AUGMENTATION OF MEDICAL IMAGING DATA USING CONSINGAN

*Maitar Asher

Columbia University

ABSTRACT

The primary objective of this project is to leverage the capabilities of ConSinGAN, a single-image synthesis technique, for artificially introducing cancer metastases into medical scans. Additionally, I showcase the capability to remove cancer metastases from medical scans when necessary. This tackles the common challenge of having a restricted availability of annotated data containing cancer metastases for training deep-learning models. Through the augmentation of existing datasets, the project seeks to improve the performance of neural network models in identifying cancerous regions within medical images.

1 Introduction

In prior studies, researchers extensively delved into and showcased the capabilities of Generative Adversarial Networks (GANs) [1], and Conditional GANs (CGANs) [2]. They demonstrated the efficacy of these models in producing high-quality visual samples through training on specific datasets. The introduction of SinGAN [3] marked a significant advancement, showcasing its unique ability to generate high-quality images from a single training image, making it invaluable for scenarios with limited data or specific image constraints. By leveraging the internal statistics of patches within a single image, SinGAN creates a powerful generative model capable of producing diverse high-quality image samples. The architecture involves a pyramid of fully convolutional lightweight GANs, each responsible for capturing patch distributions at different scales, enabling the generation of images with arbitrary dimensions that semantically resemble the training image while incorporating new object configurations and structures. SinGAN accomplishes diverse image manipulation tasks, such as paint-to-image, editing, harmonization, super-resolution, and animation, by leveraging a unified learning framework that models the internal patch distribution within a single natural image. ConSinGAN [4], an evolution of SinGAN, further refines the landscape of image generation. Through strategic architectural and training modifications such as parallel training with different learning rates, direct feature propagation between stages, and an enhanced rescaling approach, ConSinGAN achieves remarkable improvements in generating realistic images while significantly reducing overall training time compared to the original SinGAN framework. This present work extends the application of SinGAN and ConSinGAN, exploring their capacity to artificially introduce metastases to medical scans and remove them. This project focuses on the independent learning of 2D slices, and future work aims to extend ConSinGAN to 3D, expanding its capabilities for volumetric image synthesis.

* This project stems from a collaborative idea with Jonathan Fisher. With his permission, I implemented and researched a portion of the work. I would like to express my gratitude to Fisher for his valuable insights and contributions to the conceptualization of this work. This project is anticipated to serve as a foundation for future collaborative research between Jonathan Fisher and myself.

2 MATERIALS AND METHODS

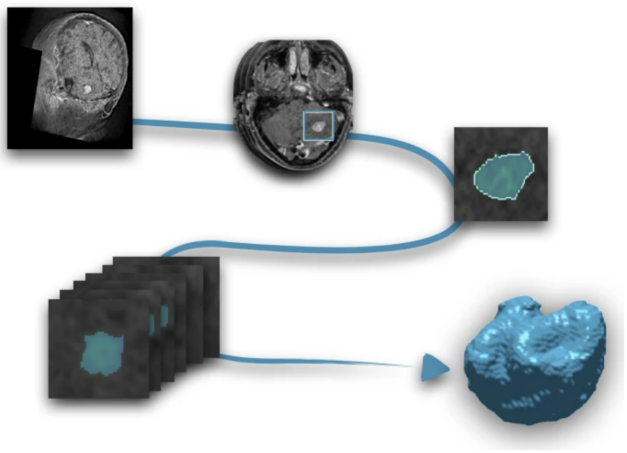

Fig. 1. An illustrative depiction of Brain Metastasis Image Segmentation sourced from the dataset.

