Heart MRI Superresolution

Domonkos M. Fris, Panni Homolya,
Zsófia E. Turcsik

Abstract: Magnetic Resonance Imaging (MRI) is a widely used tool in medicine for the diagnosis of patients. High-resolution diagnostic images take a long time to acquire, so nowadays Machine Learning tools are used to convert low-resolution images into high-resolution ones. In our project, we use a database of cardiac MRI (Bernard, et al., 2018) to find a solution to the problem described above. The dataset contains 4D images that we have converted into 3D and 2D images. Furthermore, the original data also contains segmented images of hearts, which were used to cut out the outline of the heart from the 4D images and convert them into 2D images. For this project, we used SRGAN (Ledig, et al., 2017) and WGAN (Chen, et al., 2018) models. For both models, we consider the data detailed above as high-resolution images and generate low-resolution images from these images. The WGAN (Chen, et al., 2018) model was initially trained on 2D data, however, we were unable to upgrade the model to return a non-noisy image. We then tried with 3D data, but this did not yield any results. The SRGAN (Ledig, et al., 2017) model was then trained on 2D data with different parameter settings. Then these models were further trained with our segmented images. We took SSIM (Brunet & Vrscay, 2011) as our main evaluation metric. Based on these results, the model using the Adam optimizer achieved the highest SSIM value of 0.41102 with a training rate of 0.001. Finally, this one model was retrained again and evaluated on the test set, we obtained a SSIM value of 0.38629.

1. INTRODUCTION AND MOTIVATION

In the medical field Magnetic Resonance Imaging, i.e. MRI, is an important tool, widely used for forming pictures of various organs in the body. In our case, we take images of the heart, which changes quite rapidly in its cycle of movement. Making high resolution MRI images takes longer time and the movement of the heart or the patient could result in the pictures being blurry, which in turn makes the diagnosis harder. Low resolution images require less time to finish, but it also makes it harder to identify any deviations from the normal. This is the motivating factor of this project, which is to use machine learning techniques to upscale low resolution MRI images to high resolution ones. This process is referred to as image super-resolution. For image processing it is best to use neural networks. We will describe the method of generating the data and then present the basic structure of the so called SRGAN network we used for the conversions of the images. Finally, we will evaluate the model and describe some metrics we used for this process.

2. DATA GENERATION

2.1 Description of Raw Data

Our raw input data consisted of 4 dimensional high resolution MRI images from 100 different patients, the fourth dimension being the time. the dimensions varied from patient to patient, roughly being 10 pieces of 216×256 images for less then 30 discrete time points. It also included ground truth files for one or two 2 dimensional pictures, which had the exact positions of the heart in the images. These images were in the Nifti format (Larobina & Murino, 2014), a format especially developed for neuroimaging.

2.2 Train, Test and Validation Sets

In order to feed our images to the model, we had to convert the original 4D images to 2 dimensional ones. We also converted the 2D pictures to the same squared 256×256 format with the cv2 package's resize method, using bicubic interpolation.

We saved these high resolution images in jpg format to three folders of train test and validation. The set of patients appearing in these folders were disjoint, i.e. if the train data included images from patient 001, then the validation and test data didn't and if the validation data included patient 001, then the train and test data didn't.

2.3 Ground Truth Images

As mentioned before, our raw data included the precise positions of the heart for certain pictures. Leaving a 10 pixel leeway in every direction, we got the union of the segmentations we got for a certain patient and cut the heart out from every picture we saved. We also used these images for training and validating the model.

2.4 Conversion of the 4D images

We also needed low resolution versions of the pictures for the model to train on, so we wrote a function that read in the 2 dimensional pictures as numpy arrays (Van Der Walt, Colbert, & Varoquaux, 2011), partitioned the images to 3×3 sections and assigned each pixel the value of the lower left pixel of the part they were in, returning a pictures three times "smaller", while actually preserving the original size of the image.

3. SRGAN NETWORK

For converting the low resolution images to high resolution, we used a super-resolution generative adversarial network, in short, SRGAN model (Ledig, et al., 2017). This is a GAN

(Goodfellow, et al., 2020) based network, which simultaneously trains two models via an adversarial process. The first one is a generative model G, that tries to capture the distribution of the training data, and the second is a discriminative model D, that estimates the probability, that a sample were in the original training data, rather than it was generated by G. The generator model's goal is for D to not be able to distinguish between the original and the generated data. This way the two models force each other to improve. Another advantage of these GAN networks, that they are able to capture more sharp and even degenerate distributions.

The generator of the GAN network consists of residual dense blocks (Song, et al., 2020) optimized on MSE.

Next to the MSE based content loss, the SRGAN more heavily relies on calculating loss on the feature maps of the VGG19 model (Simonyan & Zisserman, 2014). This way, the SRGAN model is less exposed to be influenced by the changes in the pixel space.

