An Optimized Volumetric Approach to Unsupervised Image Registration

Naseem Alsadi*[a], Waleed Hilal[a], Onur Surucu[a], Alessandro Giuliano[a], Stephen A. Gadsden[a], John Yawney[b]

[a]McMaster University, 1280 Main St W, Hamilton, ON L8S 4L8 ; [b]Adastra Corporation, 8500 Leslie St #600, Thornhill, ON L3T 7M8

ABSTRACT

Medical image analysis continues to evolve at an unprecedented rate with the integration of contemporary computer systems. Image registration is fundamental to the task of medical image analysis. Traditional methods of medical image registration are extremely time consuming and at times can be inaccurate. Novel techniques, including the amalgamation of machine learning, have proven to be fast, accurate and reliable. However, supervised learning models are difficult to train due to the lack of ground truth data. Therefore, researchers have endeavoured to explore variant avenues of machine learning, including the implementation of unsupervised learning. In this paper, we continue to explore the use of unsupervised learning for the task of image registration across medical imaging. We postulate that a greater focus on channel-wise data can largely improve model performance. To this end, we employ a sequence generation model, a squeeze excitation network, a convolutional neural network variation of long-short term memory and a spatial transformer network for a channel optimized image registration architecture. To test the proposed approach, we utilize a dataset of 2D brain scans and compare the results against a state-of-the-art baseline model.

Keywords: Medical Image Registration, Medical Image Analysis, Unsupervised Learning

1. BRIEF INTRODUCTION

Contemporary medical image analysis techniques have paved the way for its continuous widespread integration within the greater scope of the healthcare process. At the fundamental layer, medical image analysis can be framed as the extraction of quantitative information, with a clear and specific goal in mind, from one or more medical images. With rapid advancements in the technological paradigm, efficiently processing vast sums of data has become considerably more manageable. These advancements have enabled the integration of computer-aided image interpretation in routine clinical practice, spanning across several domains and applications. This in turn has replaced the tedious and time-consuming traditional practice procedure entailing manual measurements being conducted by human experts.

Medical image analysis, however, commonly deals with the quantification of specific geometric features and the assessment of anatomical changes over time. Medical image analysis is highly application-specific, requiring thorough design and formulation prior to practical implementation [1]. Nonetheless, there are two core tasks in the domain of medical image analysis, namely, image segmentation and registration. Image segmentation involves the detection of objects of interest while drawing and establishing object boundaries. Image segmentation can be a prerequisite for numerous tasks where the identification of geometric properties is necessary, such as the definition of object shape and texture, including registration [2]. Image registration involves the realization of spatial relationships between numerous images. Establishing spatial correspondence across various images allows for image complementation [3]. Thus, variant images taken from different temporal or spatial alignment, or even variant modalities, can be effectively utilized to provide a holistic depiction of the object/region of interest.

Medical image registration aims at resolving the variability between medical images introduced by the utilization of variant imaging modalities, temporal differences, or varying subjects. A study that aims at examining the change across numerous brain images may find it extremely challenging when provided with images taken with two or more varying modalities, such as Computerized Tomography (CT) and Magnetic Resonance Imaging (MRI). In addition, images of the same participant may have been taken during different times, causing increased variability between the proposed images. Evidently so, the task of achieving spatial correspondence between the images, or image registration, is vital to the central thesis of medical image analysis [4].

Big Data IV: Learning, Analytics, and Applications, edited by Fauzia Ahmad,
Panos P. Markopoulos, Bing Ouyang, Proc. of SPIE Vol. 12097, 120970D
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2618647

The process of medical image registration begins with the introduction of two, in the case of pair-wise registration, or more, in the case group-wise registration, medical images with a common region of interest. One of the images is referred to as the moving or source and the others are referred to as the target, fixed or sensed image. Image registration involves spatially transforming the source/moving image(s) to align with the target image. The image registration procedure can be broken down into three separate components. The first of which is the identification of similarity between the proposed images, completed with the assistance of a similarity or dissimilarity measure. Transformation is then applied to one of the selected images with the aim of achieving spatial correspondence with its image pair. However, a wide variety of transformations can be applied to the image, therefore the algorithmic process involves the estimation of the optimal transformation, which when employed maximizes or minimizes the similarity or dissimilarity, respectively [5].

