



HAL
open science

Assessing Layer Normalization with BraTS MRI Data in a Convolution Neural Net

Akhilesh Rawat, Rajeev Kumar

► **To cite this version:**

Akhilesh Rawat, Rajeev Kumar. Assessing Layer Normalization with BraTS MRI Data in a Convolution Neural Net. 5th International Conference on Computational Intelligence in Data Science (ICCIDS), Mar 2022, Virtual, India. pp.124-135, 10.1007/978-3-031-16364-7_10 . hal-04381305

HAL Id: hal-04381305

<https://inria.hal.science/hal-04381305v1>

Submitted on 9 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Assessing Layer Normalization with BraTS MRI Data in a Convolution Neural Net

Akhilesh Rawat¹ and Rajeev Kumar²
Data to Knowledge (D2K) Lab,
School of Computer & Systems Sciences,
Jawaharlal Nehru University, New Delhi 110 067, India

¹ *akhile62_scs@jnu.ac.in*
² *rajeevkumar.cse@gmail.com*

Abstract. Deep learning-based Convolutional Neural Network (CNN) architectures are commonly used in medical imaging. Medical imaging data is highly imbalanced. A deep learning architecture on its own is prone to overfit. As a result, we need a generalized model to mitigate total risk. This paper assesses a layer normalization (LN) technique in a CNN-based 3D U-Net for faster training and better generalization in medical imaging. Layer Normalization (LN) is mostly used in Natural Language Processing (NLP) tasks such as question-answering, handwriting sequence generation, etc. along with Recurrent Neural Network (RNN). The usage of LN is yet to be studied in case of medical imaging. In this context, we use brain MRI segmentation and train our model with LN and without normalization. We compare both models and our LN-based model gives 32% less validation loss over without normalization-based model. We achieve validation dice scores of unseen input data passes to LN based model of 0.90 (7.5% higher than without normalization) for edema, 0.74 (12.5% higher than without normalization) for non-enhancing tumor and 0.95 (1.5% higher than without normalization) for enhancing tumor.

Keywords: Convolution Neural Net · 3D U-Net · Generalization · Layer Normalization · Batch Normalization · Group Normalization · Instance Normalization · BraTS · MRI.

1 Introduction

Recent advances in the interpretation of medical images specially in brain tumor segmentation make substantial use of deep neural networks [16], [5], [11], [12], [18], [10]. Convolutional Neural Network (CNN) is most often used as deep learning architecture for medical image analysis. We used BraTS 2016 and 2017 datasets [17] for brain MRI tumor segmentation and applying layer normalization on CNN based 3D U-Net to generalize the deep neural network. By default, traditional approaches employ Batch Normalization (BN) with a 3D U-Net architecture [12], [3]. Many researchers apply Group Normalization (GN) [11], [6] and

Instance Normalization (IN) [13] in medical imaging. Layer Normalization (LN) is mostly used in NLP tasks such as question-answering, handwriting sequence generation, etc. with Recurrent Neural Network (RNN) [1]. CNN served as an effective tool for representing and presenting the input data. CNN-produced features were so effective that handcrafted features became obsolete. This is not to say that domain specialists are unnecessary. Domain knowledge aids in the selection of the CNN architecture, resulting in a more performant CNN model.

The backpropagation learning technique poses difficulties for deep learning networks. Numerous gradient descent algorithms may be accelerated using various optimization approaches, such as momentum optimization, Adagrad, and so on. Additionally, normalization approaches improve the stability and learning speed of DL. Many studies employed batch normalization [4], instance normalization [20], group normalization [21], layer normalization [1], and other normalizing approaches in DNN. When CNN is used for computer vision or medical image analysis, a problem called "covariate shift" occurs [15]. This issue confuses the training process, and the model provides biased output. Lofe and Szegedy [4] devised the method of Batch Normalization (BN) to solve this issue. They normalize the input data and output of the hidden layers and make learning faster and helps in faster convergence. The primary issue with BN is that performance is highly dependent on batch size. Also, we cannot normalize until the complete batch is operated on during batch mode training. This research assessing normalization with Layer Normalization (LN) approach in a CNN-based 3D U-Net [2] deep learning architecture. The same technique has been used in Recurrent Neural Network (RNN) [1]. We normalized the input layer data and hidden layer data through LN. There are several parameters and weights in the hidden layer, and each layer is a weighted sum of the activations of the preceding layers. Back-propagation updates all weights repeatedly. Hence, the distribution may not be consistent over epochs. As a result, we're facing the identical issue of covariate shift [15]. To see the impact of this issue on training, we train our model using layer normalization as well as without normalization. Finally, we assess training effects in CNN-based 3D U-Net on the brain tumor segmentation with BraTS MRI dataset.

