

Training Data Generation for U-Net Based MRI Image Segmentation using Level-Set Methods

Şükrü Ozan¹

¹AdresGezgini AŞ Ar-Ge Merkezi Bayraklı, İzmir

Article Info

Keywords: Active contours, Deep neural networks, Evolving boundaries, Image segmentation, Level-set, MRI, Region of interest, U-net

2010 AMS: 00A69, 62M40, 62M45

Received: 19 April 2022

Accepted: 16 September 2022

Available online: 2 December 2022

Abstract

Image segmentation has been a well-addressed problem in pattern recognition for the last few decades. As a sub-problem of image segmentation, the background separation in biomedical images generated by magnetic resonance imaging (MRI) has also been of interest in the applied mathematics literature. Level set evolution of active contours idea can successfully be applied to MRI images to extract the region of interest (ROI) as a crucial preprocessing step for medical image analysis. In this study, we use the classical level set solution to create binary masks of various brain MRI images in which black color implies background and white color implies the ROI. We further used the MRI image and mask image pairs to train a deep neural network (DNN) architecture called U-Net, which has been proven to be a successful model for biomedical image segmentation. Our experiments have shown that a properly trained U-Net can achieve a matching performance of the level set method. Hence we were able to train a U-Net by using automatically generated input and label data successfully. The trained network can detect ROI in MRI images faster than the level-set method and can be used as a preprocessing tool for more enhanced medical image analysis studies.

1. Introduction

Since the deep neural networks started to perform human-level performance in the image classification tasks with ImageNet [1], many problems in computer vision have been solved by using application-specific DNN architectures. Moreover, today, it is also possible to find data-driven solutions to solve complex nonlinear partial differential equations using deep neural networks with a supervised learning approach. The recent related works [2], [3] and [4] shows the effectiveness of DNN based solution methods with some classical problems in the fields like fluid mechanics and quantum mechanics.

The most important contribution of the DNN based solution methods is to solve the related problems accurately without solving complex PDEs. Moreover, the trained DNNs make very fast predictions against previously unseen data during inference time. Hence, DNN based PDE solution methods are also more time-efficient than classical numerical and analytical solutions.

In this study, the MRI image segmentation problem is revisited. The aim is to propose a DNN based preprocessing framework that detects the region of interest in an MRI image. The idea is to find the boundary surrounding the corresponding organ in the image. We used a publicly available brain MRI image dataset [5] for the experiments.

We start with an efficient numerical solution called level-set methods. This method has been applied to various 3D computer graphics and 2D computer vision problems [6], [7]. In computer graphics, level-set methods effectively solve problems like surface reconstruction from unorganized noisy point clouds. In computer vision, this method has also been used for image segmentation problems in different digital image sources, like MRIs.

In the second phase of this study, we use the segmentation results of the renowned level-set method to train a deep neural network. Our architecture preference is the U-Net [8], a state-of-the-art DNN for image segmentation.

2. Material and Methods

2.1. Dataset

The data samples are the images from Brain Tumor Classification (MRI) dataset. The images are of size (256,256) RGB JPEG images. The dataset comprises four classes, three for different brain tumors called glioma, pituitary, and meningioma, and one for the normal brain. In this study, we do not address the tumor type classification problem. We instead concentrate on a more primer problem called ROI detection in brain MRI images.

In Figure 2.1, one sample for each class can be seen. The MRIs are taken from different segments and different orientations of the human skull. Hence, the sample regions in the images are not uniform in size, shape, and position. Moreover, since the images are obtained from different hardware, the images' dynamic range, resolution, and sharpness differ. These variations make the ROI detection problem a complex nonlinear problem that rule-based explicit programming approaches can not solve.

