Technical Note

Not peer-reviewed version

A Naive Trick to Accelerate Training of LNCC-Based Deep Image Registration Models

Xi Jia [*] , Xinxing Cheng , Jinming Duan , Bartłomiej W. Papież

Posted Date: 28 February 2025

doi: 10.20944/preprints202502.2200.v1

Keywords: Image Registration; Local Normalized Cross-Correlation; Deformable; Efficient

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 February 2025 doi:10.20944/preprints202502.2200.v1

Technical Note

A Naive Trick to Accelerate Training of LNCC-Based Deep Image Registration Models

Xi Jia [1,2,*], Xinxing Cheng [3], Jinming Duan [4] and Bartłomiej W. Papież [1,2]

1 Nuffield Department of Population Health, University of Oxford
2 Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford
3 School of Computer Science, University of Birmingham
4 Division of Informatics, Imaging and Data Sciences, University of Manchester
* Correspondence: xi.jia@ndph.ox.ac.uk

Abstract: Many public PyTorch repositories implement Local Normalized Cross-Correlation Loss (LNCC) using five sequential convolution operations. This implementation is, however, slow, failing to utilize modern hardware's performance potential fully. By simply replacing these convolutions with one single group convolution, we found the training time of LNCC-based deep registration models can be ***halved*** *without affecting the numerical results*, leading to notable cost savings. We hope that this simple approach will be beneficial to the community. An example code is available at https://github.com/xi-jia/FastLNCC.

Keywords: image registration; Local Normalized Cross-Correlation; deformable; efficient

---

1. Introduction

Local Normalized Cross-Correlation Loss (LNCC)[1], measuring the negative correlation of two input images in a sub-window manner, has been widely adopted for deformable image registration tasks [2,3]. Specifically, for a pair of source image I and reference image J, it is defined as

$$L_{\text{lncc}}(I,J) = 1 - \frac{1}{|\mathcal{N}|}\sum_{k\in\mathcal{N}} \frac{\sum_{i\in\mathcal{W}_k}(I_i-\bar{I}_k)(J_i-\bar{J}_k)}{\sqrt{\sum_{i\in\mathcal{W}_k}(I_i-\bar{I}_k)^2}\sqrt{\sum_{i\in\mathcal{W}_k}(J_i-\bar{J}_k)^2}},$$

where:

$$\begin{aligned}
\mathcal{N} &= \text{set of all voxels},\\
\mathcal{W}_k &= \text{local window centered at voxel } k,\\
\bar{I}_k &= \frac{1}{|\mathcal{W}_k|}\sum_{i\in\mathcal{W}_k} I_i \quad (\text{mean of } I \text{ in window } \mathcal{W}_k),\\
\bar{J}_k &= \frac{1}{|\mathcal{W}_k|}\sum_{i\in\mathcal{W}_k} J_i \quad (\text{mean of } J \text{ in window } \mathcal{W}_k),\\
I_i, J_i &= \text{intensity values of images } I \text{ and } J \text{ at voxel } i.
\end{aligned}$$

Following the traditional convolution-based implementation [1], many PyTorch-based registration models, such as VoxelMorph [4,5][1], DeepReg [6][2], SYMNet[7] [3], MultiPropReg [8][4], TransMorph [9][5],

---

1 https://github.com/voxelmorph/voxelmorph/blob/dev/voxelmorph/torch/losses.py
2 https://github.com/DeepRegNet/DeepReg/blob/main/deepreg/loss/image.py
3 https://github.com/cwmok/Fast-Symmetric-Diffeomorphic-Image-Registration-with-Convolutional-Neural-Networks/blob/master/Code/Models.py
4 https://github.com/Alison-brie/MultiPropReg/blob/main/losses.py
5 https://github.com/junyuchen245/TransMorph_Transformer_for_Medical_Image_Registration/blob/main/TransMorph/losses.py

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 February 2025 doi:10.20944/preprints202502.2200.v1

LKU-Net[10][6], MONAI [11][7], RAN [12][8], WiNet [13][9] and Rethink-Reg [14][10], just to name a few, implement LNCC using five sequential convolution operations, as outlined in Algorithm 1. We, however, found that the computations may be accelerated.