The remainder of the paper is laid out as follows: the literature review is included in Section 2. The normalizing approach is explained in Section 3. Section 4 discusses the dataset and explains the functioning of the 3D U-Net architecture with Layer Normalization. Section 5 concludes this work.

2 Literature Review

In general, normalization improves the efficiency, speed, and accuracy of the learning process. When a deep neural network is trained and an image dataset is passed to network through batches, one of the problems encountered by Shimodaira [15] is referred to as covariate shift. It is defined as the shift in the distribution of network activations caused by training parameter changes. To address this issue, Ioffe and Szegedy [4] developed a technique called Batch Nor-

malization (BN). This paper defines how BN works. They presented an algorithm for how to train a network with BN. They used the MNIST dataset to predict the digit class and the ImageNet dataset to perform classification. They employed a CNN architecture with BN with a batch size of 32 for their analysis. They found that BN required fewer iterations during training to achieve higher accuracy than a normal architectures. To address the shortcoming of batch normalization Yao et al. [22] have proposed a cross iteration batch normalization (CBN) in which numerous recent iterations are jointly utilizes to improve performance. They used benchmark dataset Image net and COCO for image classification and object detection respectively. With batch size 4, Naive CBN surpasses BN by 1.7 % in accuracy.

Wu and He [21] introduced the technique of Group Normalization (GN). This technique is used to normalize the values. They demonstrated that GN is a straightforward representation of Batch Normalization (BN). They used ImageNet, COCO dataset for image segmentation and detection, and the Kinetics dataset for video classification. They trained our model using ResNet-50. As a result, they determined that GN had a 10.6 % lower error rate than BN (batch size 2). When we reduce the batch size or use a tiny batch size during training, BN becomes worse. GN is independent of batch sizes, so it outperforms BN.

Ba et al. [1] have proposed another normalization technique called Layer Normalization (LN). They noticed that in BN, performance suffers when the batch size is decreased. The BN-related architecture can not normalize the input until the complete batch is processed. In LN, they computed the mean and variance used to normalize all the features extracted from a single example. For the experiment, they used the MNIST dataset for classification and RNN for image-sentence ranking, contextual language modeling, etc.

Shrestha et al. [16] have enhanced the batch normalization technique to make deep learning faster. They used this for segmentation in medical imaging, specifically prostate cancer segmentation. These images are magnetic resonance (MR) images. They found that DL had not accurately performed prostate segmentation on MR images, so they used BN in deep neural architecture. They used the U-Net architecture and modified the state-of-the-art BN through a modification in the loss function. Finally, their proposed method reduces the processing time and enhances accuracy. To compute the accuracy, they use the dice similarity coefficient.

Reinhold and Jacob [14] have investigated seven different techniques for normalizing the intensity distribution of MRI brain images over three contrasts (T1-w, T2-w, and FLAIR). They demonstrated with the Kirby-21 dataset. The experiments have shown that synthesis results are robust to the choice of the normalization technique. They recommended normalization as a pre-processing step before data is served as an input in patch-based DNN.

