2023 IEEE 7th Portuguese Meeting on Bioengineering (ENBENG)
22-23 June, 2023. Porto, Portugal

# Generative Adversarial Networks in Healthcare: A Case Study on MRI Image Generation

1st Beatriz Cepa
*Department of Informatics*
*INESC TEC and University of Minho*
Braga, Portugal
beatriz.cepa@inesctec.pt

2nd Cláudia Brito
*Department of Informatics*
*INESC TEC and University of Minho*
Braga, Portugal
claudia.v.brito@inesctec.pt

3rd António Sousa
*Department of Informatics*
*INESC TEC and University of Minho*
Braga, Portugal
antonio.l.sousa@inesctec.pt

2023 IEEE 7th Portuguese Meeting on Bioengineering (ENBENG) | 979-8-3503-2257-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ENBENG58165.2023.10175330

***Abstract*—Medical imaging, mainly Magnetic Resonance Imaging (MRI), plays a predominant role in healthcare diagnosis. Nevertheless, the diagnostic process is prone to errors and is conditioned by available medical data, which might be insufficient. A novel solution is resorting to image generation algorithms to address these challenges. Thus, this paper presents a Deep Learning model based on a Deep Convolutional Generative Adversarial Network (DCGAN) architecture. Our model generates 2D MRI images of size 256×256, containing an axial view of the brain with a tumor. The model was implemented using ChainerMN, a scalable and flexible framework that enables faster and parallel training of Deep Learning networks. The images obtained provide an overall representation of the brain structure and the tumoral area and show considerable brain–tumor separation. For this purpose, and owing to their previous state-of-the-art results in general image-generation tasks, we conclude that GAN-based models are a promising approach for medical imaging.**

***Index Terms*—Deep Learning, Generative Adversarial Networks, Image Generation, Medical Images**

## I. Introduction

It is known that an accurate medical diagnosis is fundamental for patient treatment, and the faster the patient is diagnosed, the greater the probability of successful treatment. Medical images, mainly those from Magnetic Resonance Imaging (MRI), play a predominant role in diagnosing and providing healthcare because decision-making by doctors frequently relies on their interpretation [1]. Nevertheless, diagnostic error is inherent to this process and may bias medical judgment and decisions regarding the (possible) pathology observed, as well as the type, period, and options of treatment. One way to achieve better diagnostic accuracy (and consequently reduce diagnostic errors) is to use computer-aided systems [2]. Hence, computer-aided diagnosis – and, focusing on medical images, the existence of Deep Learning (DL) models – presents a viable option compared to manual diagnostics to attempt to reduce diagnostic errors. However, the publicly available medical data for model training and research are limited because they depend on the number of medical imaging scans performed in a healthcare institution. Data augmentation techniques have been extensively used to expand datasets [3], and can be used to mitigate this problem. Data augmentation "encompasses a suite of techniques that enhance the size and quality of training datasets" [4] and can be divided into two groups: oversampling and data warping. Some data augmentation techniques include flipping, rotation, cropping, noise injection, and translation [4]. Image generation with Deep Generative Models may present itself as a data augmentation strategy because this type of model enables the generation of artificial samples from a dataset such that they embody similar features to those of the original image set. Generative Adversarial Networks (GANs), for instance, have received increased interest and attention for their application in data augmentation because of their ability to synthesize new training samples [3], [4].

Medical image analysis is closely related to a high volume of data; for each patient, hundreds of image slices may be generated using diverse imaging techniques and dimensions (e.g., time series and 3D images). Each slice is generally stored as an individual image in the Digital Imaging and Communications in Medicine (DICOM) [5] format or as a volume in the Neuroimaging Informatics Technology Initiative (NIfTI) format [6] and might have a significant size. Moreover, this type of image is typically more challenging to analyze than generic images because of the complex parameters (such as contrast agents and specific settings) and anatomic differences. Traditional methods become obsolete when deployed to large-scale medical data environments, whether because of the data volume or the long processing time associated with it [7]. With this, the use of distributed techniques and frameworks is an alternative to address these challenges, as it provides fast computing while handling large volumes of data.