2.1 Dataset

This project leverages a dataset [5] comprising magnetic resonance (MR) images of brain metastases (BMs) from 75 patients, accompanied by semi-automatic segmentations of 593 BMs. Rigorous validation by an expert radiologist, in collaboration with experienced medical image specialists, en-

sured the reliability of each segmentation. The dataset provides a nuanced understanding of BMs, as each segmentation mask includes two labels: those ending in 1 denote contrast-enhancing (CE) regions of the tumor, while those ending in 2 represent the non-enhancing or necrotic areas. Every selected MRI scan, along with its corresponding segmentation mask, underwent a transformation in which each 3D scan was converted into multiple 2D scans. Due to resource and time constraints, only a subset of these 2D scans was utilized for this project. All slices used in the methodology were resized to a consistent dimension of 250 by 250 pixels. This standardization was necessary due to variations in sizes across the training data and aligns with the ConSinGan model's learning capability, which effectively scales images up to a resolution of 250 by 250 pixels.

2.2 Architectural Insights and Image Manipulation Capabilities of ConSinGAN

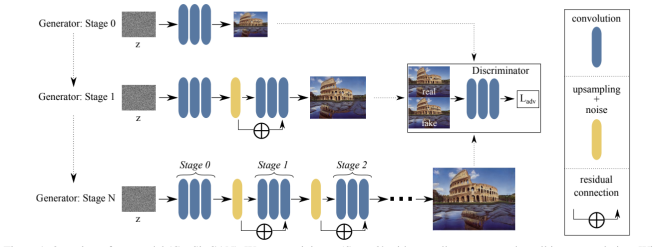

Fig. 2. Overview of ConSinGAN Model: Training initiates at 'Stage 0' with a small generator and low image resolution. As the number of stages increases, both the generator capacity and image resolution progressively increase [4].

In contrast to SinGAN, which focuses solely on training the current (highest) stage of its generator and freezing the parameters of all previous stages, the ConSinGAN model adopts a multi-stage training approach. It starts training at a low resolution and enhances the generator in successive stages by incorporating raw features from the previous stage. A progressive growing technique is applied by concurrently training multiple, but not all, stages with decreasing learning rates at lower stages to avoid overfitting. The ConSinGAN model maintains the same patch discriminator architecture and loss function as SinGAN [3]. At each stage, the optimization involves minimizing both an adversarial loss and a reconstruction loss. For a more comprehensive understanding, kindly refer to the original paper [4].

$$\min_{G_n} \max_{D_n} \mathcal{L}_{\text{adv}}(G_n, D_n) + \alpha \mathcal{L}_{\text{rec}}(G_n)$$

The capabilities of ConSinGAN and SinGAN in image manipulation span a spectrum of tasks, including generating diverse image samples, harmonizing images, achieving super-resolution, and handling animation. In the context of this project, I specifically concentrate on two manipulation tasks: harmonizing an external object into a given image and editing an image. The visual examples in Figure 3 and 4 provide a clearer illustration of these tasks. Harmonization involves seamlessly integrating an external object, exemplified by the focus on harmonizing tumors into specific image slices. The model will be trained on a tumor-free image slice, and the naive input for this task will be the same slice but with the tumor naively pasted. On the other hand, editing entails the removal of objects, and in this case, the task is to eliminate tumors from designated image slices. Thus the model will be trained on slices containing tumors, and the input image for the task will be the same slice but with the tumor naively removed.

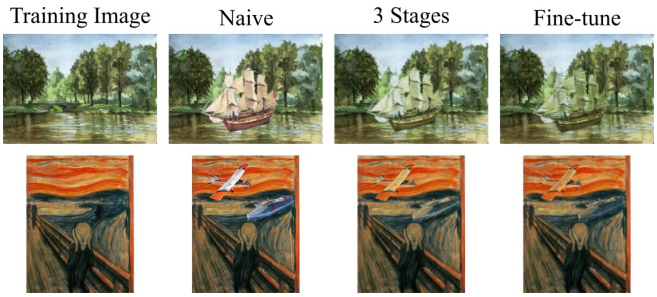

Fig. 3. Harmonization Manipulation Task using ConSinGAN [4].

Fig. 4. Editing Manipulation Task using ConSinGAN [4].