Numerous authors [11], [6], [12], [13] have applied data augmentation and normalization techniques on the BraTS dataset of MR images. Data normalization built new patches from the existing patches to overcome the large variation in brain tumor heterogeneity and scarcity of data. Andriy [11] has used group

normalisation followed by ReLU and traditional CNN to iteratively decrease the dimension of images. Zeyu and Changxing have proposed a novel two stage U-Net architecture to segment the brain tumor from coarse to fine. They used group normalisation as pre-processing technique followed by ReLU on BraTS 2019 dataset. Kumar and Rockett [8] proposed a bootstrapping procedure to partition data across modules and thus improving generalization through training of multiple neural modules [9].

Zeyu et al. [6] have proposed a novel two stage U-Net architecture to segment the brain tumor from coarse to fine. They used group normalisation as pre-processing technique followed by ReLU on BraTS 2019 dataset. Hiba and Ines have proposed a pre-processing technique based on contrast enhancement and intensity normalization on MRI data. Automatic segmentation is a challenging due to the high spatial and structural heterogeneity of brain tumours. Sergio et al. [13] have presented a CNN based architecture for automatic segmentation on brain tumor (BraTS-2013 and 2015). MR images. They applied intensity normalization at pre-processing level to make the contrast and intensity ranges between patients and acquisitions more consistent. Experimentally, they shown that intensity normalization enhances the dice similarity coefficient matrix. Nosheen et al. [18] have proposed a smart segmentation technique for whole brain tumor based on standard dataset BraTS-2019 and BraTS-2020. They use 3D UNET architecture followed by batch normalization layer.

3 Normalization

Normalization is a technique used to normalize the heterogeneous data on the same scale. This approach is used in deep learning architectures to accelerate learning, maintain network stability, and improves the overall performance. Normalization procedures are classified into four categories. Normalization by Batch (BN) [4], Layer (LN) [1], Instance (IN) [20], and Group (GN) [21]. We employ layer normalization to train our CNN-based 3D U-Net. In LN, we consider the feature maps calculated by multiple kernels over a single training data. Then we calculate mean and standard deviation over these feature maps derived from one training sample. Another mean and s.d. derived from another training sample, and so on. The dimensionality of μ and σ vector is the same as the channel 'C.' Let us suppose we have 'N' number of training images and C number of channels and every channel is of width 'W' and height 'H' provided by CNN. x represents one feature element in channel C at point j, k in that particular channel feature.

$$\mu_N = \frac{1}{CWH} \sum_{i=1}^C \sum_{j=1}^W \sum_{k=1}^H x_{Nijk} \quad (1)$$

$$\sigma_N^2 = \frac{1}{CWH} \sum_{i=1}^C \sum_{j=1}^W \sum_{k=1}^H (x_{Nijk} - \mu_N)^2 \quad (2)$$

We calculate μ_N and σ_N^2 , the mean and variance of the same data across the same layer, using the preceding Equations 1 and 2. Now, we'll normalize feature by using Equation 3, where ϵ is a tiny positive constant and prevent the division by zero condition.

$$\hat{x} = \frac{x - \mu_N}{\sqrt{\epsilon + \sigma_N^2}} \quad (3)$$

$$Y_i = \gamma \hat{x}_i + \beta \quad (4)$$

After applying LN, we do reparameterization. We have to compute Y_i (Equation 4), where γ and β are new parameters that define the distribution of the data. During the backpropagation, γ and β are tunable, and it can be tuned in the same gradient descent procedure along with the weight vector of the neural network.

In Batch Normalization (BN), we collect all identical channels from all training examples in the same batch and group them together. We make mini batches and compute their mean and standard deviation using Equations 5 and 6, where μ_C and σ_C^2 are mean and variance of a particular mini batche, respectively.

$$\mu_C = \frac{1}{NWH} \sum_{i=1}^N \sum_{j=1}^W \sum_{k=1}^H x_{iCjk} \quad (5)$$

$$\sigma_C^2 = \frac{1}{NWH} \sum_{i=1}^N \sum_{j=1}^W \sum_{k=1}^H (x_{iCjk} - \mu_C)^2 \quad (6)$$

After calculating the values of μ_C and σ_C^2 from identical channels, we use Equation 7 to normalize the values of mini batches. The fundamental issue with BN is that its performance is highly dependent on the batch size. The performance will decrease if we reduce the batch size.

$$\hat{x} = \frac{x - \mu_C}{\sqrt{\epsilon + \sigma_C^2}} \quad (7)$$

$$Y_i = \gamma \hat{x}_i + \beta \quad (8)$$

Finally, Equation 8 does the reparameterization; during the backpropagation phase, γ and β values are tunable with the same gradient descent procedure as the weight vector of the neural network. In BN, we cannot normalize the data unless the entire batch is processed.