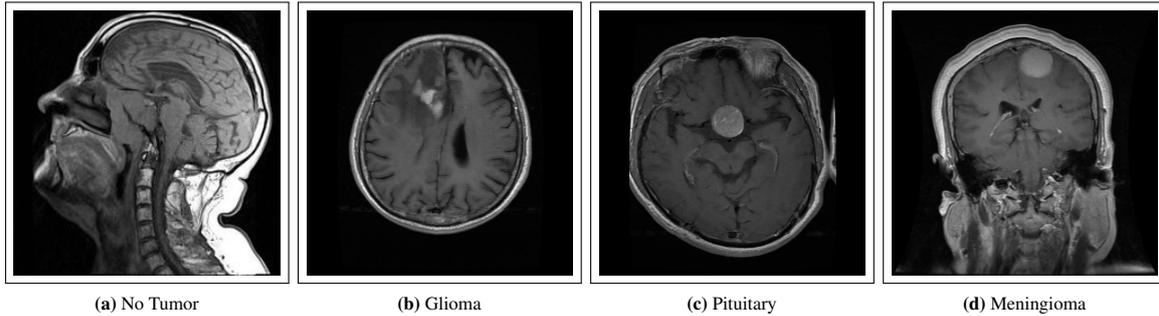


Figure 2.1: There are four basic classes in Brain Tumor Classification (MRI) dataset [5]. One sample selected from these classes and shown in the figure. (a) No tumor sample (b) Glioma tumor sample (c) Pituitary tumor sample (d) Meningioma tumor sample

To increase the number of samples in the dataset, we performed data augmentation by randomly combining the images and obtaining even more complex ROIs from plain brain MRIs. Some augmented samples can be seen in Figure 2.2. We primarily had 3172 images from the original dataset. After data augmentation, we increased the number of data to 6172.

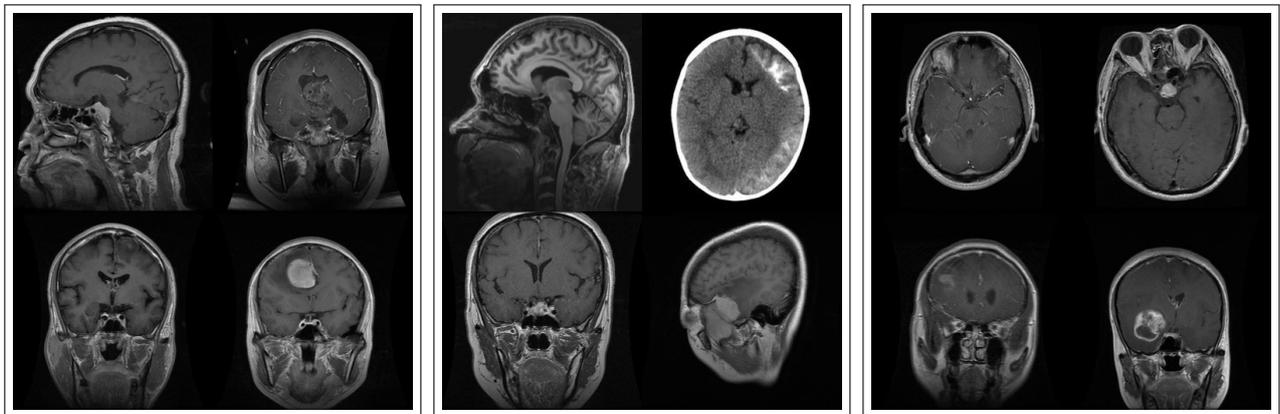


Figure 2.2: Augmented data samples

2.2. Implementation

The implementation procedure comprises two main parts called Level-Set and UNet-MRI, which are publicly available at the corresponding GitHub repositories ([9], [10], respectively) for researchers and enthusiasts who want to reproduce the reported results. The U-Net repository [10] also includes trained model parameters to be directly used for ROI segmentation in MRIs.

For the U-Net training, the input images are the dataset introduced in Section 2.1. The label images are generated by using the level-set-based image segmentation procedure, which will be explained in Section 3 in detail.

3. Theory

3.1. Evolving boundaries

We will apply the level set method to an initial implicit boundary. The implicit functions make it possible to capture complex curves without even explicitly defining them analytically. The function $\phi(\vec{x})$ can be defined on \mathbb{R}^n without loss of generality. The implicit representation idea can be depicted with the one-dimensional and two-dimensional cases in Figure 3.1. For $n = 3$ the implicit function represents a 3D surface. For example, the implicit function $\phi(\vec{x}) = x^2 + y^2 + z^2 - 1$ represents the unit sphere boundary $\partial\Omega = \{\vec{x} \mid |\vec{x}| = 1\}$ with the

exterior region $\Omega^+ = \{\vec{x} \mid |\vec{x}| > 1\}$ and the interior region $\Omega^- = \{\vec{x} \mid |\vec{x}| < 1\}$, at $\phi(\vec{x}) = 0$ isocontour, i.e. the zero level set of ϕ . Accordingly, we will be capturing the zero level set of an evolving curve at each iteration.