2.3 Methodology for Naively Adding Metastases

In the process of incorporating metastases into brain MRI scans, I explored various techniques. The foundational method revolves around utilizing a binary tumor mask, represented as a matrix, to selectively blend tumor pixels with the neighboring non-tumor pixels. This blending is accomplished through the following formula (refer to Figure 5):

$$\text{NaiveSlice} = \text{TumorMask} \times \text{SliceWithTumor} +$$

$$(1 - \text{TumorMask}) \times \text{SliceWithoutTumor}$$

Alternatively, another approach involves displacing the original tumor mask by specifying row and column offsets. This slight shift aims to simulate a natural variation in tumor location within the brain region, as depicted in Figure 6.

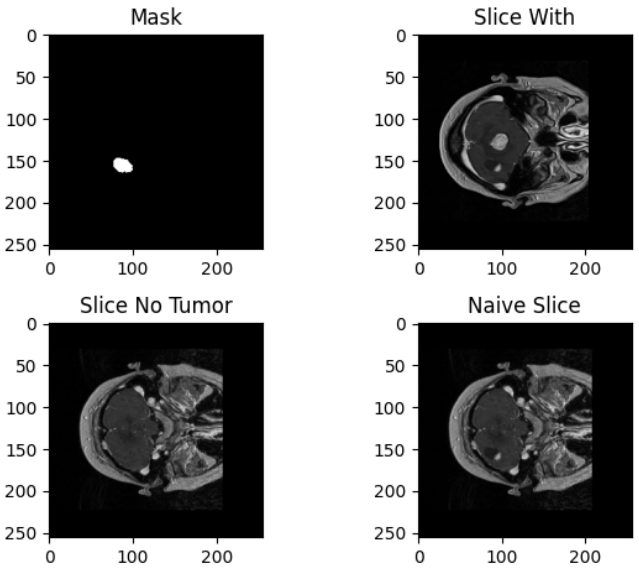

Fig. 5. Introducing Tumor in the Same Locations Naively

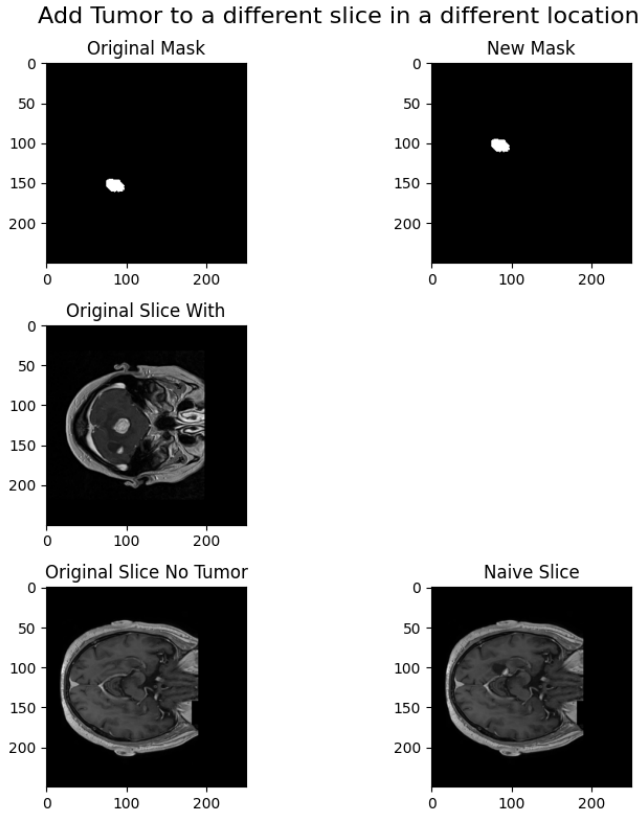

Fig. 6. Introducing Tumor in Different Locations Naively

2.4 Methodology for Naively Removing Metastases

To seamlessly eliminate metastases from brain MRI scans, I employ a simple yet effective methodology. The core idea revolves around generating a mirrored mask, achieved through the application of **numpy.flipud** – a NumPy function designed to vertically flip arrays. Specifically, I use **numpy.flipud** on the original tumor mask, transferring the indexes from this mirrored counterpart to the original mask. This strategic copying process essentially substitutes the existing tumor distribution with its mirrored representation, thereby accomplishing the removal of metastases. Refer to Figure 7.