In Instance Normalization (IN), we consider every individual channel and normalize them with respect to that channel's mean μ and standard deviation σ . We do not consider features belonging to other channels for computing the mean and standard deviation. We normalize the features in that particular channel only. After applying IN, we do the reparameterization as in Equation 8 and both γ and β are tunable during the backpropagation.

In Group Normalization (GN), the different channels are grouped into different groups. We compute the mean and standard deviation of each group. After computing μ and σ we normalize the features belonging to that group only with respect to the corresponding mean and standard deviation. Furthermore, we follow the same process of reparameterization that projecting back the normalized data as $y_i = \gamma \hat{x}_i + \beta$. The values of γ and β will be tuned during the backpropagation along with the parameters or weight vector of the neural network.

4 Experimental Setup and Results

4.1 Dataset Description

In this paper, we have used MRI multisequence data for brain tumor segmentation from the BraTS 2016 and BraTS 2017 datasets. Simpson [17] et al. combined these datasets and made them publicly available. This dataset was used in the challenge (brain tumor segmentation) organized by the Medical Segmentation Decathlon [17]. The segmentation task aims to segment three brain tumors called Edema, Enhancing tumor and Non-Enhancing tumor. The dataset contains 750 MRI scans of patients receiving High Grade Glioma (HGG) or Low Grade Glioma (LGG) treatment. All images are a 4-dimensional array of size $240 \times 240 \times 160 \times 4$. The MRI scans are multi-parametric and comprise native (T1) and post-Gadolinium (Gd) contrast T1-weighted (T1-Gd) volumes, as well as native T2-weighted (T2) and T2- Fluid Attenuated Inversion Recovery (FLAIR) volumes shown in Fig. 1 .

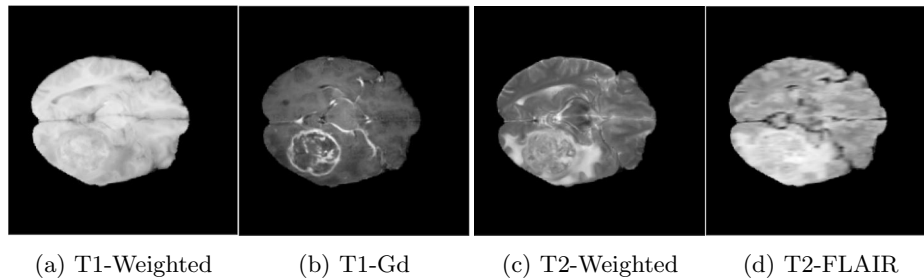


Fig. 1: Four Modalities of MRI Images in 2D space for visualization.

The BraTS dataset includes MRI scans from 19 medical institutes. These MRI scans were performed routinely on patients at specific hospitals. There are 484 data points in this dataset. For our experiments, we divided these data points into two partitions: training and testing. The 80% of all data points are utilized for training, while the rest 20% are used for prediction. As a result, our training set has 387 data points, whereas our test data contains 97. Every MRI scan is

rescaled to a $1mm^3$ isotropic resolution, and the skull is eliminated. These scan is done in four modalities, we denoted these modalites with numbers (ranging from 0 to 3); $0 \rightarrow$ FLAIR, $1 \rightarrow$ T1-weighted, $2 \rightarrow$ T1-Gd, and $3 \rightarrow$ T2-weighted . Multidimensional MRI images (each shape $240 \times 240 \times 160 \times 4$) are given as input to the 3D U-Net architecture. The first three dimensions represent each point in 3D volume called a voxel. A voxel is a data point in a three dimensional volume. The fourth dimension aids in accurately detecting tumor locations by providing additional information beyond the voxel data. These images (represented in voxel) are associated with four labels. Various numbers denote these labels (ranging from 0 to 3), i.e., 0 denotes the background, 1 denotes edema, 2 denotes non-enhancing tumors, and 3 denotes enhancing tumors.

