Super-resolution of MRI Imaging Data with Generative Adversarial Networks

Samuel Moerman
Dept. of Computer Science
New York University
New York, USA
shm9853@nyu.edu

Luis Huiza
Dept. of Computer Science
New York University
New York, USA
lgh7953@nyu.edu

Jackson Oleson
Dept. of Computer Science
New York University
New York, USA
jo1449@nyu.edu

Abstract—The focus of this project is the application of super-resolution imaging in conjunction with generative adversarial networks, to be able to create higher-resolution MRIs from lower-resolution samples.

Index Terms—super-resolution, generative adversarial networks, MRI imaging

I. Introduction

Magnetic Resonance Imaging, or MRI for short, is a non-invasive imaging technology that is employed to make detailed pictures of the physiology and anatomy of patients. The technology uses powerful magnets that create a magnetic field that forces protons inside of the body to align themselves with the field. Then a radiofrequency current is made to pulse through the body of the patient, stimulating the protons that are aligning with the field, making them spin. When the radiofrequency is then turned off, the MRI sensors can detect the energy released from the spinning protons as they realign with the magnetic field and use their behavior to map out the inside of the patient's body. However, being able to get a proper image from this technique can be an exhausting process. [2], [9], [10] To produce higher-resolution images from the MRI scans, patients have to stay inside the MRI scanner for extended periods of time. Not only that, but they have to stay still during the time it takes to capture images, or they can end up blurring the final result. Because of this, there has been research on the use of digital imaging to obtain accurate, higher-resolution images from MRI scans without subjecting patients to extraneous circumstances needed to normally get imaging of that quality. There lies the focus of our project, we aim to combine the use of super-resolution technology and combine it with deep learning techniques utilizing generative adversarial networks to produce higher quality MRIs that are as accurate as possible while also making them more efficient and cost-effective.

II. Related Work

The goal of super-resolution (SR) methods is to recover a high-resolution (HR) image from one or more low-resolution (LR) images. In research by Farsiu et al. [4], SR is performed by taking a set of low resolution images of the same scene with small misalignments. Each LR image poses a set of linear constraints on the unknown HR image, and with enough images available, the set of equations becomes determined and can be solved to recover the HR image. More recently, deep learning methods have been utilized to reconstruct the HR image from a single LR image. Single image SR is a very broad topic, given that a multitude of approaches can be used. Because of this, lots of solutions arise involving strong priors or a learned mapping between LR and HR image pairs. Work by Dong et al. [3] uses patch extraction/representation, nonlinear mapping, and reconstruction to recover the HR image, while work by Ledig et al. [6] uses Generative Adversarial Networks along with a weighted perceptual loss function to improve perceptual similarity over similarity at the pixel level.

III. Model

Our goal is to perform super-resolution of MRI images using state of the art SR methods, more specifically Generative Adversarial Networks, to recover HR MRI images from downsampled LR images.

A. Data Preparation

We used a dataset first presented by Buda et al. [1] containing brain MRI images of patients with lower-grade gliomas. It contains roughly 4000 (256×256) 3-channel images collected from 110 patients. We produced the LR images by convolving with a Gaussian blur with a kernel size of 2 and subsequently downsampling the original HR images using a bicubic interpolation method.

B. GAN's

We outline a brief background on Generative Adversarial Networks (GAN's). A more comprehensive review can be found in the work of Goodfellow et al. [5]. Generative Adversarial Networks (GAN) are deep learning frameworks that are used for a wide range of applications, a prominent one of which is image generation. GANs have a unique architecture in the fact that they are composed of two distinct neural networks: a Generator $\mathcal{G}$, and a Discriminator $\mathcal{D}$, which assist in the training process of each other. The Generator is trained by taking a random sample of noise $p_z(z)$ and mapping it to a space represented by $G(z, \theta_g)$, where G is a differentiable function represented by a deep convolutional network with parameters θ_g. The Discriminator is trained concurrently with the generator, which takes as input the generated "image" from the generator, and maps it to a probability $\in [0, 1]$ that represents the probability that the input x came from the ground truth data rather than p_g (generated). The differentiable function $D(x, \theta_d)$ is represented by a deep convolutional network with parameters θ_d. The Generator and the Discriminator together make up the GAN and are trained against each other in a game-theoretic fashion, where the Generator attempts to fool the discriminator by generating an image that resembles the original data, and the discriminator attempts to discern whether the generated image is from the original data or the generator's distribution. The training ends when $D(x) = 1/2$ i.e. when the discriminator cannot detect whether the generated image is real or fake, and does no better than simply random guessing.

