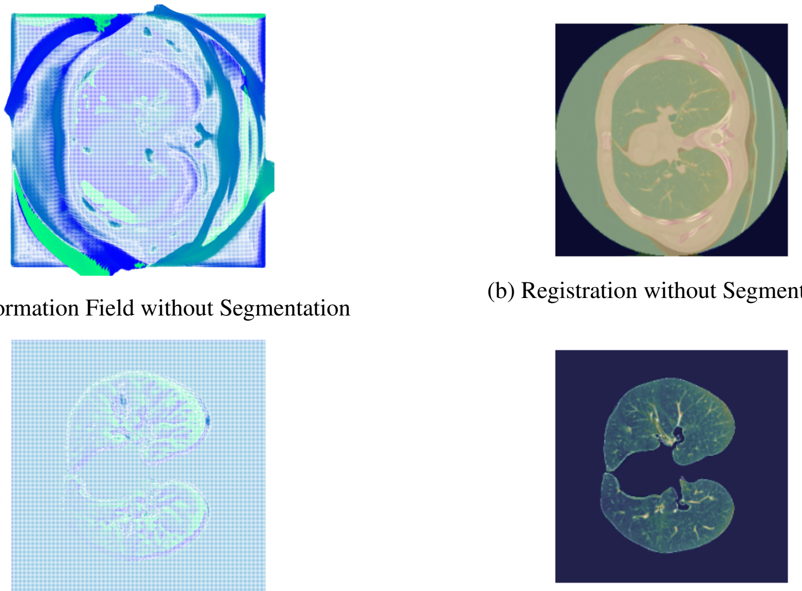



(a) Without Segmentation




(b) With Segmentation

Figure 11: Performance of VoxelMorph on COPD10



(a) Deformation Field without Segmentation

(b) Registration without Segmentation

(c) Deformation Field with Segmentation

(d) Registration with Segmentation

Figure 12: Deformation Field predicted by VoxelMorph and the corresponding Registration for COPD10

Table 2: Experimental Results of Registration using Elastix Based Registration with Mask

| **Volume** | **Before Registration** | **Affine and B-Spline** | **Affine** | **B-Spline** |
|---|---|---|---|---|
| COPD1 | 26.15 (11.34) | 1.16 (1.02) | 11.70 (6.63) | **1.13** (1.04) |
| COPD2 | 21.64 (6.42) | 2.34 (3.11) | 9.66 (5.35) | **2.16** (2.50) |
| COPD3 | 12.56 (6.37) | 1.16 (0.96) | 3.62 (1.99) | **1.10** (1.03) |
| COPD4 | 29.58 (12.92) | 1.45 (1.06) | 7.36 (3.68) | **1.42** (1.14) |
| COPD7 | 21.55 (7.72) | **1.12** (1.05) | 6.49 (3.87) | 1.13 (1.04) |
| COPD8 | 26.36 (13.23) | 1.42 (2.02) | 8.34 (6.48) | **1.34** (1.72) |
| COPD9 | 14.82 (9.80) | 1.13 (1.23) | 8.28 (7.67) | **1.11** (1.35) |
| COPD10 | 21.79 (10.49) | 1.54 (1.12) | 11.47 (4.79) | **1.49** (1.06) |

Table 3: Experimental Results of Registration using Elastix Based Registration without Mask

| **Volume** | **Before Registration** | **Affine and B-Spline** | **Affine** | **B-Spline** |
|---|---|---|---|---|
| COPD1 | 26.15 (11.34) | 5.79 (5.24) | 25.71 (11.19) | **5.49** (5.11) |
| COPD2 | 21.64 (6.42) | 10.13 (6.62) | 23.38 (5.41) | **9.62** (6.48) |
| COPD3 | 12.56 (6.37) | 3.40 (3.23) | 7.27 (3.37) | **3.33** (3.32) |
| COPD4 | 29.58 (12.92) | 9.35 (5.45) | 22.67 (9.55) | **7.17** (4.98) |
| COPD7 | 21.55 (7.72) | **3.56** (3.00) | 11.84 (5.56) | 3.64 (3.22) |
| COPD8 | 26.36 (13.23) | **6.08** (4.72) | 19.46 (8.70) | 6.37 (4.94) |
| COPD9 | 14.82 (9.80) | **3.44** (3.28) | 12.92 (5.51) | 3.56 (3.43) |
| COPD10 | 21.79 (10.49) | 19.72 (10.42) | 30.70 (15.12) | **9.37** (7.27) |

Table 4: Experimental Results of Registration using VoxelMorph

| Volume | Before Registration | VoxelMorph | VoxelMorph with Segmentation |
|---|---|---|---|
| COPD1 | 26.15 (11.34) | **21.75** (9.84) | 26.78 (13.10) |
| COPD2 | 21.64 (6.42) | 21.00 (6.79) | **20.56** (6.27) |
| COPD3 | 12.56 (6.37) | **11.12** (5.46) | 12.05 (7.09) |
| COPD4 | 29.58 (12.92) | **23.48** (9.07) | 26.85 (12.02) |
| COPD7 | 21.55 (7.72) | **16.26** (6.46) | 22.90 (13.33) |
| COPD8 | 26.36 (13.23) | **18.75** (7.94) | 21.10 (10.17) |
| COPD9 | 14.82 (9.80) | **11.44** (6.42) | 17.47 (13.79) |
| COPD10 (test) | 21.79 (10.49) | **17.16** (8.31) | 19.41 (9.12) |