We first initialize an implicit function $\phi(x)$, such that it represents a rectangular area that ensures the enclosure of ROI. We want the implicit boundary of the box to evolve in time such that eventually, it will represent the ROI. In order to capture the surface evolution in time, we add a temporal variable t to ϕ . Hence the zero level set of temporal ϕ becomes Equation (3.1).

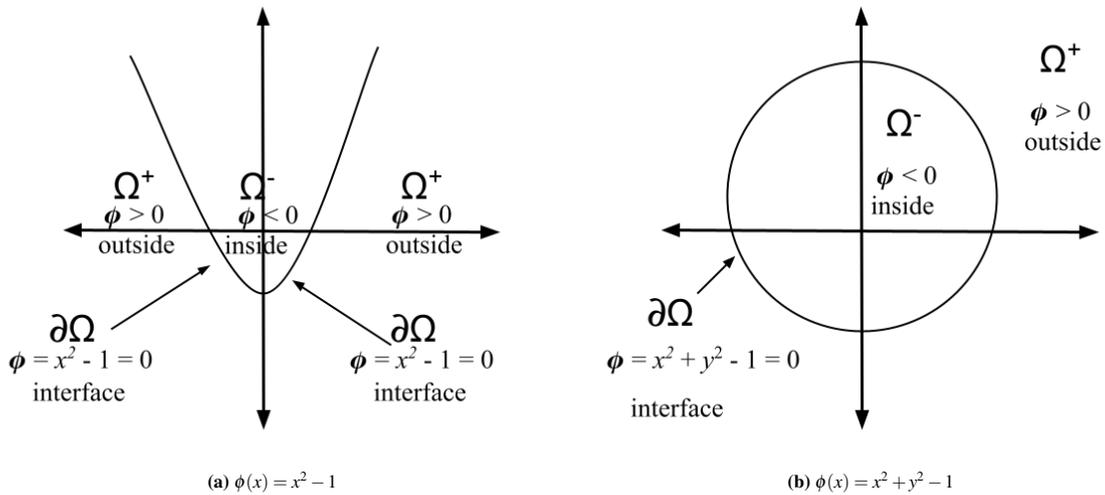


Figure 3.1: Implicit representations of two functions. (a) Implicitly defined function $\phi(x) = x^2 - 1$ defines the regions Ω^+ , Ω^- and the boundary $\partial\Omega$ (b) Implicitly defined function $\phi(x) = x^2 + y^2 - 1$ defines the regions Ω^+ , Ω^- and the boundary $\partial\Omega$

$$\phi(x(t), t) = 0 \tag{3.1}$$

In order to track the movement of the zero level set $\phi(x(t), t) = 0$, we have to take its derivative with respect to t . Since the implicit function represents the position, its average temporal change implies the velocity of each point in the computational domain. Considering the chain rule, derivative of Equation (3.1) becomes Equation (3.2).

$$\frac{\partial\phi}{\partial x(t)} \frac{\partial x(t)}{\partial t} + \frac{\phi}{\partial t} = 0 \tag{3.2}$$

As a further interpretation, we know that $\partial\phi/\partial x$ is the gradient of the curve, i.e., $\nabla\phi$. By following the notation convention in [11], we can rewrite the equation as can be rewritten as Equation (3.3). This form is known as level set equation [12]. F is called the speed function, which will be defined over the computation domain by the gradient of the MRI image.

$$\phi_t + F|\nabla\phi| = 0 \tag{3.3}$$

By using finite difference method, specifically forward differencing, the partial differential equation (3.3) can be reformulated as Equation (3.4). This final form is the evolution equation we will use through the iterations. At time t , ϕ' is the value of ϕ after next iteration at $t + \Delta t$.