3 Results

In this study, I evaluated the effectiveness of ConSinGAN with varying parameters in two tasks: harmonization and editing of brain MRI slices. For a comprehensive and quantitative assessment, I employed the **Fréchet Inception Distance (FID) score** (used FID Score for Py-Torch [6]). FID is a well-established metric in the field of generative models, particularly suited for tasks like image synthesis and generation. It quantifies the quality and diversity of generated images by measuring how closely they align with the distribution of real images. It's essential to note that FID scores typically range from 0 to positive infinity, with lower scores indicating better performance. A lower FID score implies that the generated images closely match the distribution of real images, signifying higher quality and diversity. While there isn't a universally defined threshold for what constitutes a "good" FID score, generally, scores closer to zero are desirable, suggesting a more faithful reproduction of real image characteristics by the generative model. In practice, a lower FID score reflects a higher degree of similarity between generated and real images, indicating superior performance in tasks such as image synthesis or generation.

3.1 Harmonization Task Results

I initially trained nine slices without tumors using two different configurations, resulting in a total of 18 models. The configurations were as follows:

Configuration 1:

- Stages: 6
- Learning Rate Scaling: 0.1
- Minimum Size: 120
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$21 minutes per model

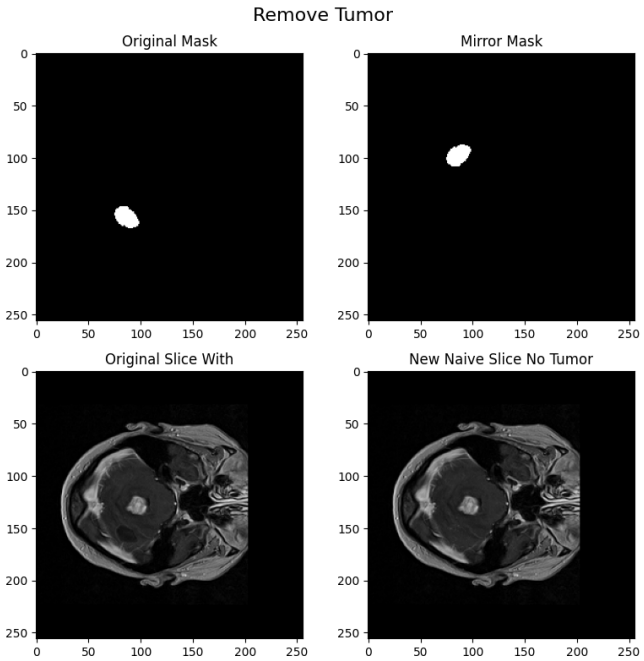

Fig. 7. Removing Tumor Naively

Configuration 2:

- Stages: 3
- Learning Rate Scaling: 0.1
- Minimum Size: 120
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$9 minutes per model

A brief explanation of the parameters:

- **Stages:** Increasing the number of stages in the model enhances its capacity to learn complex patterns. With a higher number of stages, the model can capture intricate details, textures, and stylistic nuances in the input images. However, this comes at the cost of increasing model complexity and training time.

- **Learning Rate Scaling:** The learning rate scaling involves adjusting the learning rates for each stage during training. A lower learning rate is applied to earlier stages to reduce overfitting. A factor δ is introduced, where stage n is trained with a learning rate $\delta^0\eta$. Stage $n-1$ is trained with a learning rate $\delta^1\eta$, stage $n-2$ is trained with a learning rate $\delta^2\eta$ and so forth. In the ConSinGAn experiments, setting $\delta = 0.1$ provided a good balance between image fidelity and diversity.

- **Minimum Size:** The minimum size represents the resolution at which the model starts learning. For instance, if the minimum size is set to 120, the model starts learning at a resolution of 120 by 120 pixels.

- **Number of Concurrently Trained Stages:** As this parameter increases, the diversity of the generated images tends to decrease.