2. PROPOSED METHODOLOGY

2.1 Problem Statement

The registration problem necessities the input of two images, a moving image, I_M, and a fixed image, I_F. The goal of the network is to establish spatial correspondence between the two input images through the application of a transformation, T, on the moving image, therein producing a moving image, I_D [6].

$$I_D = T(I_M) \tag{1}$$

2.2 Optimization Problem

To solve the problem of unsupervised image registration we can take an optimization-based viewpoint. Simply stated, the goal of the model will be to utilize a similarity metric, E, to iteratively compare the moved image, I_D, to the respective fixed image, I_F. At each iteration, the model will aim to make the necessary updates to the learning parameters, such that the value of the similarity metric is reduced from the previous iteration. The completion flag of the registration procedure is invocated at the arrival of a set number of epochs or predefined similarity metric output value, $\mathcal{L}_{sim_{min}}$ [7].

$$T_{optimal} = optimum\left(\mathcal{L}_{\text{sim}}(I_{\text{F}} - \text{I}_{\text{D}})\right) = optimum\left(\mathcal{L}\left(I_{\text{F}} - T\,(I_{\text{M}})\right)\right) \tag{2}$$

Note that the proposed methodology is unsupervised in nature. Therefore, no 'true' optimal transform is provided to the model. In theory, a random deformation can be applied to I_M, therein producing an I_F with a known transformation. However, with aim of generating practical results that can be truly compared with a state-of-the-art baseline, we do not utilize this approach. Rather, I_M and I_F are images sampled from a dataset where the 'true' transformation is not known apriori.

2.3 Overview

The model architecture can be decomposed into three variant sequential phases, namely the sequence generator, registration field prediction and transformation. The architecture draws inspiration from state-of-the-art unsupervised image registration algorithm, VoxelMorph, however, differs heavily in processes leading to the production of the pixel-oriented registration field and application of the transformation [8]. The proposed approach relies heavily on channel-wise features and exploits their interdependencies.

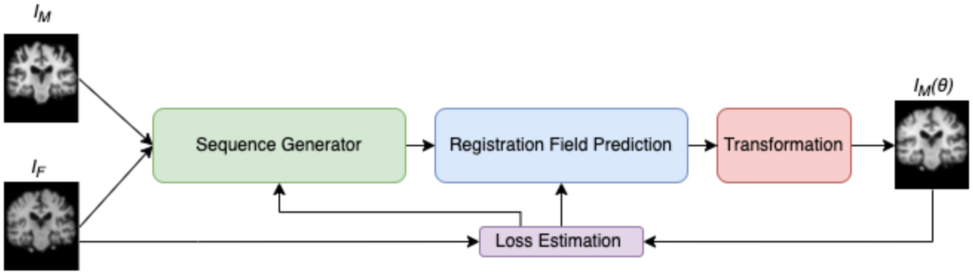


Figure 1: Cumulative Model Overview

2.4 Squeeze and Excitation

The fundamental building block of the network is the squeeze and excitation block, which is depicted in Figure 2. The goal of the proposed model is to utilize channel interdependencies throughout the architecture, regardless of if they are input channels or feature map channels. The squeeze and excitation network enables this by using a parallel path and a skip connection [9].

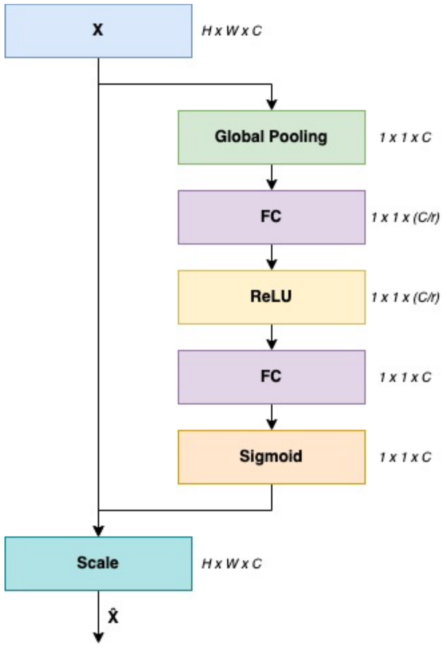


Figure 2: Squeeze and Excitation Network

The parallel path will squeeze the input, with the dimensionality of $H \times W \times C$, into a dimensionality of $1 \times 1 \times C$, where C is the number of channels in the input image. This is conducted with the employment of global average pooling, although more complex aggregation methods may be used. The process is formulated as:

$$z_c = F_{sq}(X_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i,j) \tag{3}$$