In this study, we developed a DL model that can be used to address the scarcity of data available for model training. Thus, we design and implement an image generation model that can be used as groundwork to be deployed in distributed environments. The DL model should receive as input a medical image set (of MRI images) containing some abnormality, pathology, or suspicious area and, from those images, would generate new images that may then be stored in a database and be available for consulting. Consequently, it is expected

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020 (Beatriz Cepa) and through a Ph.D. Fellowship SFRH/BD/-146528/2019 (Cláudia Brito)

to obtain a set of new images – realistic and analogous to the original ones – so that more data can be provided to train new DL models in the identification of specific conditions and, therefore, improve medical care.

## II. Our Approach

### A. Data and Pre-Processing

We used training data from the Brain Tumor Segmentation (BraTS) 2020 Dataset [8]–[10]. It is composed of "routine clinically-acquired pre-operative multimodal MRI scans of glioblastoma (GBM/HGG) and lower grade glioma (LGG), with a pathologically confirmed diagnosis", acquired using various scanners from 19 institutions and with different clinical protocols [11]. The scans are available as NIfTI [6] files and include T1, post-contrast T1-weighted, T2-weighted, and T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes from 369 subjects. All of the data are "co-registered to the same anatomical template, interpolated to the same resolution ($1\ mm^3$) and skull-stripped" [11].

We selected the T2-FLAIR volumes from 15 subjects (subject ID 001 to 015) and sliced them by the axial plane using the 3D Slicer software [12]. From the 155 slices obtained for each subject, only 48 (slices #60 to #107) were selected due to their better representation of the brain and tumor structures, yielding a total of 720 images. GAN outputs are typically square, and since their original size was also approximately square, we considered that resizing the images would not have a significant impact on the proportionality of the brain and tumor structure. Thus, they were resized to 256×256 pixels.

### B. Developed Model

The model developed in this paper consists of an implementation based on the Deep Convolutional Generative Adversarial Network (DCGAN) architecture proposed by Radford et al. [13]. The Generator (G) receives as input a 128-dimensional distribution and is composed of a fully-connected layer and seven 2D transposed convolutional layers, with the initial seven layers being followed by a batch normalization layer. The activation is with Leaky ReLU ($\alpha = 0.2$) in the first six layers and Tanh in the last convolutional layer. Furthermore, as suggested by Isola et al. [14], dropout was applied for every layer of G. This architecture is summarized in Table I.

The Discriminator (D), in turn, consists of nine 2D convolutional layers and a fully-connected layer, where each layer is followed by batch normalization, except for the input and fully-connected layers. ReLU activation is used in the initial eight convolutions and Sigmoid in the last. Moreover, noise samples from a Normal distribution were added as input in every layer, as suggested by Arjovsky et al. [15]. This architecture is summarized in Table II.

### C. Training Parameters

The model was trained using the Wasserstein loss concept [16] and a batch size of eight. The weights were initialized as proposed by Radford et al. [13], that is, from a zero-centered Normal distribution with a standard deviation of 0.02. The optimization algorithm was Adam [17] with $\alpha = 0.0002$, $\beta_1 = 0.5$, and a Weight Decay rate of 0.001 for both the Generator and the Discriminator.

The DL model was implemented using the ChainerMN framework [18], enabling the distribution of the dataset and all parameters associated with the training process by several nodes. The training process was performed using an NVIDIA GeForce RTX 2070 SUPER GPU.

## III. Results and Discussion

The model generates several stacks of 100 3-channel 2D images of size 256×256 pixels, each containing an axial view of a brain with a tumor, where, similarly to the original T2-FLAIR MRI image scans, the gray area represents the brain tissue and the white area represents the tumoral tissue. Fig. 1 shows images from the original dataset and synthetic images generated by the model, while Fig. 2 shows a comparison between images generated in the initial and final experiments (the initial experiments were performed without dropout, with a 100-dimensional input distribution and a batch size of 64, while the final experiments were done with the architecture and parameters described in Section II).