$$\phi' = \phi + \Delta t F|\nabla\phi| \tag{3.4}$$

We want our initial surrounding box boundary to evolve so that it will eventually cover the ROI. Hence, we want the speed function F to be high outside the ROI and ideally zero at the ROI boundary. Concretely, deriving F from the edge features of the image is a convenient method. We can use the edge indicator function g in Equation (3.5), where ∇I is the gradient of the MRI image.

$$g(I) = \frac{1}{1 + \|\nabla I\|^2} \tag{3.5}$$

Two samples of generated F speed function images can be seen in Figure 3.2. As a preprocessing step, we subtract the mean intensity value of the image from each pixel. It eliminates measurement noise in the dark areas and makes it possible to obtain clear speed function images. The method needs a stopping condition to end the iterations. We use the mean square error between consecutive images representing ϕ and ϕ' and stop iterations if this value is smaller than a predetermined threshold value. The overall procedure can be followed using Algorithm 1.

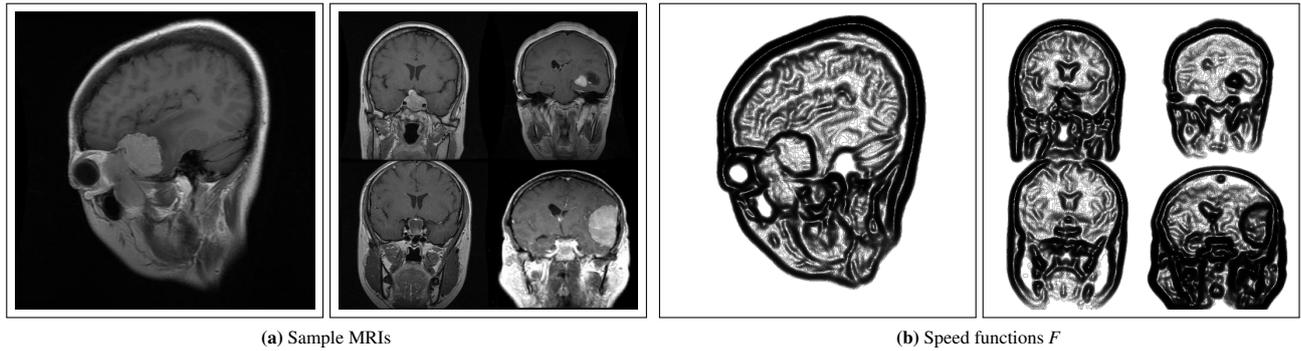


Figure 3.2: (a) Two sample MRI images from the dataset. (b) Speed function images generated from the sample MRI images in (a)

Algorithm 1 Level-set boundary evolution algorithm

- 1: $I \leftarrow$ an MRI from Dataset;
 - 2: $I \leftarrow I - \text{MEAN}(I)$;
 - 3: $F \leftarrow 1./ (1. + \|\nabla I\|^2)$;
 - 4: $\phi \leftarrow$ Initial box surrounding ROI;
 - 5: $\Delta \leftarrow$ A large value (e.g. $1e+15$);
 - 6: $\varepsilon \leftarrow$ A small value (e.g. $1e-15$);
 - 7: **while** $\Delta > \varepsilon$ **do**
 - 8: $\phi_{new} \leftarrow \phi + F|\nabla\phi|$;
 - 9: $\Delta \leftarrow \text{MSE}(\phi_{new}, \phi)$;
 - 10: $\phi \leftarrow \phi_{new}$;
 - 11: **end while**
 - 12: **return** ϕ ;
-