A gating mechanism is then employed to capture channel-wise dependencies. Two fully connected layers are utilized to limit model complexity while enhancing generalization.

$$s = F_{ex}(z, W) = \sigma\big(g(z, W)\big) = \sigma\big(W_2 \delta(W_1 z)\big), \tag{4}$$

where r is the reduction ratio controlling computational cost and capacity, σ and δ refers to the sigmoid and rectified linear unit (ReLU) function, $W_1 \in R^{\frac{C}{r} \times C}$ and $W_2 \in R^{C \times \frac{C}{r}}$. In the proposed approach, we use an r value of 8, however we find that modifying the r value for the according data can assist with overall model performance.

Lastly, the original input is scaled using channel-wise multiplication of the channel scalar, s_c and feature map u_c:

$$\hat{X} = F_{scale}(u_c, s_c) = s_c \cdot u_c, \tag{5}$$

where $\hat{X} = [\hat{x}_1, \hat{x}_2 \dots, \hat{x}_C]$ and $u_c \in R^{H \times W}$.

2.5 Architecture

The initial step of the proposed architecture is the sequence generator, which aims at generating a set of sequential, channel oriented, feature maps. For this purpose, a channel attention variant of the U-Net is employed. In a typical U-Net architecture, the encoder, or down-stack, provides the model with contextual information about the input images, while the decoder, or up-stack, projects the lower resolution features to the pixel space [10]. The encoder and decoder are coupled with skip connections, which will assist in the transfer of pixel-related data that may be lost because of the pooling operations. The output of a typical U-Net model is a segmented image corresponding to its respective input image.

However, the goal of the proposed sequence predictor is not to generate a segmented variation of the input images, like a typical U-Net implementation, but rather to generate a sequential set of feature maps from both input images, which will later assist in the prediction of the flow field. The sequential set of feature maps will be discussed shortly, but for now, we focus on the necessary modifications that need to be made to the U-Net architecture to increase channel attention.

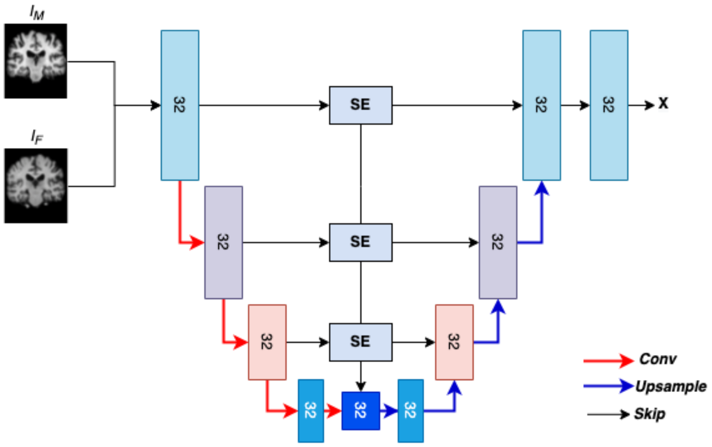


Figure 3: Sequence Generation Model

There is a fundamental lack of focus on channel-wise attention throughout the design of the classical U-Net. Rundo *et al.* showcased this with the addition of squeeze and excitation blocks after each step in the encoder and decoder pathways, to which they achieved excellent generalization results [11]. The addition of squeeze and excitation blocks to the network allows for a low complexity channel attention scheme. We implement a similar but reduced architecture, where channel attention is employed solely across the skip connections. Applying the squeeze and excitation blocks to only the skip connections achieves the channel attention benefits necessary to assist in bridging the semantic gap without the increased computational complexity entailed when they are added to each step in the encoder and decoder.

In addition, we implement a funnelling approach to connect the highest layers of the U-Net architecture to the lowest levels. This approach allows for the amalgamation of the highest resolution features to the lowest resolution features. The goal, again, is to bridge the semantic gap between the encoder and decoder. This connection is made after the application of channel attention, conducted on the skip connections.