Algorithm 1 LNCC loss calculation with five sequential convolutions

Require: I (input image), J (reference image), w (window size)

Define a convolutional filter W of size $w \times w \times w$ with all elements equal to $\frac{1}{w^3}$

Compute $\mu_I = W * I$ ▷ Local mean of I

Compute $\mu_J = W * J$ ▷ Local mean of J

Compute $\sigma_I^2 = W * (I^2) - \mu_I^2$ ▷ Local variance of I

Compute $\sigma_J^2 = W * (J^2) - \mu_J^2$ ▷ Local variance of J

Compute $\sigma_{IJ} = W * (I \cdot J) - \mu_I \cdot \mu_J$ ▷ Local covariance of I and J

Compute $\text{LNCC} = \frac{\sigma_{IJ}}{\sqrt{\sigma_I^2 \cdot \sigma_J^2 + \epsilon}}$ ▷ Add small ϵ to avoid division by zero

Compute loss: $\text{Loss} = 1 - \text{mean}(\text{LNCC})$

return Loss

2. Implementation

2.1. Pseudo Code

The approach in Algorithm 1 uses five consecutive convolutions to compute the LNCC loss, which fails to fully leverage the parallel computing capabilities of modern hardware. We propose to replace the five consecutive convolutions with one single group convolution, as illustrated in Algorithm 2. This simple change can rapidly reduce the computational time on modern hardware (as illustrated in Figure 1), while ***the numerical results are not affected.*** (The computed loss values and gradients will be identical.)

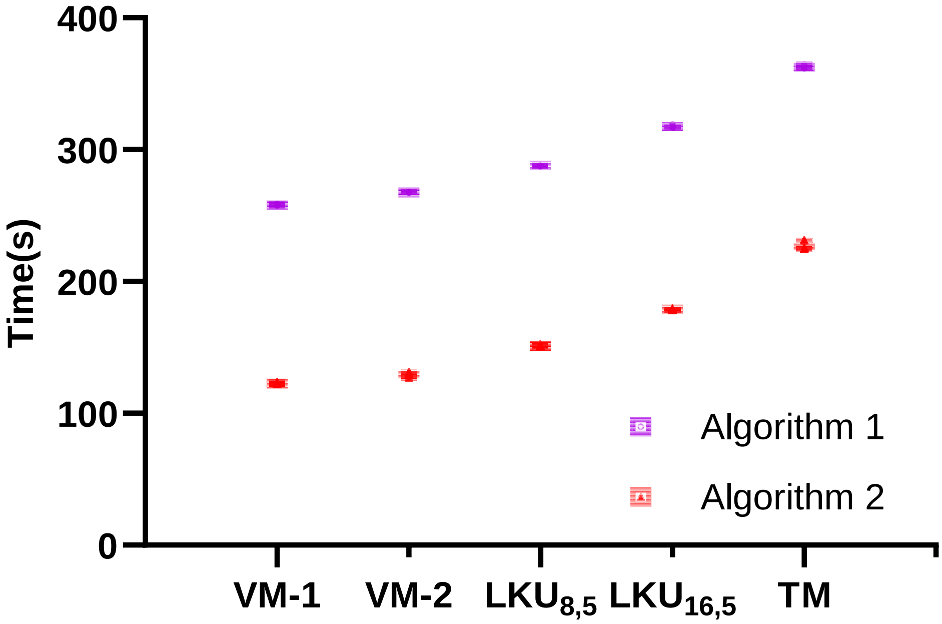

Figure 1. Training time (in seconds) required for every epoch (out of 5 epochs) using Algorithms 1 and 2, respectively. The latter reduces the training time significantly.

[6] https://github.com/xi-jia/LKU-Net/blob/main/LKU-Net_3D_IXI/Models.py
[7] https://github.com/Project-MONAI/MONAI/blob/dev/monai/losses/image_dissimilarity.py
[8] https://github.com/jianqingzheng/res_aligner_net/blob/main/ran_func/losses.py
[9] https://github.com/x-xc/WiNet/blob/main/WiNet_code/loss.py
[10] https://github.com/BailiangJ/rethink-reg/blob/main/models/losses/lncc.py

Algorithm 2 LNCC loss calculation with one group convolution

Require: I (input image), J (reference image), w (window size)

Define a convolutional filter W of size $w \times w \times w$ with all elements equal to $\frac{1}{w^3}$

Compute $Set = W * (I, J, I^2, J^2, I \cdot J)$ ▷ **Group convolution**

Get $\mu_I = Set[0]$ ▷ Local mean of I

Get $\mu_J = Set[1]$ ▷ Local mean of J

Compute $\sigma_I^2 = Set[2] - \mu_I^2$ ▷ Local variance of I

Compute $\sigma_J^2 = Set[3] - \mu_J^2$ ▷ Local variance of J

Compute $\sigma_{IJ} = Set[4] - \mu_I \cdot \mu_J$ ▷ Local covariance of I and J

Compute LNCC $= \frac{\sigma_{IJ}}{\sqrt{\sigma_I^2 \cdot \sigma_J^2 + \epsilon}}$ ▷ Add small ϵ to avoid division by zero

Compute loss: Loss $= 1 - \text{mean}(\text{LNCC})$

return Loss

Additionally, this approach may be applicable to other similarity measures that require repeated sliding window-based computations, such as the Structural Similarity Index Measure (SSIM).