4. WGAN NETWORK

It is important to know that similar to the SRGAN model the WGAN (Chen, et al., 2018) model's backbone is also a residual dense block architecture. The main difference between the two models is the training process itself. While the SRGAN model uses a somewhat sophisticated training, with extracted feature based loss (Dong, Loy, He, & Tang, 2016) (Shi, és mtsai., 2016), the WGAN adds another level of complexity with different phases in the training process. The training consists of five parts. First, they train the generator alone for a while. After that, both models are trained simultaneously with a specialized loss. In the third phase the discriminator is trained alone for some time, which is followed by imbalanced training, meaning several discriminator steps followed by a single generator update. The final phase is simultaneous training and after every few hundred steps the discriminator is trained alone for a couple hundred steps.

5. VALIDATION METRICS

5.1 Normalized Root MSE

We used the normalized_root_mse function of the skimage.metrics package to evaluate our prediction. We first calculated the normalized root mean-squared error for every picture and then got the average. There are several ways you can normalize the RMSE, but we used the 'euclidean' method, which means the following formula for single image NRMSE calculation:

$$NRMSE = \frac{RMSE \cdot 256}{\sum_{i=1}^{256}\sum_{j=1}^{256}|a_{ij}|^2},$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{256}\sum_{j=1}^{256}(\hat{a}_{ij} - a_{ij})^2}{256 \cdot 256}},$$

Where a_{ij} and $\hat{a}_{ij}$ denotes the original and predicted values of the pixels respectively. Our target is to minimize this measure.

5.2 Peak Signal to Noise Ratio

Our next method for evaluating our models were the peak signal to noise ratio (PSNR). This metric measures the maximum possible value of the signal, i.e. the maximum value of our target array and the maximum value of the noise. Since the signal's value could move on a wide, dynamic range, this metric is usually represented in the logarithmic scale as follows:

$$PSNR = 20 \cdot \log_{10} \frac{\max\limits_{1\le i\le 256, 1\le j\le 256}\{a_{ij}\}}{RMSE},$$

the notation being the same as before. Since we would like the RMSE to be as small as possible, the higher PSNR usually means, better results.

5.3 Structural Similarity

Our final metric for evaluation was the structural similarity index measure (SSIM). Unlike RMSE or PSNR, which try to estimate absolute errors, this measure is based on the idea, that pixels located near each other have strong inter-dependencies. As you can derive from its name, this metric tries to capture the structure of the data. It is calculated on the following way:

$$SSIM = \frac{(2\mu_a\mu_{\hat{a}} + c_1)(2\sigma_{a\hat{a}} + c_2)}{(\mu_a{}^2 + \mu_{\hat{a}}{}^2 + c_1)(\sigma_a{}^2 + \sigma_{\hat{a}}{}^2 + c_2)},$$

with the following notation:

μ_a is the mean of the pixels of the original picture,

$\mu_{\hat{a}}$ is the mean of the pixels of the predicted picture,

σ_a is the variance of the pixels of the original picture,

$\sigma_{\hat{a}}$ is the variance of the pixels of the predicted picture,

$\sigma_{a\hat{a}}$ is the covariance of the pixels of the original and the predicted picture.

The values of this metric is falling in the interval $[-1, 1]$, -1 meaning negative correlation, 0 meaning there is no correlation, and 1 meaning perfect correlation.

6. RESULTS

Our main results came from the SRGAN model. The table below (Table 1) contains the evaluation of three of our models. The difference between these models is that their learning rate differs from one another, it being $0.001, 0.0002$ and 0.00001 respectively. Each model was first trained for two epochs on our full sized data, and then trained them for another two epochs on our segmented data and evaluated on the validation set.

Metrics	Models		
	Model_1	Model_2	Model_3
Discriminator loss	0.00120	0.00408	0.00024
Generator Loss	0.42242	0.43362	0.46546
SSIM average	0.41102	0.36827	0.28807
PSNR average	57.04140	56.84810	55.68944
NRMSE average	0.57115	0.58439	0.66737