In abstract, the process can be viewed as the selective weighting of high-resolution features, propagated by the highest layers in the U-Net architecture, amalgamated with the selectively weighted low-resolution features of the lowest levels. The resultant feature map is then processed using Long Short-Term Memory (LSTM), such that each channel is processed sequentially [12]. The channels are therefore treated as spatiotemporal sequences.

$$[x, y, n] \rightarrow n \ \times [x, y, 1] \tag{6}$$

The formulation of a classical LSTM is given by [13,14]:

$$i_t = \sigma(W_{xi}\, x_t + W_{hi} h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma\big(W_{xf}\, x_t + W_{hf} h_{t-1} + W_{cf} \circ c_{t-1} + b_f\big) \tag{8}$$

$$c_t = f_t \circ c_{t-1} + i_t \ \circ \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \tag{9}$$

$$o_t = \sigma(W_{xo}\, x_t + W_{ho} h_{t-1} + W_{co} \circ c_t + b_o) \tag{10}$$

$$h_t = o_t \ \circ \tanh(c_t) \tag{11}$$

where ∘ represents the Hadamard product. i, f, o are the input gate's activation vector, forget gate's activation vector, output gate's activation vector, respectively; c is the memory cell state vector and W is a weight matrix to be learned during the training phase.

However, as stated, the input and output of the model will be spatiotemporal sequences. Therefore, we modify the structure of the LSTM for the purpose of processing spatiotemporal sequences with the convolution operation. The ConvLSTM can be formulated as, where the convolution operator is represented with $*$ [15]:

$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \tag{12}$$

$$f_t = \sigma\big(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\big) \tag{13}$$

$$c_t = f_t \circ \mathcal{C}_{t-1} + i_t \ \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \tag{14}$$

$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \tag{15}$$

$$\mathcal{H}_t = o_t \ \circ \tanh(\mathcal{C}_t) \tag{16}$$

The produced set of sequential feature maps is then propagated through a squeeze and excitation block, in aim of modeling feature map interdependencies. A convolution operation is then conducted on the output of the squeeze and excitation block, however, as the goal is to continually capture and exploit channel relationships, the convolution operation is initially performed on individual channels. This operation is known as a depthwise convolution, where a single convolutional filter is applied to each individual channel [16]. Our primary goal in doing such is to create a dense representation of the features prevalent in each channel, which later will be fused with other channels.

$$[W, H, C] \to [Conv([W, H]_{c_1}), \dots, Conv([W, H]_{c_n})] \tag{17}$$

The last layer is a 2-dimensional convolution layer with a set of ϕ filters, where ϕ represents the dimensionality of the optical flow. For this paper, the registration is conducted on 2-D moving and fixed images; therefore, $\phi = 2$. For 3-D registration spaces $\phi = 3$. This has essentially allowed for greater modelling of channel interdependencies in the set of channel sequences, therefore increasing the accuracy of the registration field prediction.

$$D = Conv([Conv([W, H]_{c_1}), \dots, Conv([W, H]_{c_n})]) \tag{18}$$

The last phase in the model architecture is the transformation stage. To conduct the necessary dense transformation, we employ a spatial transformer network where bilinear interpolation is utilized to predict the new value of pixels in accordance with their respectively assigned flow [17].

$$V_i^C = \sum_{n}^{H} \sum_{m}^{W} U_{nm}^{c} \max(0, 1 - |x_i^s - m|) \max\,(0, 1 - |y_i^s - n|) \tag{19}$$

However, it is crucial to note that in the spatial transformer network proposed by in 2015 by Jaderberg, sampling is conducted identically for each channel for the purpose of maintaining spatial consistency. This can be a large issue if the input channels carry varying semantic concepts. To prevent this, we employ a squeeze and excitation block to the registration field input of the spatial transformer.