We extracted five subvolumes from each input image to efficiently train and test the model. Each volume size is $160 \times 160 \times 16 \times 4$. We have removed those subvolumes that contain 5% tumor regions. We used the HDFS-5 data format to store these subvolumes and made a large dataset for training. 3D U-Net architecture to learn with these subvolumes. For training, we have 1560 subvolumes, and 440 subvolumes are used for validation. Before training we standardize the input data with its mean (μ) and standard deviation (σ). The sub-volume obtained is standardized over each channel and Z plane.

4.2 3D U-Net Architecture

To learn the underlying pattern, a deep neural network requires a substantial amount of input samples. In medical applications, the training data is small, which is insufficient to quell the deep learning architecture’s surge. As a result, we need a system capable of learning from a small sample size. For automated brain tumour image segmentation. Fig. 2 depicts our suggested architecture. It includes an encoder and a decoder. Both contraction and expansion exist. These two portions are symmetric. 3D U-Net uses 3D volumetric images. Hmeed et al. [3] have used the U-Net architecture with the Batch Normalization (BN) for MRI brain tumor segmentation. The contraction component gets the input image first. The model accepts input of size $4 \times 160 \times 160 \times 16$. The input layer is followed by downsampling units with one 3D convolution with $3 \times 3 \times 3$ filter size and $1 \times 1 \times 1$ stride. Then normalization, activation, dropout, 3D convolution, and max pooling with a pool size of $2 \times 2 \times 2$.

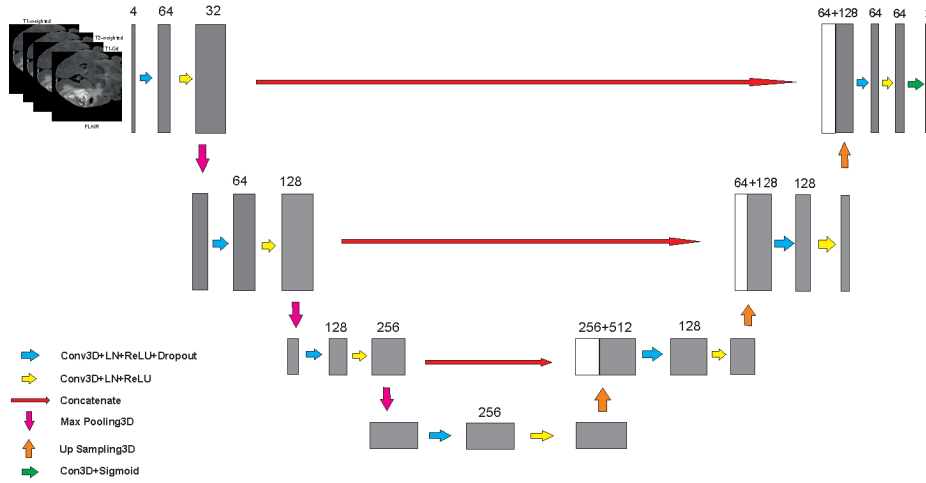


Fig. 2: Layer Normalization based 3D U-Net Architecture for MRI segmentation

The network with a 25% dropout rate has four downsampling units and symmetric upsampling till the output layer. Each upsampling layer has a concatenation layer. The network uses ‘ReLU’ activation function for hidden layers, whereas the output layer uses the ‘sigmoid’ activation function. We employ standardization, dropout, layer normalization, and data augmentation techniques to improve brain tumor image segmentation through this architecture. We also train the 3D U-Net architecture shown in Fig. 2 without normalization layer and evaluate both performances.