2.2. Runtime Analysis

In Figure 1, we compare the training time of a deep registration model taken using the previous Algorithm 1 and the new Algorithm 2. Our experimental setup is as follows:

- Data: 403 $160 \times 192 \times 224$ training pairs from the IXI dataset (https://brain-development.org/ixi-dataset/) pre-processed by TransMorph [9] are selected as training data.
- Architectures: 5 benchmarking deep registration models, i.e., VoxelMorph-1, VoxelMorph-2 [4,5], LKU-Net$_{8,5}$, LKU-Net$_{16,5}$ [10], and TransMorph [9] are selected as comparing methods.
- Training: Each one of the five models is trained in five epochs using the Algorithm 1 and the new Algorithm 2, respectively.
- Results: The training time required for each epoch is reported to compare the difference between the two algorithms.
- Others: The batch size of all networks is set to 1. The sub-window of LNCC is fixated at $9 \times 9 \times 9$. All models are trained on the same Nvidia A100-40G GPU with Pytorch. (More experiments are in the Appendix A.)

We can observe that the runtime of the new LNCC implementation (Algorithm 2) is consistently and significantly less than the Algorithm 1 on all compared architectures. Specifically, the average training time on 5 epochs is reduced 52.49%, 51.72%, 47.54%, 43.69%, and 37.61% for VoxelMorph-1, VoxelMorph-2, LKU-Net$_{8,5}$, LKU-Net$_{16,5}$, and TransMorph. (Detailed numbers are included in Table 1.) The trend is easy to interpret as the feed-forward and back-propagation will account for more computations in larger models, making the computations of LNCC less critical.

Table 1. Training time in seconds. Left: Algorithm 1; Right: Algorithm 2.

Models	Epoch1	Epoch2	Epoch3	Epoch4	Epoch5	Avg	Epoch1	Epoch2	Epoch3	Epoch4	Epoch5	Avg
VM-1	258.95	258.07	257.69	258.02	257.62	258.07	123.37	122.31	122.40	122.52	122.46	122.61
VM-2	267.78	267.57	267.64	267.67	267.90	267.71	131.07	129.66	129.66	129.10	126.77	129.25
LKU$_{8,5}$	288.10	287.61	287.95	287.90	287.67	287.84	152.01	150.71	150.85	150.70	150.72	151.00
LKU$_{16,5}$	319.11	316.98	316.94	316.97	317.01	317.40	179.67	178.49	178.47	178.71	178.28	178.72
TM	364.44	362.00	361.99	361.95	362.01	362.48	231.30	224.78	224.59	225.09	225.08	226.17

3. Conclusion

We are gaining access to millions of scans from various imaging modalities to develop AI-driven healthcare applications, of which, image registration is often a fundamental step [15][11]. Researchers in the registration community contrive to accelerate the registration process for such large-scale cohorts.

[11] https://learn2reg.grand-challenge.org/

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 February 2025 doi:10.20944/preprints202502.2200.v1

Many deep image registration models have proven that their inference time can be orders of magnitude faster than conventional models, while the lengthy training time for deep models is the elephant in the room. Using a single group convolution, the training time for LNCC-based deep registration models can be nearly halved on modern hardware such as Nvidia A100 GPU, significantly accelerating the development of more advanced registration and downstream medical analysis models. We hope this approach will demonstrate practical value to the community.

Acknowledgments: The authors would like to acknowledge the funding from Novo Nordisk to support this work.