The generated images provided an overall representation of the brain structure and tumoral area. As can be observed, these images possess a noticeable grid pattern rather than being smooth and uniform. This is a result of the transposed convolutions on the Generator architecture used to perform upsampling of the input images. As an attempt to diminish that

TABLE I
Summary of the Generator architecture, where T. Conv is Transposed Convolution

| Generator | Activation | Output shape |
|---|---|---|
| **Input noise** | — | 128 × 1 × 1 |
| **Fully-connected layer** | — | 16384 |
| **Reshape** | — | 1024 × 4 × 4 |
| **T. Conv 4 × 4** | Leaky ReLU | 512 × 8 × 8 |
| **T. Conv 4 × 4** | Leaky ReLU | 256 × 16 × 16 |
| **T. Conv 4 × 4** | Leaky ReLU | 128 × 32 × 32 |
| **T. Conv 4 × 4** | Leaky ReLU | 64 × 64 × 64 |
| **T. Conv 4 × 4** | Leaky ReLU | 32 × 128 × 128 |
| **T. Conv 4 × 4** | Leaky ReLU | 3 × 256 × 256 |
| **T. Conv 3 × 3** | Tanh | 3 × 256 × 256 |

TABLE II
Summary of the Discriminator architecture

| Discriminator | Activation | Output shape |
|---|---|---|
| **Input** | — | 3 × 256 × 256 |
| **Conv 4 × 4** | ReLU | 32 × 128 × 128 |
| **Conv 4 × 4** | ReLU | 64 × 64 × 64 |
| **Conv 4 × 4** | ReLU | 128 × 32 × 32 |
| **Conv 4 × 4** | ReLU | 156 × 16 × 16 |
| **Conv 3 × 3** | ReLU | 156 × 16 × 16 |
| **Conv 4 × 4** | ReLU | 512 × 8 × 8 |
| **Conv 3 × 3** | ReLU | 512 × 8 × 8 |
| **Conv 4 × 4** | ReLU | 1024 × 4 × 4 |
| **Conv 3 × 3** | Sigmoid | 1024 × 4 × 4 |
| **Fully-connected layer** | — | 1 |

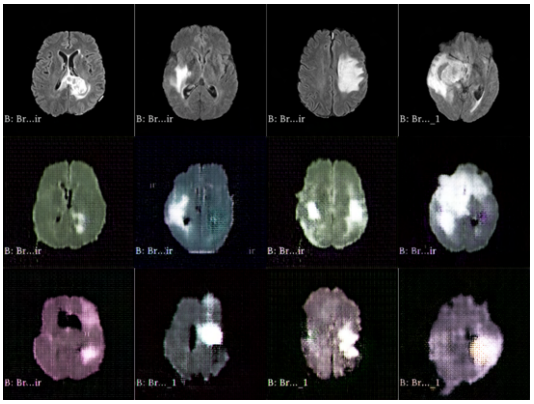

Fig. 1. Comparison between real MRI images (upper row), success cases (middle row), and failure cases (lower row)

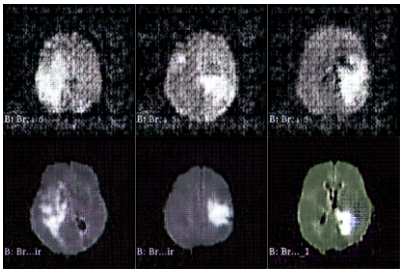

Fig. 2. Images generated in the initial (upper row) and final (lower row) experiments

"checkerboard" effect, we followed Odena et al.'s [19] suggestion and added the last convolutional layer to the Generator (represented in Table I by **T. Conv 3 $\times$ 3**). Nevertheless, it resulted in only a slight improvement (not as much as desired).

The images show considerable brain-tumor separation, achieved with changes in the activation functions of Generator layers. Leaky ReLU on the hidden layers yielded better results than ReLU since it substantially improved the distinction between the two areas and the image definition (Fig. 3). On the last convolution, between Tanh and Sigmoid, the latter resulted in blurrier images and less evident borders, so the activation chosen was Tanh.