The FID scores obtained were as follows:

- FID score between real and naive slices: 10.018
- FID score between real and slices generated by models with 6 stages (first configuration): 99.386
- FID score between real and slices generated by models with 3 stages (second configuration): 105.757

While it was anticipated that the models with 6 stages would yield a better FID score, the observed results were not promising. The FID scores indicated a significant difference between the naive tumor-injected slices and the model-generated slices. Further investigation revealed that the images produced by models exhibited lower quality, appearing more blurry and with reduced resolution compared to the naive tumor-injected slices. This disparity is visually evident in Figure 8. Consequently, additional experiments were performed to explore whether parameter adjustments could improve model-generated images' resolution. The subsequent tests revealed that increasing the minimum size parameter significantly enhanced the quality of the generated images produced by the model. In Figure 9, the results of these experiments are presented. Each subplot's title corresponds to the specific parameter adjustment made, while all other parameters not explicitly mentioned were maintained according to the specifications outlined in the first configuration. To further confirm the positive impact of increasing the minimum size parameter on image resolution, three slices were trained under two configurations, with the FID scores indicating promising results for the first time. The configurations were as follows:

Configuration 3:

- Stages: 6
- Learning Rate Scaling: 0.1
- Minimum Size: 240
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$42 minutes per model

Configuration 4:

- Stages: 3
- Learning Rate Scaling: 0.1
- Minimum Size: 240
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$16 minutes per model

The obtained FID scores were as follows:

- FID score between real and naive slices: 88.96
- FID score between real and slices generated by models with 6 stages (third configuration): 88.07

- FID score between real and slices generated by models with 3 stages (fourth configuration): 94.04

While these results are promising, they come at the expense of increased training time. Therefore, further hyperparameter tuning is deemed essential to strike an optimal balance. Additionally, it is advisable to train more models under the specified configurations to confirm the observed trend in the FID scores.

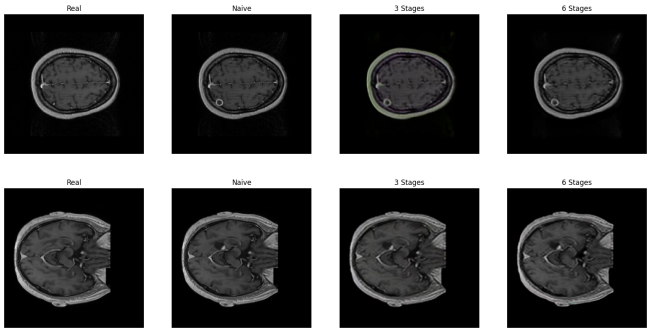

Fig. 8. Initial Results for Tumor Harmonization Task

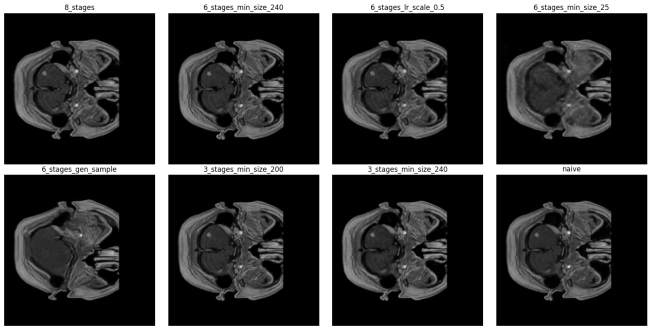

Fig. 9. Experiments Results for Tumor Harmonization

3.2 Editing Task Results

I trained four slices with tumors using two different configurations, resulting in a total of 8 models. The configurations were as follows:

Configuration 1:

- Stages: 6
- Learning Rate Scaling: 0.1
- Minimum Size: 25 (default suggested for this task)
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$9 minutes per model

Configuration 2:

- Stages: 3
- Learning Rate Scaling: 0.1
- Minimum Size: 25
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$6 minutes per model

In contrast to the harmonization task, the process of removing tumors did not consistently produce visually pleasing outcomes across all slices (refer to Figure 10). This fluctuation in performance was evident in the FID scores, which exhibited higher values compared to the FID scores obtained in the harmonization task. The following FID scores were obtained:

- FID score between real and naive slices: 4.463
- FID score between real and slices generated by models with 6 stages (first configuration): 187.324
- FID score between real and slices generated by models with 3 stages (second configuration): 184.125

It is noteworthy that the model with 3 stages produced slightly better results than the model with 6 stages in this task. This observation raises questions and prompts further investigation into the factors influencing the performance of the editing task.