The goal in doing such is to weigh each of the channels in the input adaptively, therefore, the model can learn an optimal weighting scheme for the bilinear interpolation procedure without directly disturbing spatial consistency. The weighted

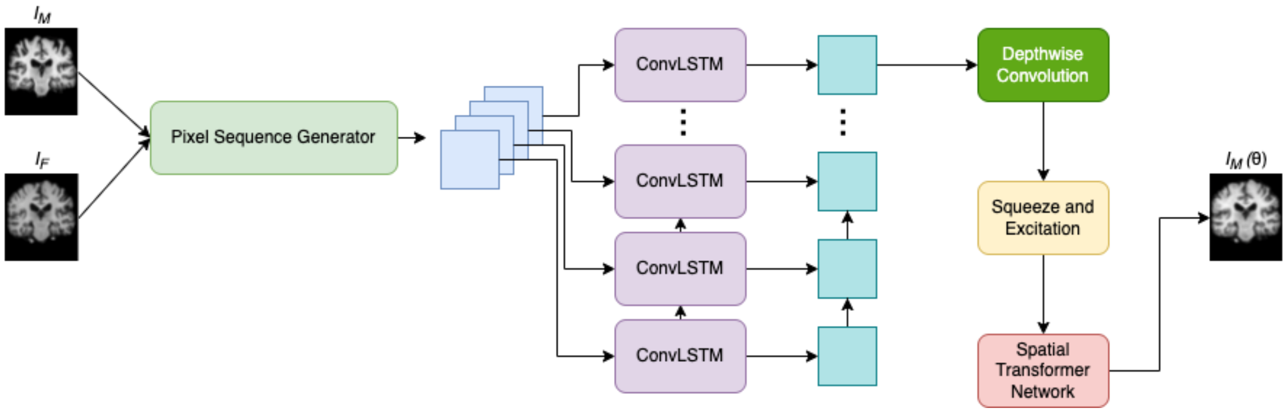


Figure 4: Cumulative Model Architecture. Moving image and fixed image are passed into the sequence generator, producing a sequential channel-wise output. The resultant output is sequentially processed with a ConvLSTM, producing a set of feature maps to be channeled down into the respective registration field. The output is passed into a spatial transformer network to produce the moved image.

channels are therein fed into the spatial transformer network, which conducts the respective bilinear interpolation with a weighted channel approach. The final transformation is given by:

$$I_D = T(I_M) = STN(D) \tag{20}$$

2.6 Defining Training Loss

As aforementioned, defining the cumulative registration process as an optimization problem will involve the implementation of a similarity measure. Mean squared error is typically implemented for solving this problem, where the average squared difference between I_f and I_d is estimated.

$$\mathcal{L}_{sim}(I_F, I_D) = \mathcal{L}_{sim}(I_F, T(I_M) = MSE\big(I_F, T(I_M)\big) = \frac{1}{\alpha}\sum_{p\exists\alpha}\left[I_{F_p} - T(I_M)_p\right]^2 \tag{21}$$

However, if we focus on only optimizing the similarity loss, then we typically produce a discontinuous T, thereby affecting the quality of I_d. A smoothness loss is also defined for the model based on diffusion regularization to combat this.

$$\mathcal{L}_{smooth}(D) = \sum_{p\exists\alpha}\left|\left|\nabla D(p)\right|\right|^2 \tag{22}$$

We can define the total loss as the sum of both the similarity loss and smoothness loss, with the addition of a regularization parameter, λ, to regulate the weighting of smoothness.

$$\mathcal{L}_{sim}(I_F, D, I_D) = \mathcal{L}_{sim}(I_F, I_D) + \lambda\,\mathcal{L}_{smooth}(D) \tag{23}$$

We note that the integration of various other losses, such as mean absolute error and Huber's loss, can be implemented. We also find that Huber's loss can decrease instability during the training procedure, as it is less sensitive to outliers. However, we choose to employ the same loss implemented by VoxelMorph, described above, with the aim of providing a controlled assessment of only the model dynamics.

3 DATASET AND IMPLEMENTATION

To comparatively test the model, we use a larger dataset of 1024 MRI brain scans acquired from various publicly available datasets, namely OASIS, MCIC, PPMI, HABS, ADHD, Harvard GSP and ABIDE. The dataset is further augmented to 1500 using simple augmentation techniques.

The proposed method was built using Keras with a Tensorflow backend. Adam is used as the gradient optimization method with a learning rate of $10e^{-4}$. The sample code is available at https://www.github.com/nalsadi/lstm-cor. The model is tested against a baseline, VoxelMorph, which was trained using the same compute instances. The code for the VoxelMorph model is publicly available at https://github.com/voxelmorph/voxelmorph.