Table 5: Experimental Results of Registration using VoxelMorph and Combine Registration

| Volume | Traditional Registration | VoxelMorph with Registration | VoxelMorph with Registration and Segmentation |
|---|---|---|---|
| COPD1 | **1.16** (1.02) | 2.08 (1.31) | 1.41 (1.04) |
| COPD2 | **2.34** (3.11) | 2.40 (2.88) | 2.61 (2.97) |
| COPD3 | **1.16** (0.96) | 1.27 (0.78) | 1.39 (0.96) |
| COPD4 | **1.45** (1.06) | 2.78 (1.71) | 1.82 (1.22) |
| COPD7 | **1.12** (1.05) | 1.25 (0.94) | 1.59 (1.07) |
| COPD8 | **1.42** (2.02) | 1.55 (1.93) | 1.75 (2.05) |
| COPD9 | **1.13** (1.23) | 1.26 (1.04) | 1.42 (1.24) |
| COPD10 (test) | **1.54** (1.12) | 1.58 (0.92) | 1.77 (1.14) |

Table 6: Experimental Results of Registration using VoxelMorph and Affine Registration

| Volume | Registration Only | VoxelMorph with Registration | VoxelMorph with Registration and Segmentation |
|---|---|---|---|
| COPD1 | 11.70 (6.63) | 13.90 (10.23) | **11.40** (6.79) |
| COPD2 | 9.66 (5.35) | **9.04** (5.19) | 10.29 (6.39) |
| COPD3 | **3.62** (1.99) | 3.79 (2.21) | 3.87 (2.31) |
| COPD4 | **7.36** (3.68) | 8.85 (3.79) | 8.18 (4.38) |
| COPD7 | **6.49** (3.87) | 6.88 (4.29) | 7.17 (4.44) |
| COPD8 | **8.34** (6.48) | 8.43 (6.53) | 9.55 (6.92) |
| COPD9 | 8.28 (7.67) | **7.95** (7.13) | 8.29 (7.26) |
| COPD10 | **11.47** (4.79) | 11.61 (5.93) | 11.57 (5.54) |

Table 7: Experimental Results of Registration using VoxelMorph and B-Spline Registration

| Volume | Registration Only | VoxelMorph with Registration | VoxelMorph with Registration and Segmentation |
|---|---|---|---|
| COPD1 | **1.13** (1.04) | 2.41 (1.75) | 1.30 (1.01) |
| COPD2 | **2.16** (2.50) | 2.34 (2.53) | 2.30 (2.41) |
| COPD3 | **1.10** (1.03) | 1.32 (0.97) | 1.28 (1.03) |
| COPD4 | **1.42** (1.14) | 1.79 (1.13) | 1.64 (1.16) |
| COPD7 | **1.13** (1.04) | 1.39 (1.03) | 1.35 (1.03) |
| COPD8 | **1.34** (1.72) | 1.63 (1.68) | 1.50 (1.70) |
| COPD9 | **1.11** (1.35) | 1.43 (1.24) | 1.31 (1.32) |
| COPD10 | **1.49** (1.06) | 1.54 (0.90) | 1.67 (1.06) |

## 5 Conclusion

To conclude, the best performing registration model was the B-Spline model. Not only was it able to achieve the best Mean TRE on COPD5 and COPD6 respectively, it also proved to be robust as it performed very well on the COPD0 data on the Challenge Day allowing us to become the winner. For the future work and possible improvements, we plan to involve addition of dataset in the best VoxelMorph Blended Registration Pipeline. We can definitely endorse Deep Learning for Registration Applications keeping in mind the availability of plethora of data on top of some magic elements of segmentation and classical transformations.

## References


[1] Martín Abadi and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. Voxelmorph: A learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38(8):1788–1800, Aug 2019.

[3] Francois Chollet et al. Keras, 2015.

[4] Manolis Kellis. Computation systems biology : Deep learning in life sciences. *MIT*, 2021.

[5] Stefan Klein*, Marius Staring*, Keelin Murphy, Max A. Viergever, and Josien P.W. Pluim. elastix: a toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196 – 205, January 2010.

[6] Nihad Meanović, Mislav Grgic, Haris Huseinagić, Matija Males, Emir Skejic, and Muamer Smajlovic. Automatic ct image segmentation of the lungs with region growing algorithm. 2011.

[7] Denis P Shamonin, Esther E Bron, Boudewijn P.F. Lelieveldt, Marion Smits, Stefan Klein, and Marius Staring. Fast parallel image registration on cpu and gpu for diagnostic classification of alzheimer's disease. *Frontiers in Neuroinformatics*, 7(50):1–15, January 2014.

[8] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.