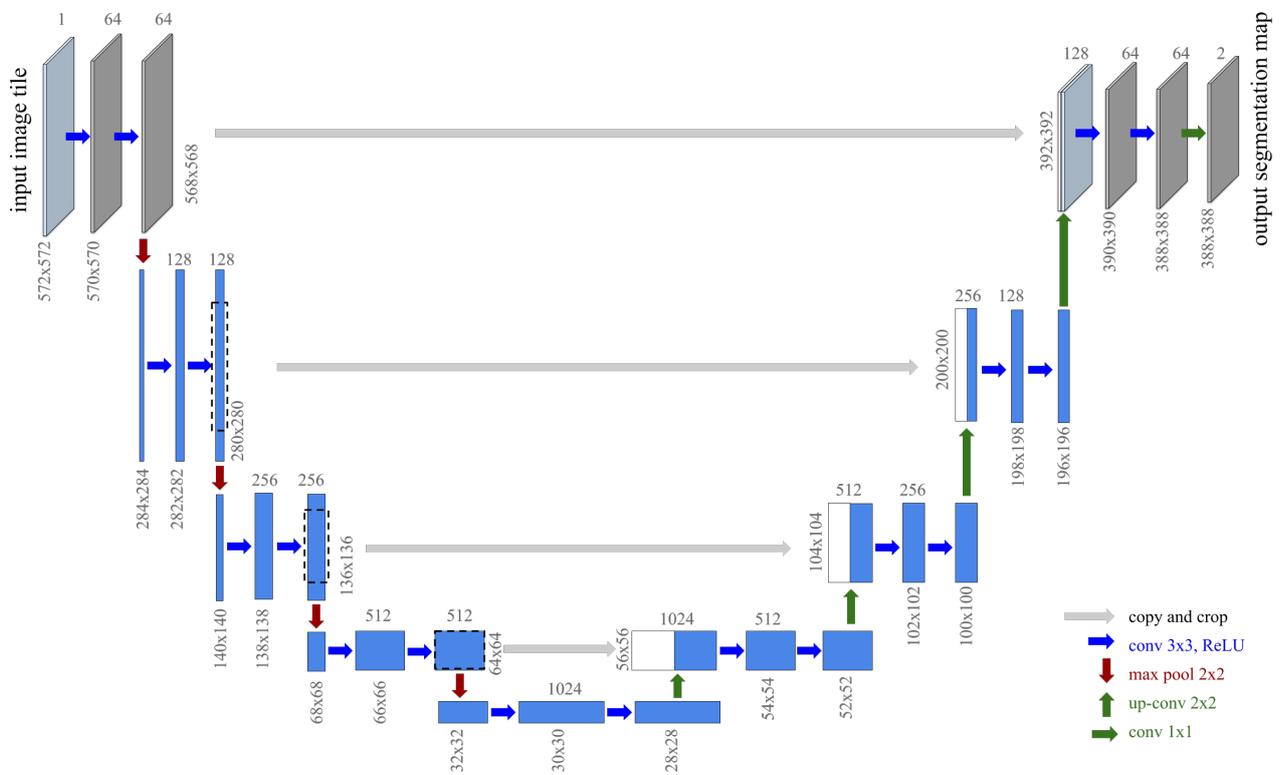


Figure 3.3: The re-depicted model architecture of the original U-Net model proposed in [8].

3.2. U-Net Training

The very essence of this work is to find a state-of-the-art alternative for the method described in Section 3.1. We chose the U-Net deep neural network model architecture, which has proven to be an effective model for image segmentation problems. The re-depicted model

architecture of the original model proposed in [8] can be seen in Figure 3.3. We used a slightly modified version of this architecture where we kept the input and output image sizes equal to 256 by 256. It resulted in a model with around 31 Million parameters.

The model training is performed by using the augmented dataset described in Section 2.1 as input. As output images, we use the mask images created by using the level set method, which is described in Section 3.1.

We used RMSProp [13] as an optimizer for the network training. Since the output is a binary image, we chose sparse categorical cross-entropy loss as the cost function. Different loss functions commonly used in deep learning are listed and compared in [14].

The data is split as %80 - %20 for training and validation, respectively. U-Net training does not need very long training epochs. Hence, we performed the training for ten epochs and achieved the best validation loss at the eighth epoch. We assured faster convergence by applying batch normalization [15], which prevents the neural network optimization deceleration due to covariate shift.

Covariate shift happens due to the complicated nature of deep neural networks. The input of each layer changes drastically as the parameters of the previous layers change. It lowers the adaptive learning rates, and hence the training eventually slows down. The related work [15] proposes Algorithm 2 which has been extensively used in deep learning literature recently.

The batch normalization is defined for each mini-batch since the RMSProp runs on mini-batches. An overview and comparison of different gradient descent procedures can be seen in [16].

Algorithm 2 Batch normalization (BN) applied to activation x over a mini-batch.