5 RESULTS

The comparative analysis is conducted with the assistance of 2D MR brain images from the dataset discussed prior. The loss during the training procedure is recorded in Figure 5, where both models are trained for 900 epochs with a batch size of 8.

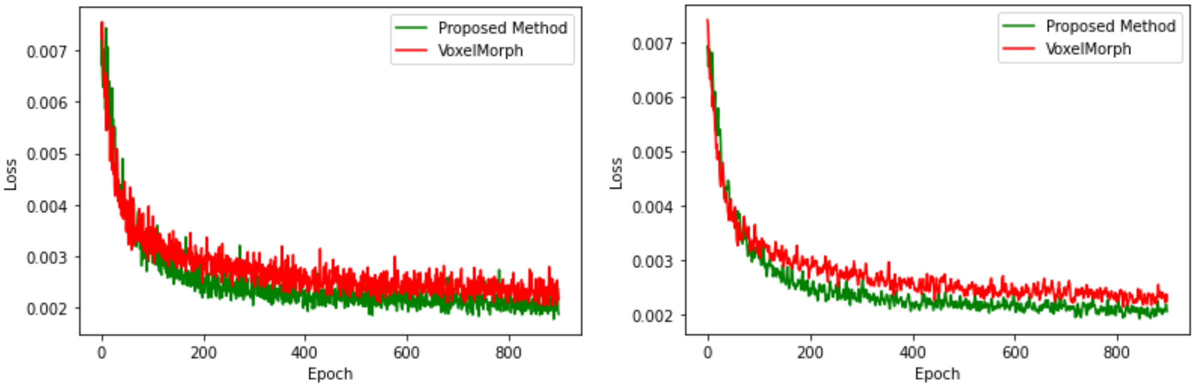


Figure 5: Training loss over 900 epochs. The figure on the left indicates the true results of the training procedure. The figure on the right showcases the same loss over the training procedure with a moving average.

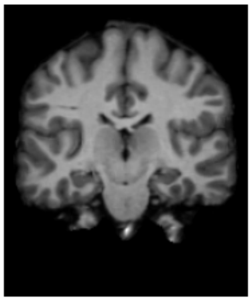
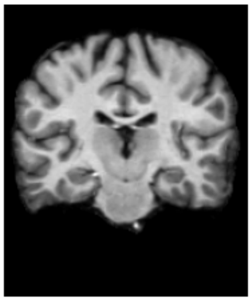
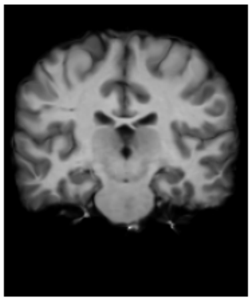
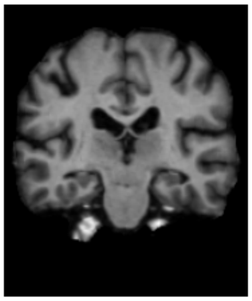
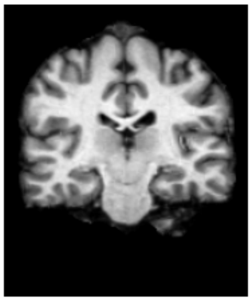
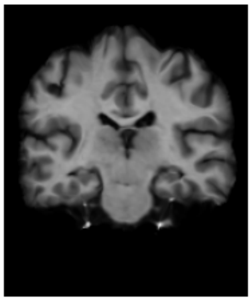
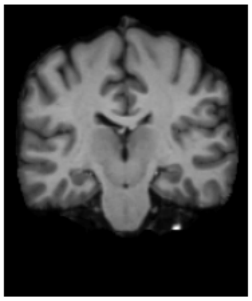
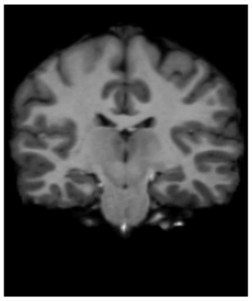
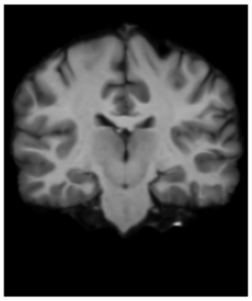
The results show that the proposed method achieved a lower average loss over the training period. This is more evident in the smoothed plot shown in Figure 5 with the assistance of a moving average. In addition, the average training loss incurred over the training period, as well as the average mean squared error and mean absolute error is listed in Table 1. The results indicate the proposed method outperformed the baseline, VoxelMorph, in achieving lower mean squared error and mean absolute error during both training and testing. Although, due to increased model complexity, the training time increased dramatically.