C. Generator Architecture

We define a generator $\mathcal{G}$ based on the work by Ledig et al. [6], which improves upon the MLP architecture presented by Goodfellow et al. [5]. The generator is a deep convolutional network with 16 identical residual blocks that perform the same operation. Contained in the residual block is the following repeated twice but with a Paremetric Relu activation function after the first sequence: a 2D convolutional layer with 64 filters and a 3×3 kernel followed by a batch-normalization layer. Finally, we upsample using two blocks of 2D convolutional layers with 256 filters and a PixelShuffle with upsampling factor $r = 2$.

D. Discriminator Architecture

We define a discriminator $\mathcal{D}$ by following the architectural guidelines of Ledig et al. [6]. It contains 8 2D convolutional layers of an increasing number of features by a factor of 2 from 64 to 512 features and each with a 3×3 kernel size. These are followed by two dense (fully connected, linear) layers and finally a sigmoid activation function to report a probability of the image being classified as real or fake.

E. Perceptual Loss

The mean square error (MSE) Loss is very simple and effective at the pixel level, and often produces good visual results which is why it is very popular among SR methods. However, the only drawback is that it operates by minimizing the error between pixel signals and does not consider the visual characteristics of the image as a whole. This can lead to some smoothness and blurring. To improve upon this, we incorporate a modified loss function that focuses on perceptual characteristics, again inspired by the work by Ledig et al. [6]. By incorporating a loss function that considers these perceptual characteristics, we intend to not just minimize some pixel-based loss function but to steer the generator towards images that visually resemble the original as perceived by the human eye. To achieve this, we leverage a post-activation layer from the VGG19 pre-trained model. [8] The goal is to extract feature maps from the deepest layer in the VGG19 network in hopes of targeting high-level image features. We define the perceptual loss:

$$l_P = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{i=1}^{H} \Big(\phi(HR) - \phi(G(LR)) \Big)^2$$

IV. Results

The generator architecture can both be used in the GAN and as a stand-alone super-resolution architecture. In order to assess the performance both the results from the generator (i.e. the SRResNet architecture) and the complete GAN (i.e. the SRGAN architecture) were examined.

A. SRResNet

This section describes the results of training the generator architecture to perform super-resolution. Both an x2 and an x4 upsampling were attempted. An Adam optimizer with a learning rate of 0.001 has been used, and the model was trained on a simple mean squared error loss metric. Additionally, the model was trained using batches of 16 images at a time and was trained with 16 residual blocks.

1) Upsampling x2: In general, the training of the SRResNet model to perform x2 upsampling was successful, given the losses on both the training dataset and the validation, as can be seen from Fig. 1. The model may decrease in performance due to over fitting from epoch 6 onwards.

Fig. 1: The training and validation losses for the SRResNet model for the x2 upscaling

However, the fact that the model was trained using the mean square error as a loss metric results in the super-resolution to not be completely satisfactory. The model is not able to capture all the fine details contained in the original image. For example, the folds of the brain are not clearly delineated. This does not come as a complete surprise, given the simplicity of the mean squared error function and previous findings by Ledig et al. [6]. However, the general outline of the morphological structures is well preserved. Some of the results can be seen in Fig. 2

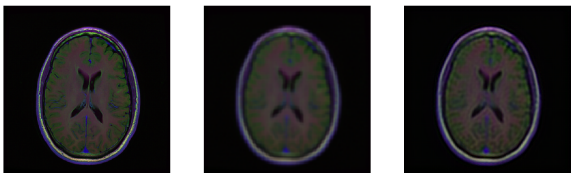

Fig. 2: The results of the x2 upsampling of the SRResNet model from epoch 6: The original high resolution image (left), the low resultion input image (middle), and the final super resolution image (right)

2) Upsampling x4: The super-resolution model using the x4 upscaling was not as successful. The training process was successful, with both the the training and validation loss decreasing over the epochs, as can be seen in Fig. 3. In contrast to the the x2 upscaling model, there does not seem to be overfitting, which may imply that the model could be marginally further improved by training for a longer time.

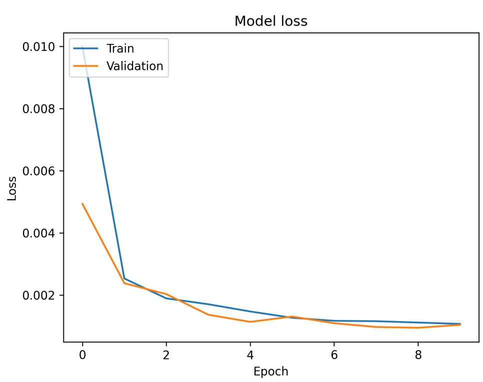

Fig. 3: The training and validation losses for the SRResNet model for the x4 upscaling

Here, the outputs displayed in Fig. 4 are much less satisfactory as with the x2 upsampling. It is no longer valid to state that the morphological features are well maintained in the output.