Input: Values of x over a mini-batch \mathcal{B} where $\mathcal{B} = \{x_1, \dots, x_m\}$

Parameters to be learned: γ, β

Output: $\{y_i = BN_{\gamma, \beta}(x_i)\}$

- 1: $\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$ #mini-batch mean
 - 2: $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2$ #mini-batch variance, small ϵ prevents division by zero
 - 3: $\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$ #normalize
 - 4: $y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i)$ #scale and shift
-

4. Results and Discussion

By using the update equation in Equation (3.4), the evolution of the initial boundary can be iterated. At each step, the exact boundary shape can be recovered by using the positive and negative regions,

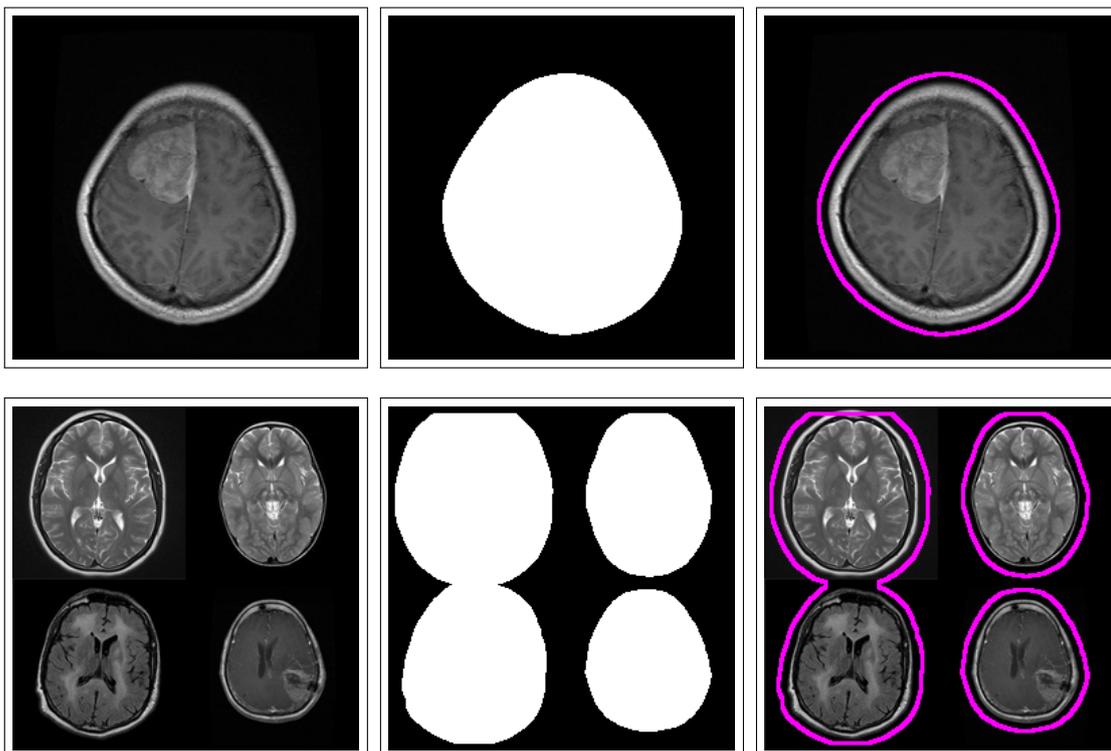


Figure 4.1: **Left column:** Sample MRIs from the dataset. **Middle Column:** Final state of the boundary after evolution iterations. The black and white colors represent the positive and negative regions, Ω^+ and Ω^- , of ϕ . **Right Column:** Shows the converged boundary (in magenta) on the original image.

Ω^+ and Ω^- , of ϕ . Three sample animated boundary evolutions can be seen in [Animation 1](#), [Animation 2](#) and [Animation 3](#). Moreover, two selected boundary evolution results can be seen in [Figure 4.1](#).

We applied the level-set-based image segmentation algorithm to all of the 6172 images in the dataset. Since the ROI size and complexity differs, the amount of iterations for these images varies. It took 7 minutes 53 seconds for [Algorithm 1](#) to detect RIO in the dataset images. The populated mask images constitute a new dataset.