The image noise was initially extremely high; consequently, it was necessary to apply conjugated changes to the Generator and Discriminator. In the former, the use of dropout on all layers significantly decreased the noise in the generated images, as shown in Fig. 4. In the Discriminator, that aspect was diminished by altering the activation functions: on hidden layers, ReLU yielded better results compared to ELU or Leaky

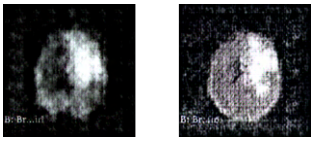

Fig. 3. Images generated by the model with ReLU (left) and Leaky ReLU (right) activation

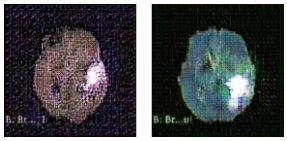

Fig. 4. Images generated before (left) and after (right) using dropout in G

ReLU, whereas on the output, Sigmoid outperformed Tanh.

Some training parameters also influenced the noise in the generated images. Training with a batch size of 8 instead of 16 or 64 was preferable because it yielded a reduction in image noise. A similar effect was observed when using the Adam optimizer rather than the Stochastic Gradient Descent (SGD) in both the Generator and Discriminator (Fig. 5), with $\alpha = 0.0002$ resulting in better image quality and more stable performance compared to $\alpha = 0.001$ or $\alpha = 0.005$. Additionally, we found that using 128 units as input noise for the Generator led to better performance than 100 or 256 units and that training with 150 epochs resulted in an "overtraining" of the model, i.e., the samples generated were of poor quality due to excessive training.

Mao et al. [20] proposed the use of Mean Squared Error (MSE) as the loss function in G and D; however, in this particular model, it resulted in images simply consisting of white blurs (Fig. 6). Hence, we opted for the Wasserstein [16] loss concept, which is implemented by multiplying the loss of real samples by -1 and the loss of generated samples by 1. This allowed the polarization of the scores for real and generated images, increasing their distance [21], [22].

However, our findings have some limitations regarding image quality, brain borders, and brain-tumor borders. The grid effect resulting from the transposed convolutions might be a reason for these limitations because it gives a pixelated appearance to the images. Moreover, the Wasserstein loss function [16], which has not been fully implemented, as suggested by the authors, may also contribute to a decrease in quality and definition. The amount of data used for training is another possible cause for the lack of detail in the generated images because GANs are usually trained with thousands of image slices. It is also possible that the values set for

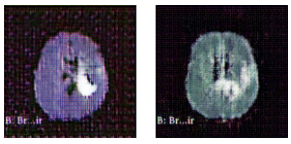

Fig. 5. Images generated with SGD (left) and Adam (right) optimization

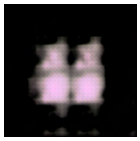

Fig. 6. Sample generated with MSE as the loss function

the hyperparameters were not the most adequate for this specific model, which would require further experimentation for verification.

It was observed that slight changes in the hyperparameters had a significant impact on the performance, and consequently, on the quality of the results. Moreover, we noticed that the last epochs of training yielded results consistent with "mode collapse" on the Generator, so it became evident that more training time does not necessarily lead to better results.

In general, the model was able to generate images with different axial views of the brain and containing tumors located in various zones. We were able to validate two hypotheses:

- We successfully implemented a complex model using an adequate framework for distributed training;
- Although the training data is inferior to that of typical image generation with GANs, we were able to obtain results that allow clear recognition of the represented structures and include images representing different axial views of the brain and tumors located in various areas.

## IV. Conclusions and Future Work

In healthcare, the need for more imaging data to improve the diagnostic process is a recurring problem that may be solved by generating new images. While resembling real images, these new images could serve the training purpose within the research and development of DL models for computer-aided diagnosis.