Table 1: Proposed Method and Baseline Performance

Approach	**Mean Training Loss**	**Mean Testing MSE**	**Mean Testing MAE**	**Dice Score**	**GPU Time**
Proposed Method	**0.002502**	**0.001227**	**0.017150**	**0.7642***	0.09(0.004)
VoxelMorph	0.002766	0.001576	0.019996	0.7521	**0.0459(0.03)**

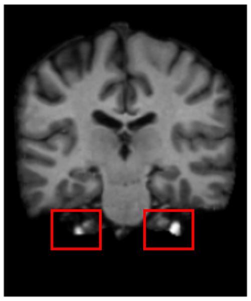
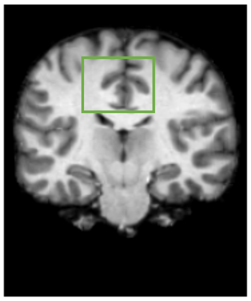
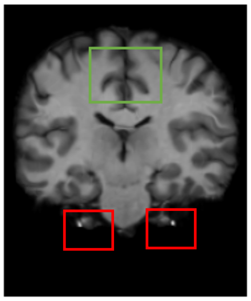
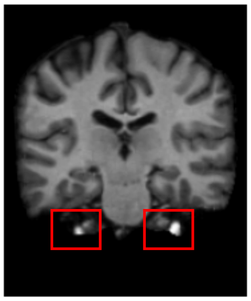
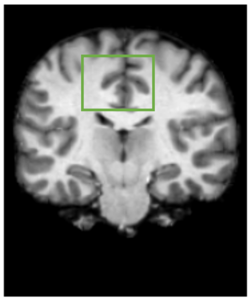
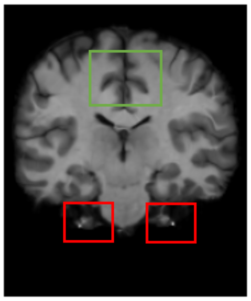
It is quite evident from the reported execution time that that model is relatively slower than the baseline. The results displayed in Table 2 show the proposed method performs very well in registering the respective moving image.

Table 2: Results of the proposed method on three pairs of variant scans.

Moving	Fixed	Moved
		
		
		

The proposed method is further compared with the baseline with a qualitative comparison of their registration results displayed in Table 3.

Table 3: Comparison of the proposed method and VoxelMorph on the same input image pair

Approach	Moving	Fixed	Moved
Proposed Method			
VoxelMorph			

The results indicate that the proposed method outperformed the baseline model in terms of accuracy, achieving a higher testing MSE and Dice score. In addition, the qualitative analysis of the performance of the proposed method shows excellent registration performance. However, the execution time has almost doubled for registering a pair of images. Figure 6 depicts the substantial increase in execution time.

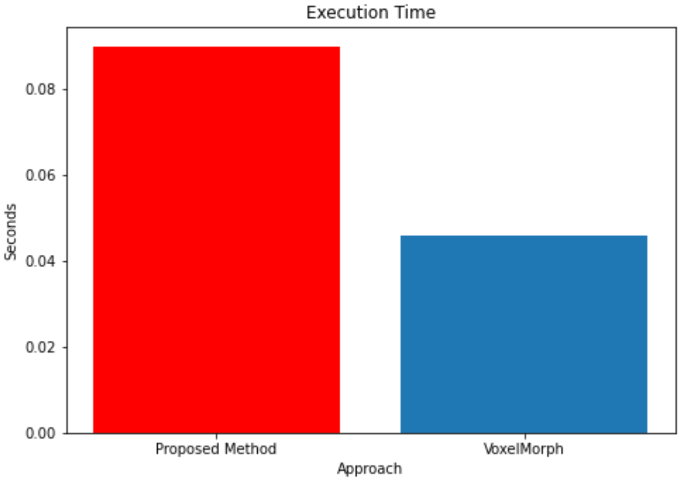


Figure 6: Execution Time of Proposed Method and Baseline Model

The execution time will need to be reduced for the proposed method to be practically implemented within clinical settings. Processing thousands of images with an increased execution time, almost doubling the baseline model, will introduce massive delays to the registration process.