In the next step, the corresponding image pairs in the original and mask images are used to train a tailored version U-Net architecture [8]. We kept the model parameters of the lowest validation loss and generated mask images for the validation set. In [Figure 4.2](#), mask generation results using both the level-set method and U-Net of two samples from the validation set can be seen. We observe that the U-Net generated mask image closely matches the level-set result.

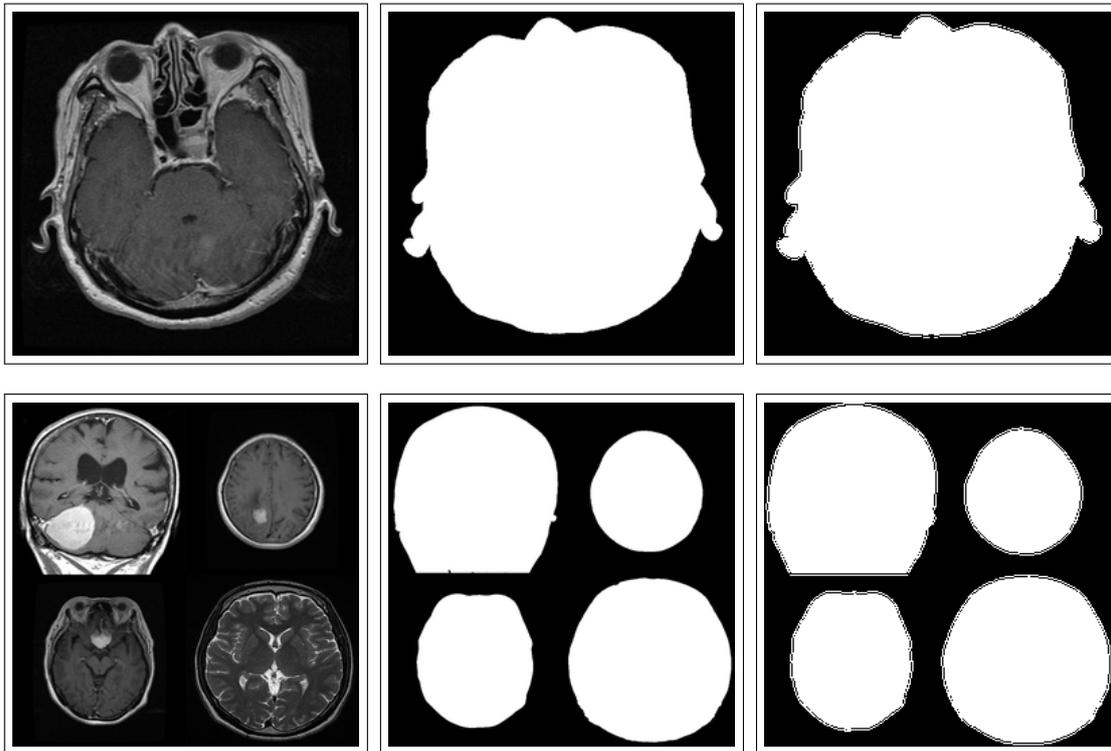


Figure 4.2: **Left column:** Sample MRIs from the dataset. **Middle Column:** The mask images representing the ROIs in sample images. **Right Column:** Shows the mask images generated by trained U-Net model.

To obtain a comparable processing time, we fed the network with all the images in the dataset. The trained U-Net DNN processed all the images in 1 minute 45 seconds, nearly five times faster than the level set method.

4.1. Conclusions

This study used a classical numerical solution for a specific image segmentation problem to create a training dataset for a deep neural network model. We chose the problem of ROI detection on MRIs since the ROI masks are essential for analyzing medical image analysis to optimize processing time [17], [18]. The ROIs on MRI slices can also be used for the 3D reconstruction of the organs [19].

We used the brain MRI dataset [5] which consists of MRI slices of tumourous and normal brains of different patients. We revisited the level-set solution to the image segmentation problem and successfully applied it to these MRIs. This numerical solution needs some preprocessing steps to obtain clear image gradients, which are to be used to specify the speed function F in the level set equation (3.3). By following the listed procedures in [Algorithm 1](#), we created a mask image of the ROI in each MRI in the dataset.