4.3 Experimental Results

This part trains our model and compares the results with and without layer normalization. We utilized Adam Optimizer to optimize [7]. The experiment was conducted using a 6 GB NVIDIA GeForce RTX 3060/144 Hz GPU and 16 GB of RAM. We utilized Python as the platform and two open-source libraries, Keras and TensorFlow. We have worked with a batch size of one. Given the computational restrictions, we trained the model for 50 epochs with an initial learning rate of 10^{-4} . We performed MRI scans on 1560 sub volumes (defined in section 4.1). It takes around one day to complete the training iterations. However, the behavior of the findings is almost the same. As a consequence of space constraints, we give just one set of findings.

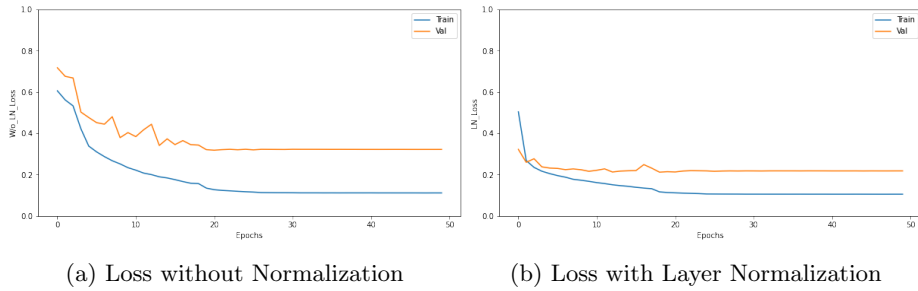


Fig. 3: Loss without normalization and with Layer Normalization

4.3.1 Generalization for Improved Performance

In Fig. 3, training and validation loss behavior keep on changing with their ups and downs in errors. However, both training and validation loss errors converge after some particular epochs. LN-based architecture (training loss 0.1115 and validation loss 0.3220) gives 32% less validation loss than without Normalization based architecture (training loss: 0.1054 and validation loss: 0.2182). This demonstrates that the LN-based model outperforms the competition and provide stability.

Additionally, we calculated performance in terms of the Dice Coefficient, as the majority of publications have done (i.e., [19], [13] etc.). Fig. 4 illustrates that the validation dice coefficient of the LN-based design has increased by 17.5% when Normalization is used. Both of the training dice Coefficient after 50 epoch almost the same. Hence, it shows that the LN-based approach gives better prediction and generalizability than without the normalization-based approach.