Appendix A

Table A1. Average runtime for one epoch on different GPUs

GPU	Quadro-rtx8000-48g			P100-sxm2-16gb		
	Algorithm1	Algorithm2	Reduced(%)	Algorithm1	Algorithm2	Reduced(%)
VM-1	669.5427791	395.2477166	40.97	1002.45329	640.9769462	36.06
VM-2	701.1765172	426.4876567	39.18	1079.542767	716.4298012	33.64
$LKU_{8,5}$	731.4320591	455.3439657	37.75	1160.251004	797.7661547	31.24
$LKU_{16,5}$	1022.4765293	748.9222134	26.75	1616.234878	1254.400603	22.39
TM	878.5947878	603.5537508	31.30	1241.964152	880.458546	29.11

Except for the runtime results on the Nvidia A100-40G GPU, we have trained the two algorithms on two other GPUs, i.e., Quadro-rtx8000-48g and P100-sxm2-16gb. The (reduced) training time for the five architectures is included in Table A1. Across all GPU and configurations, Algorithm 2 consistently shows a reduction in runtime compared to Algorithm 1. The reduction in runtime varies across different configurations, ranging from 22.39% to 40.97%.

References

1. Cachier, P.; Bardinet, E.; Dormont, D.; Pennec, X.; Ayache, N. Iconic feature based nonrigid registration: the PASHA algorithm. *Computer vision and image understanding* **2003**, *89*, 272–298.
2. Avants, B.B.; Epstein, C.L.; Grossman, M.; Gee, J.C. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis* **2008**, *12*, 26–41.
3. Lorenzi, M.; Ayache, N.; Frisoni, G.B.; Pennec, X.; (ADNI, A.D.N.I.; et al. LCC-Demons: a robust and accurate symmetric diffeomorphic registration algorithm. *NeuroImage* **2013**, *81*, 470–483.
4. Balakrishnan, G.; Zhao, A.; Sabuncu, M.R.; Guttag, J.; Dalca, A.V. An unsupervised learning model for deformable medical image registration. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9252–9260.
5. Balakrishnan, G.; Zhao, A.; Sabuncu, M.R.; Guttag, J.; Dalca, A.V. VoxelMorph: a learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging* **2019**, *38*, 1788–1800.
6. Fu, Y.; Brown, N.M.; Saeed, S.U.; Casamitjana, A.; Baum, Z.M.C.; Delaunay, R.; Yang, Q.; Grimwood, A.; Min, Z.; Blumberg, S.B.; et al. DeepReg: a deep learning toolkit for medical image registration. *Journal of Open Source Software* **2020**, *5*, 2705.
7. Mok, T.C.; Chung, A.C. Fast Symmetric Diffeomorphic Image Registration with Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
8. Liu, R.; Li, Z.; Fan, X.; Zhao, C.; Huang, H.; Luo, Z. Learning deformable image registration from optimization: perspective, modules, bilevel training and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**, *44*, 7688–7704.
9. Chen, J.; Frey, E.C.; He, Y.; Segars, W.P.; Li, Y.; Du, Y. TransMorph: Transformer for unsupervised medical image registration. *Medical Image Analysis* **2022**, p. 102615.

 doi:10.20944/preprints202502.2200.v1

10. Jia, X.; Bartlett, J.; Zhang, T.; Lu, W.; Qiu, Z.; Duan, J. U-net vs transformer: Is u-net outdated in medical image registration? In Proceedings of the International Workshop on Machine Learning in Medical Imaging. Springer, 2022, pp. 151–160.
11. Cardoso, M.J.; Li, W.; Brown, R.; Ma, N.; Kerfoot, E.; Wang, Y.; Murrey, B.; Myronenko, A.; Zhao, C.; Yang, D.; et al. Monai: An open-source framework for deep learning in healthcare. *arXiv preprint arXiv:2211.02701* **2022**.
12. Zheng, J.Q.; Wang, Z.; Huang, B.; Lim, N.H.; Papież, B.W. Residual Aligner-based Network (RAN): Motion-separable structure for coarse-to-fine discontinuous deformable registration. *Medical Image Analysis* **2024**, *91*, 103038.
13. Cheng, X.; Jia, X.; Lu, W.; Li, Q.; Shen, L.; Krull, A.; Duan, J. WiNet: Wavelet-based Incremental Learning for Efficient Medical Image Registration. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2024, pp. 761–771.
14. Jian, B.; Pan, J.; Ghahremani, M.; Rueckert, D.; Wachinger, C.; Wiestler, B. Mamba? Catch The Hype Or Rethink What Really Helps for Image Registration. In Proceedings of the International Workshop on Biomedical Image Registration. Springer, 2024, pp. 86–97.
15. Hering, A.; Hansen, L.; Mok, T.C.; Chung, A.C.; Siebert, H.; Häger, S.; Lange, A.; Kuckertz, S.; Heldmann, S.; Shao, W.; et al. Learn2Reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. *IEEE Transactions on Medical Imaging* **2022**, *42*, 697–712.