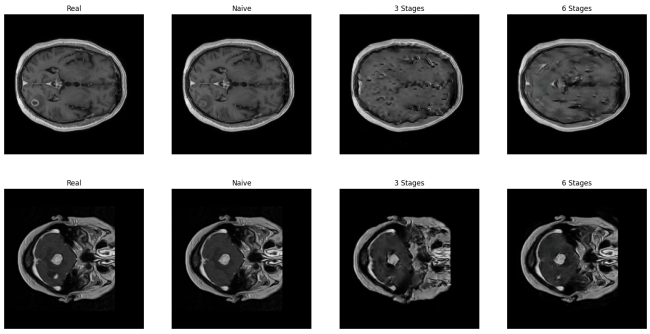

Fig. 10. Initial Results for Tumor Removal Task

To assess whether the minimum size parameter has a positive impact on image resolution and task performance in the editing context, I conducted additional experiments on a single slice under two distinct configurations. The configurations were as follows:

Configuration 3:

- Stages: 6
- Learning Rate Scaling: 0.1
- Minimum Size: 240
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$34 minutes per model

Configuration 4:

- Stages: 3
- Learning Rate Scaling: 0.1
- Minimum Size: 240
- Number of Concurrently Trained Stages: 3
- Training Time: $\sim$14 minutes per model

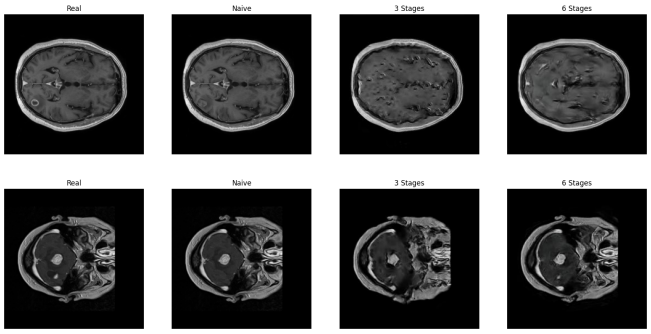

Fig. 11. Additional Results for Tumor Removal Task

Interestingly, similar to the harmonization task, increasing the minimum size parameter led to better results in terms of image resolution (see Figure 11). Although this improvement came at the expense of increased training time, the promising outcomes suggest that further investigation and fine-tuning of hyperparameters are essential to strike an optimal balance.

4 FUTURE WORK

Despite ConSinGAN's demonstrated capability in adding and removing tumors, further investigation is needed to explore methods for enhancing the model's ability to generate images with improved resolution while considering computational efficiency. Additionally, there is a planned extension of the model to 3D. The exploration of ConSinGAN's adaptability to 3D scans aims to evaluate its effectiveness in seamlessly adding or removing metastases within entire volumetric scans. This extension presents an exciting avenue for improving medical image synthesis, enhancing the model's applicability in clinical scenarios, and providing more accurate representations of anatomical structures.

5 References

[1] Ian J Goodfellow, Mehdi Mirza, Bing Xu, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[2] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[3] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli, "Singan: Learning a generative model from a single natural image," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[4] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter, "Improved techniques for training single-image gans," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1300–1309.

[5] B. Ocaña-Tienda, J. Pérez-Beteta, and J.D. Villanueva-García et al., "A comprehensive dataset of annotated brain metastasis mr images with clinical and radiomic data," *Sci Data*, vol. 10, pp. 208, 2023.

[6] Maximilian Seitzer, "pytorch-fid: FID Score for PyTorch," https://github.com/mseitzer/pytorch-fid, August 2020, Version 0.3.0.