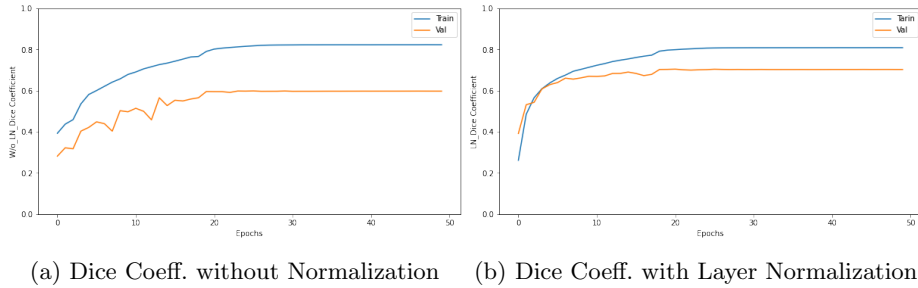


Fig. 4: Dice Coefficient without normalization and with Layer Normalization

4.3.2 Segmentation for Abnormality

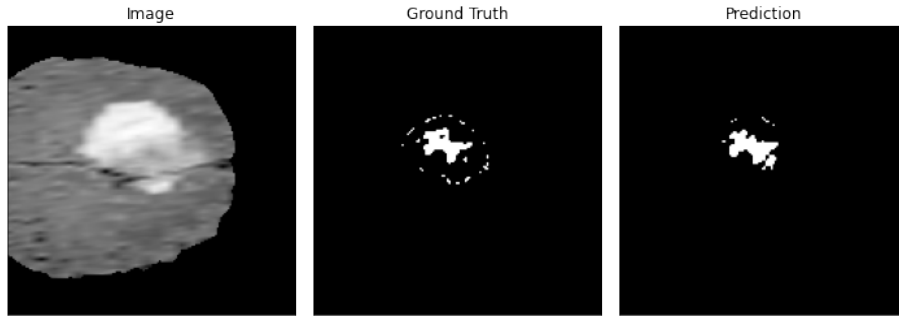


Fig. 5: Tumor prediction with layer Normalization deep learning architecture

This section demonstrates that how all LN-based empirical results are better than without normalization based model. In high-dimensional MRI data, the LN-based 3D U-Net model is trained to correctly determine the shape, size, and location of the tumorous cell. We have also evaluated the segmentation of the subvolumes. The tumor region of ground truth and predicted data can be visualized in 2D images. Both architectures probabilistically predict cancer for each pixel. We have chosen 0.5 as a threshold for visualization to discretize the output.

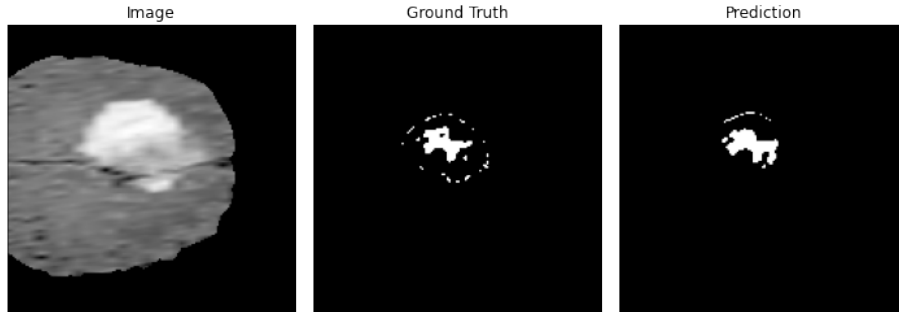


Fig. 6: Tumor prediction with out layer Normalization deep learning architecture

The segmentation results of unseen random data with corresponding ground truth are shown in Fig. 5 and Fig. 6 with LN-based and without normalization-based 3D U-Net architectures, respectively. For the evaluation of both model predictions, we used different evaluation measures [13], including sensitivity, specificity, FPR, and dice score.

Table 1: Performance measures with Layer Normalization 3D U-NET

	Edema	Non-Enhancing-Tumor	Enhancing Tumor
Sensitivity	0.8785	0.7391	0.9231
Specificity	0.9973	0.9978	0.9989
FPR	0.0027	0.0022	0.0011
Dice_Score	0.9014	0.7419	0.9465

Table 1 and Table 2 show the empirical results of both architectures. Dice_score of the LN-based architecture gained 7.5% higher than without normalization-based architecture for Edema, 12.5% higher Dice_score for Non-Enhancing-Tumor, and 1.5% higher Dice_score for Enhancing Tumor. Specificity outperforms sensitivity in both architectures (Table 1 and Table 2).

Table 2: Performance measures without Normalization in 3D U-NET

	Edema	Non-Enhancing-Tumor	Enhancing Tumor
Sensitivity	0.8355	0.7440	0.9117
Specificity	0.9940	0.9956	0.9982
FPR	0.0060	0.0044	0.0018
Dice_Score	0.8381	0.6591	0.9317

LN based model shows less FPR than, without normalization based model. For all three forms of cancer, the LN-based model is more generalized and exhibits significant outcomes on random input.

5 Conclusion

In summary, We assessed the LN technique on CNN based 3D U-Net architecture for brain MRI tumor segmentation. Additionally, we trained 3D U-Net architecture without using any normalizing approaches. During training phase with 50 epoch and batch size 1, LN based approach has given 32% reduction in validation loss compared to the without normalization based approach. Tumor segmentation has also done better in empirical studies using LN-based architecture. By and large, Layer Normalization minimizes covariate shift effects during training, resulting in faster learning and a more generic architecture.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: Proc. Int. conf.

- Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 424–432. Springer (2016)
3. Hmeed, A.R., Aliesawi, S.A., Jasim, W.M.: Enhancement of the u-net architecture for mri brain tumor segmentation. In: Next Generation of Internet of Things, pp. 353–367. Springer (2021)
 4. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. Int. conf. Machine Learning (ICML). pp. 448–456. PMLR (2015)
 5. Jiang, X., Chang, L., Zhang, Y.D.: Classification of alzheimer’s disease via eight-layer convolutional neural network with batch normalization and dropout techniques. *J. Medical Imaging & Health Informatics* **10**(5), 1040–1048 (2020)
 6. Jiang, Z., Ding, C., Liu, M., Tao, D.: Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In: Proc. Int. MICCAI Brainlesion Workshop. pp. 231–241. Springer (2019)
 7. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arxiv (2014). arXiv preprint arXiv:1412.6980 (2017)
 8. Kumar, R., Chen, W.C., Rockett, P.: Bayesian labelling of image corner features using a grey-level corner model with a bootstrapped modular neural network. In: Proc. Fifth Int. Conf. Artificial Neural Networks (Conf. Publ. No. 440). pp. 82–87 (1997). <https://doi.org/10.1049/cp:19970706>
 9. Kumar, R., Rockett, P.: Multiobjective genetic algorithm partitioning for hierarchical learning of high-dimensional pattern spaces: a learning-follows-decomposition strategy. *IEEE Trans. Neural Networks* **9**(5), 822–830 (1998). <https://doi.org/10.1109/72.712155>
 10. Lin, F., Wu, Q., Liu, J., Wang, D., Kong, X.: Path aggregation u-net model for brain tumor segmentation. *Multimedia Tools and Applications* **80**(15), 22951–22964 (2021)
 11. Myronenko, A.: 3D MRI brain tumor segmentation using autoencoder regularization. In: Proc. Int. MICCAI Brainlesion Workshop. pp. 311–320. Springer (2018)
 12. Mzoughi, H., Njeh, I., Wali, A., Slima, M.B., BenHamida, A., Mhiri, C., Mahfoudhe, K.B.: Deep multi-scale 3d convolutional neural network (cnn) for mri gliomas brain tumor classification. *J. Digital Imaging* **33**, 903–915 (2020)
 13. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Medical Imaging* **35**(5), 1240–1251 (2016)
 14. Reinhold, J.C., Dewey, B.E., Carass, A., Prince, J.L.: Evaluating the impact of intensity normalization on mr image synthesis. In: Medical Imaging: Image Processing. vol. 10949, p. 109493H. Proc. Int. Society Optics and Photonics (2019)
 15. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Statistical Planning & Inference* **90**(2), 227–244 (2000)
 16. Shrestha, S., Alsadoon, A., Prasad, P., Seher, I., Alsadoon, O.H.: A novel solution of using deep learning for prostate cancer segmentation: enhanced batch normalization. *Multimedia Tools & Applications* **80**(14), 21293–21313 (2021)
 17. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019)
 18. Sohail, N., Anwar, S.M., Majeed, F., Sanin, C., Szczerbicki, E.: Smart approach for glioma segmentation in magnetic resonance imaging using modified convolutional network architecture (u-net). *Cybernetics and Systems* **52**(5), 445–460 (2021)

19. Tseng, K.L., Lin, Y.L., Hsu, W., Huang, C.Y.: Joint sequence learning and cross-modality convolution for 3d biomedical segmentation. In: Proc. IEEE conf. Computer Vision and Pattern Recognition (CVPR). pp. 6393–6400 (2017)
20. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
21. Wu, Y., He, K.: Group normalization. In: Proc. European Conf. Computer Vision (ECCV). pp. 3–19 (2018)
22. Yao, Z., Cao, Y., Zheng, S., Huang, G., Lin, S.: Cross-iteration batch normalization. In: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR). pp. 12331–12340 (2021)