CONCLUSION

In conclusion, we present a novel approach to unsupervised image registration where channel attention is prioritized. Squeeze and excitation blocks introduce an adaptive weighting scheme of channel-wise information captured via a sequence generator, composed of an encoder and decoder, from the input of a moving and fixed image. The output feature maps are viewed as a channel-wise spatiotemporal sequence. We process this sequence with the employment of a modified LSTM, namely the ConvLSTM. The features maps are then processed using convolution operations on each channel or a depthwise convolution layer. The output is then fed into a squeeze and excitation block and a spatial transformer network to produce a weighted channel approach for bilinear interpolation without disturbing spatial consistency. We test the proposed method on a two-dimension registration space with brain scans and compare it against a baseline, VoxelMorph. The proposed method achieved a lower MSE during the training and testing procedure and a higher DICE score, 0.7642, than the baseline. However, we note that the proposed method increased computational time but did so with an improved accuracy across all other metrics. Future work will be dedicated to maintaining accuracy while reducing execution time.

REFERENCES

[1] F. Maes, D. Robben, D. Vandermeulen, and P. Suetens, “The Role of Medical Image Computing and Machine Learning in Healthcare,” in *Artificial Intelligence in Medical Imaging: Opportunities, Applications and Risks*, E. R. Ranschaert, S. Morozov, and P. R. Algra, Eds. Cham: Springer International Publishing, 2019, pp. 9–23. doi: 10.1007/978-3-319-94878-2_2.

[2] S. K. Zhou, “Chapter 1 - Introduction to Medical Image Recognition, Segmentation, and Parsing,” in *Medical Image Recognition, Segmentation and Parsing*, S. K. Zhou, Ed. Academic Press, 2016, pp. 1–21. doi: 10.1016/B978-0-12-802581-9.00001-9.

[3] J. V. Hajnal and D. L. G. Hill, Eds., *Medical Image Registration*. Boca Raton: CRC Press, 2001. doi: 10.1201/9781420042474.

[4] J. B. A. Maintz and M. A. Viergever, “An Overview of Medical Image Registration Methods,” p. 22.

[5] X. Chen, A. Diaz-Pinto, N. Ravikumar, and A. Frangi, “Deep learning in medical image registration,” *Prog. Biomed. Eng.*, Dec. 2020, doi: 10.1088/2516-1091/abd37c.

[6] S. Shan, W. Yan, X. Guo, E. I.-C. Chang, Y. Fan, and Y. Xu, “Unsupervised End-to-end Learning for Deformable Medical Image Registration,” *ArXiv171108608 Cs*, Jan. 2018, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1711.08608

[7] G. Song, J. Han, Y. Zhao, Z. Wang, and H. Du, “A Review on Medical Image Registration as an Optimization Problem,” *Curr. Med. Imaging Rev.*, vol. 13, no. 3, pp. 274–283, Aug. 2017, doi: 10.2174/1573405612666160920123955.

[8] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “VoxelMorph: A Learning Framework for Deformable Medical Image Registration,” *IEEE Trans. Med. Imaging*, vol. 38, no. 8, pp. 1788–1800, Aug. 2019, doi: 10.1109/TMI.2019.2897538.

[9] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” *ArXiv170901507 Cs*, May 2019, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1709.01507

[10] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *ArXiv150504597 Cs*, May 2015, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1505.04597

[11] L. Rundo *et al.*, “USE-Net: incorporating Squeeze-and-Excitation blocks into U-Net for prostate zonal segmentation of multi-institutional MRI datasets,” *ArXiv190408254 Cs*, Jul. 2019, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1904.08254

[12] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks,” *ArXiv190909586 Cs*, Sep. 2019, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1909.09586

[13] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.

[14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” *ArXiv150604214 Cs*, Sep. 2015, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1506.04214

[16] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” *ArXiv161002357 Cs*, Apr. 2017, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1610.02357

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” *ArXiv150602025 Cs*, Feb. 2016, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1506.02025