We showed that similar segmentation masks could also be obtained using a U-Net once trained adequately on an appropriate training dataset. We used the dataset images and their generated ROI masks to train our U-Net model architecture till the model overfit and kept the best parameters where we achieved the lowest validation loss at the eighth epoch. The inference results showed that U-Net could create closely matched ROI segmentation masks for input MRI images in the validation dataset.

Using a DNN to solve the corresponding segmentation problem did have three significant advantages over the classical numerical solution using level-sets. First, the input images can directly be processed by the U-Net without any need for image preprocessing. Second, the DNN based solution method did not require numerical solutions to a partial differential equation like the level-set equation. Lastly, the trained U-Net model could process images five times faster than the level-set-based method, which is the most critical aspect of the proposed method for practical use since it can save a significant amount of time for vast datasets.

Article Information

Acknowledgements: The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions.

Author's contributions: All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Conflict of Interest Disclosure: No potential conflict of interest was declared by the author.

Copyright Statement: Authors own the copyright of their work published in the journal and their work is published under the CC BY-NC 4.0 license.

Supporting/Supporting Organizations: No grants were received from any public, private or non-profit organizations for this research.

Ethical Approval and Participant Consent: It is declared that during the preparation process of this study, scientific and ethical principles were followed and all the studies benefited from are stated in the bibliography.

Plagiarism Statement: This article was scanned by the plagiarism program. No plagiarism detected.

Availability of data and materials: Not applicable.

References

- [1] A. Krizhevsky, I. Sutskever, G. Hinton *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems, **25** (2012), 1097-1105.
- [2] M. Raissi, P. Perdikaris, G. E. Karniadakis *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., **378** (2019), 686-707.
- [3] M. Raissi, P. Perdikaris, G. E. Karniadakis *Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations*, arXiv preprint arXiv:1711.10561 (2017).
- [4] M. Raissi, P. Perdikaris, G. E. Karniadakis *Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations*, arXiv preprint arXiv:1711.10566 (2017).
- [5] S. Bhuvaji, A. Kadam, P. Bhumkar, S. Dedge, S. Kanchan *Brain Tumor Classification (MRI)*, Kaggle (2020).
- [6] J. A. Sethian *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, Cambridge University Press (1999).
- [7] S. Osher, R. Fedkiw *Level set methods and dynamic implicit surfaces*, Appl. Math. Sci., **153** (2003).
- [8] O. Ronneberger, P. Fischer, T. Brox *U-net: Convolutional networks for biomedical image segmentation*, International Conference on Medical image computing and computer-assisted intervention, **9351** (2015), 234-241.
- [9] Ş. Ozan *Level set based contour evolution on brain MRI images*, GitHub, <https://github.com/sukruozan/level-set> (2021).
- [10] Ş. Ozan *Unet training for MRI segmentation*, GitHub, <https://github.com/sukruozan/unet-mri> (2021).
- [11] C. Li, C. Xu, C. Gui, M.D. Fox *Level set evolution without re-initialization: A new variational formulation*, IEEE Trans. Image Process., **14** (2005), 2098-2104.
- [12] S. Osher *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., **79** (1988), 12-49.
- [13] T. Tieleman, G. Hinton *Lecture 6.5 - rmsprop*, coursera, Coursera (2012).
- [14] J. Katarzyna, W. M. Czarnecki *On Loss Functions for Deep Neural Networks in Classification*, arXiv preprint arXiv:1702.05659 (2017).
- [15] S. Ioffe, C. Szegedy *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, Proceedings of the 32nd International Conference on Machine Learning, **37** (2015), 448-456.
- [16] S. Ruder *An overview of gradient descent optimization algorithms*, arXiv preprint arXiv:1609.04747 (2016).
- [17] D. Pilutti, M. Buchert, S. Hadjidemetriou *Registration of abdominal tumor DCE-MRI data based on deconvolution of joint statistics*, 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2611-2614.
- [18] X. Xu, T. Meng, Q. Wen, M. Tao, P. Wang, K. Zong, Y. Shen *Dynamic changes in vascular size and density in transgenic mice with Alzheimer's disease*, Impact Journals LLC, **12** (2020), 17224-17234.
- [19] S. Urvashi, S. Meenakshi, P. Emjee *Region of interest based selective coding technique for volumetric MR image sequence*, Multimedia Tools and Applications, **80** (2021), 1-23.