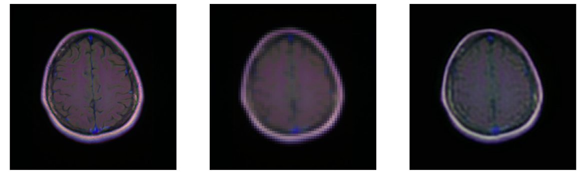

Fig. 4: The results of the x4 upsampling of the SRResNet model from epoch 6: The original high resolution image (left), the low resultion input image (middle), and the final super resolution image (right)

B. SRGAN

The SRGAN model was trained to perform x2 upsampling. Many hypterparameters were attempted to train this model. The final parameters that were used were Adam optimizers with a learning rate of 0.0001 for both the generator and the discriminator. Similar to the training of the SRResNet mode, 16 residual blocks were used. Here, no batching was used in an attempt to increase the performance and generalizability. The training was not very satisfactory and it proved not feasible to train model that is able to perform the super-resolution. In general, it is difficult to interpret the loss functions that are generated during the training of a GAN (vide infra) and the performance should rather be assessed by means of visual inspection of the outputs. However, we can see that both the generator loss and discriminator loss generally decrease throughout the training process, as illustrated in Fig. 5.

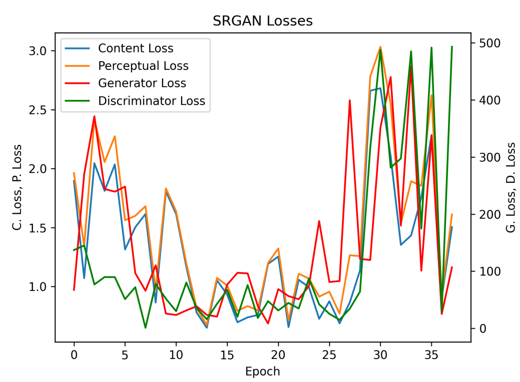

Fig. 5: The validation losses for the SRResNet model for the x4 upscaling

However, the model does not converge during the training process, as can be observed from Fig. 6. The model is clearly able to recognize some of the morphological features presented in the input images, but it is not able to train the generator to output an image that is similar to the original input image. It is possible to see how the GAN progresses from low-level features to more complex and specific features, where it learns to output some images resembling the original input. However, as the training

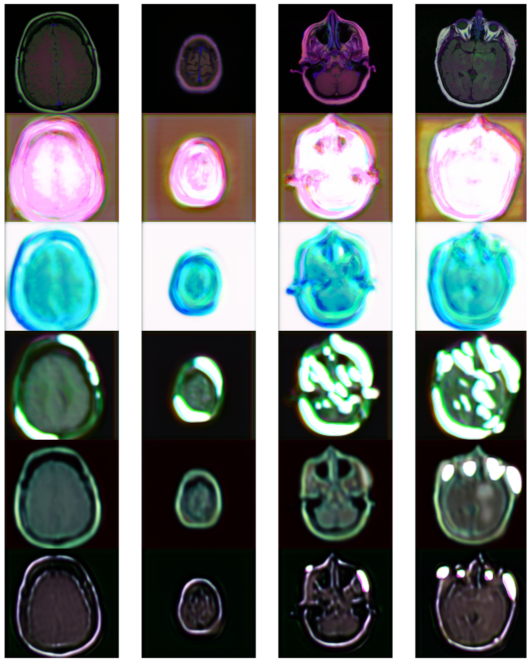

Fig. 6: Input high resolution images (top) and outputs throughout the training process of the SRGAN model (below)

continues, the model becomes unstable, which can be seen both in the visual outputs and the loss functions. In general, the training process is very variable from run to run and the instability is not easily reproducible. It could be possible that the MRI images are simply too different from the input data of the VGG19 model, which are normal pictures. In this case, the perceptive loss function would cause the entire model to become unstable during training.

V. Limitations

A. Difficulty of Training GANs

While GANs have been seen to generate very realistic results, they are notoriously hard to train. As stated in a follow-up paper by Salimans et al. [7], GANs are designed to find a Nash equilibrium of a non-convex game with continuous high-dimensional parameters. Meanwhile gradient descent - the method by which GANs are typically trained - is designed to minimize the provided loss function, not to find a Nash equilibrium of a game. This causes issues with convergence, as when the loss of one network goes down, the other will go up correspondingly. This unfortunate consequence of GAN architecture causes them to be very unstable in training and extremely sensitive to hyper parameters and initial conditions.