Table 1: Evaluation of our models

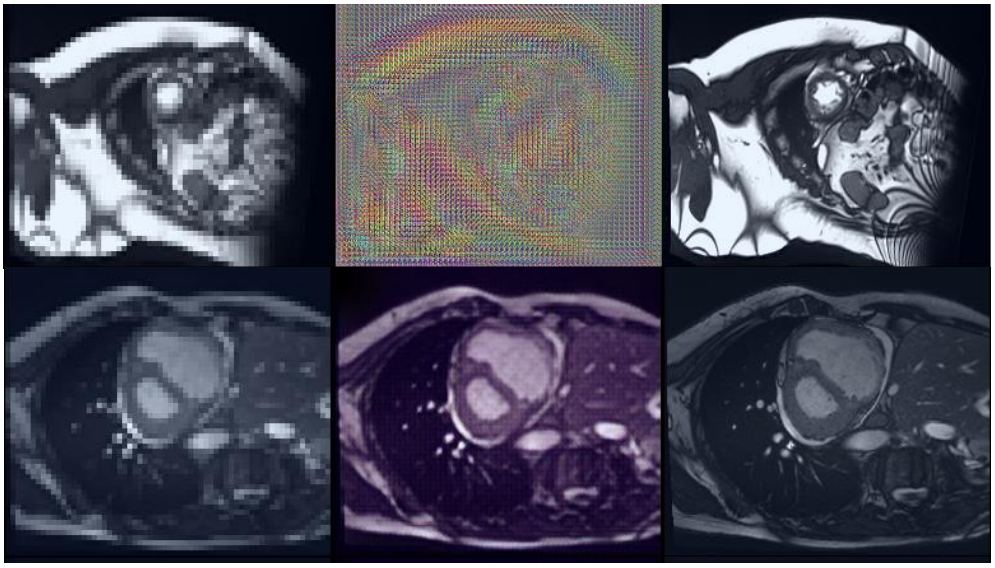

Figure 1: From left to right you can see the low resolution, the prediction and the high resolution image

In Figure 1, first you can see the performance of the early training period, and then in the second row, you can observe the obvious development for the end of the second epoch.

The idea was that at first we ought to use as many MRI data as possible, hence the full sized images, but then we wanted to try and focus on the heart itself.

After this we chose the second model based on its performance on the validation data and trained it for another 20 epochs on the segmented data. The results of the evaluation of the test set can be seen in Table 2 below and in Figure 2.

Metrics	Final model
Discriminator loss	0.00079
Generator Loss	0.39883
SSIM average	0.38629
PSNR average	57.54778
NRMSE average	0.57218

Table 2: Evaluation of our final model

Unfortunately, these further trainings couldn't improve the performance of the model, and on average it achieved basically the same results.

7. CONCLUSIONS FUTURE PLANS

Although we did not manage to find any research made into heart MRI super-resolution specifically, there are plenty of other relevant work. Therefore, our goal became utilizing these existing solutions in solving our problem. At first we wanted to use the fully pre-trained model provided by (Chen, et al., 2018). However, data incompatibility problem arose. The WGAN model was trained on significantly different data than ours. First, the model is supposed to be working with 3D data, meaning it requires a full MRI scan as input, in this case a $320 \times 320 \times 256$ image, compared to our data, which was around $216 \times 256 \times 10$. The difference in volume was too high and our futile attempt of multiplying the third dimension and resizing the other two was met with a meaningless noisy output.

This is why we resorted to the SRGAN model, which was not necessarily trained and developed on MRI data, but is nonetheless a useful tool for super-resolution.

For future development there is always room for fine tuning the hyperparameters. What seems to be our main obstacle tough, is that our models do not seem to be learning anything after the first couple of epochs. There are a few possible ways around this. First, due to the available data being scarce, we could try training our models first on other MRI scans and then fine tune the models to our data. Lastly, perhaps we exhausted the model's capability to learn MRI super resolution, so maybe increasing its depth or complexity could provide some improvement.

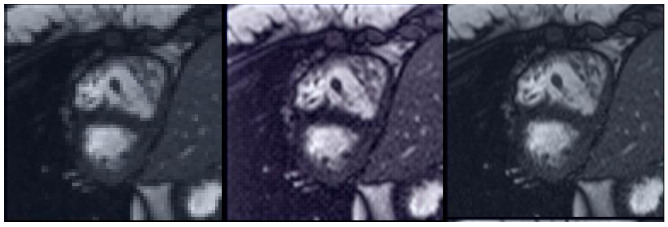

Figure 2: Final test result for segmented data

REFERENCES

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P. A., & Jodoin, P. M. (2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37(11), 2514-2525.

Brunet, D., & Vrscay, E. R. (2011). On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4), 1488-1499.

Chen, Y., Shi, F., Christodoulou, A. G., Xie, Y., Zhou, Z., & Li, D. (2018). Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 91-99.

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. .., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.

Larobina, M., & Murino, L. (2014). Medical image file formats. *Journal of digital imaging*, 27(2), 200-206.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A. .., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681-4690.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., . . . Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1874–1883.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv*, 1409.1556.

Song, D., Xu, C., Jia, X., Chen, Y., Xu, C., & Wang, Y. (2020). Efficient residual dense block search for image super-resolution. *In Proceedings of the AAAI Conference on Artificial Intelligence Vol. 34, No. 07*, 12007-12014.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2), 22-30.