Following this, we developed a DL model based on the DCGAN architecture that receives real MRI scans as input and generates 2D MRI images of size 256×256 that contain an axial view of the brain with a tumor. Our findings provide a general representation of the brain structure and the tumoral area and show an easy differentiation between the two structures. Moreover, they are diverse in terms of axial views and tumor location.

Current limitations can be addressed in future work. First, the grid effect should be minimized. This can be achieved by changing the transposed convolutions so the kernel size is divisible by the stride (thus removing overlaps) or by using a different upsampling strategy (e.g., upsample layer followed by convolutional layer). Secondly, it is important to obtain high-definition images with more visible details and clearer borders between the brain and the tumoral tissue. Third, theoretically, a larger amount of input images should allow for improving the model performance. Also, this work should accompany clinical validation of the results, for example, with visual Turing tests applied to medical professionals.

Finally, the distribution of DL workloads is a hot research area when applied to the healthcare environment. This work builds upon a DL framework that allows resorting to distributed environments, namely High-Performance Computing (HPC). An important future direction is to experimentally validate the previous model in these fully-distributed environments and with real-world scenarios.

## V. Acknowledgments

The experimental dataset involving human subjects described in this paper was approved for research use by the Center for Biomedical Image Computing and Analytics (CBICA).

## References

[1] R. M. Reilly, "Magnetic Resonance Imaging (MRI) Technology", in *Medical Imaging for Health Professionals: Technologies and Clinical Applications*, R. M. Reilly, Ed. 2019, pp. 87–105.

[2] A. Meyer-Baese and V. Schmid, "Introduction", in *Pattern Recognition and Signal Analysis in Medical Imaging*, Elsevier, 2014, pp. 1–20.

[3] R. Osuala et al., "Data synthesis and adversarial networks: a review and meta-analysis in cancer imaging", in *Medical Image Analysis*, vol. 84, no. 102704, Feb. 2023.

[4] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *J. Big Data*, vol. 6, no. 1, pp. 1–8, Dec. 2019.

[5] "About DICOM: Overview", [Online]. Available: https://www.dicomstandard.org/about-home.

[6] Data Format Working Group (DFWG), "NIfTI: Neuroimaging Informatics Technology Initiative", Dec. 18, 2013, [Online]. Available: https://nifti.nimh.nih.gov/.

[7] Z. Li, X. Zhang, H. Müller, and S. Zhang, "Large-scale retrieval for medical image analytics: a comprehensive review", *Med. Image Anal.*, vol. 43, pp. 66–84, Jan. 2018.

[8] B. H. Menze et al., "The multimodal brain tumor image segmentation benchmark (BRATS)", *IEEE Trans. Med. Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.

[9] S. Bakas et al., "Advancing the Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Sci. Data*, vol. 4, Sep. 2017.

[10] S. Bakas et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge", Nov. 2018.

[11] Center for Biomedical Image Computing & Analytics (CBICA), "Multimodal Brain tumor Segmentation Challenge 2020: Data", 2020, [Online]. Available: https://www.med.upenn.edu/cbica/brats2020/data.html.

[12] BWH and 3D Slicer contributors, "3D Slicer image computing platform", Apr. 08, 2022, [Online]. Available: https://www.slicer.org/.

[13] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", Jan. 2016.

[14] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, and B. A. Research, "Image-to-image translation with conditional adversarial networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1125–1134.

[15] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks", 2017.

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks", in *Proceedings of the 34th International Conference on Machine Learning*, Aug. 2017, pp. 214–223.

[17] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", Dec. 2014.

[18] Preferred Networks, "Distributed Deep Learning with ChainerMN", 2015, [Online]. Available: https://docs.chainer.org/en/stable/chainermn/index.html.

[19] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and Checkerboard Artifacts", Distill, vol. 1, no. 10, Oct. 2016.

[20] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2794–2802.

[21] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study", in *Advances in Neural Information Processing Systems*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2020, pp. 700–709.

[22] J. Brownlee, "A Gentle Introduction to Generative Adversarial Network Loss Functions", Sep. 02, 2019, [Online]. Available: https://machinelearningmastery.com/generative-adversarial-network-loss-functions/.