B. Initial Architecture

Some difficulties were encountered during the initial training process. The first GAN architecture we attempted to train was a much simpler one than outlined in this paper, with both the generator and discriminator containing four 2D convolutional layers that mirror each other in architecture but perform their respective mappings. Outputs from this were promising for the first few epochs, but quickly descended into instability.

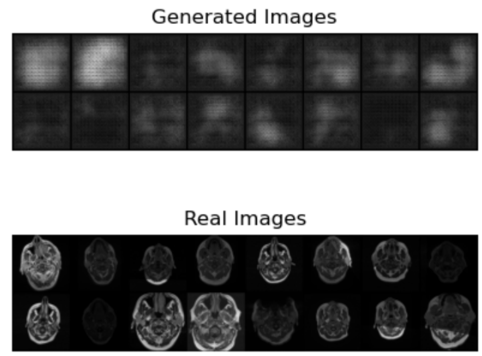

Fig. 7: Outputs from the original GAN

We proceeded to follow the recommendations of Salimans et al. [7] and add batch-normalization layers. However, with this architecture, we experienced some mode collapse difficulties. Mode collapse refers to the behavior where the generator finds some features that can fool the discriminator and repeatedly generates small variations of those without exploring the entire distribution of the data or diversifying the generated samples. This causes each of the generated images to be more or less identical.

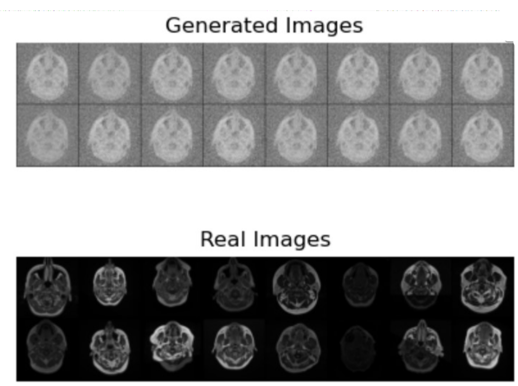

Fig. 8: Mode collapse illustrated by our primal GAN

These results showed that it was necessary to explore more complex and deeper architectures, such as the one we have presented in this paper.

VI. Conclusion

The architecture of the generator (SRResNet) seems to be fundamentally solid for performing super-resolution of MRI brain images, which is supported by the results presented in section IV-A. This gives indication that the architecture can be used as a stepping stone for further research. At this point in time, the main limiting factor seems to be the loss function on which the generator is trained. The mean squared error gives reliable results, but suffers a more blurred output, which has previously been experienced in other research too. The VGG19-based GAN approach suffers from instability, and it proved more difficult to train the model than initially expected. Some conjectured causes for this instability include the inherent difficulties or training a GAN and the incompatibility of the VGG19 model with the MRI input images.

Immediate future work could include more robust hyper-parameter tuning to attempt to improve the difficulties with instability, which in this work has been more trial-and-error. Besides this, an attempt could be made to use a more robust perceptive loss function. One possible strategy could be to fine-tune the VGG19 model on the original input set, which could improve the feature extraction. With an even more ambitious scope, some more advanced deep-learning techniques can be attempted to perform super-resolution of the MRI images. One possible approach could be based on a composed 3D input that leverages the fact that adjacent slices from an MRI session are heavily correlated. Finally, this work could potentially differentiate itself from the literature by combining the idea of performing super-resolution with segmentation of low grade gliomas, which is the underlying subject of the dataset.

VII. Individual Contributions

A. Jackson Oleson

- Model architecture and training
- Project organization
- Research and development

B. Luis Huiza

- Model testing
- Performance testing
- Project organization

C. Moerman Samuel

- Model architecture and training
- Hyper-parameter Tuning on HPC
- Notebooks

References

[1] Mateusz Buda, Ashirbani Saha, and Maciej A Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Computers in biology and medicine, 109:218–225, 2019.

[2] Ivana Despotović, Bart Goossens, and Wilfried Philips. Mri segmentation of the human brain: challenges, methods, and applications. Computational and mathematical methods in medicine, 2015, 2015.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2015.

[4] Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. IEEE transactions on image processing, 13(10):1327–1344, 2004.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, 63(11):139–144, 2020.

[6] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4681–4690, 2017.

[7] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[9] Paul Suetens. Fundamentals of medical imaging. Cambridge university press, 2017.

[10] Anjali Wadhwa, Anuj Bhardwaj, and Vivek Singh Verma. A review on brain tumor segmentation of mri images. Magnetic resonance imaging, 61:247–259, 2019.