Project Title: Application of deep-learning to enable quantitative brain hemodynamic mapping using resting-state fMRI

Project Members: Majed AlFaraj

Project Mentor: Jean Chen

Abstract:

Cerebrovascular reactivity (CVR) is the ability of blood vessels to dilate and constrict in response to vasoactive stimuli. It is a key indicator of cognitive health, particularly used to diagnose the risk of cerebrovascular diseases such as dementia and Alzheimer's. Conventionally, CVR is quantified as the ratio of the change in blood-oxygenation-level-dependent (BOLD) signal to the change in partial pressure of CO2 (PaCO2). However, PaCO2 measurement requires specialized equipment typically unavailable in small clinics and hospitals. The BOLD signal is a function of cerebral blood flow (CBF), cerebral blood volume (CBV), and oxygen metabolism (CMRO2), with CBF being the most readily measurable physiological factor. This raises the question: can we estimate CVR using solely CBF, without PaCO2 measurements? Since the exact relation between CBF and CVR is unknown and likely non-linear, this is an ill-posed inverse problem—we have no assurance that a deterministic, unique, or stable mapping exists from CBF to CVR. In addition, current CVR estimation methods assume linearity between BOLD and PaCO2, which neglects known non-linear edge cases such as vascular saturation. We will explore a deep learning approach to predict voxel-wise CVR maps given 3D resting-state CBF images. We will train a 3D U-Net model on paired CBF-CVR images to learn spatial correlations between the two modalities. The model will be evaluated on voxelwise mean absolute error and region-based R2, with attention to spatial consistency. We expect a substantial correlation between predicted and empirical CVR maps in cortical regions, where CBF is more indicative of cerebrovascular dynamics. If successful, this approach could enable smaller hospitals and clinics to utilize arterial spin labeling-derived CBF to estimate CVR, improving vascular assessment feasibility in routine clinical practice.

Goals and Objectives:

Image-to-image translation from a 3D CBF map to a 3D CVR map

U-Nets

- The first approach is to try and train a 3D U-Net model and see how well it performs.
- The caveats are that the relationship between CBF and CVR is complex, unknown, non-linear, and may be non-deterministic
- Loss: Hybrid between MSE + SSIM, with larger emphasis on MSE due to importance of numerical accuracy

cGAN

- If vanilla U-Nets deem to be insufficient for reliable CVR map estimations, an approach of using conditional GANs will be attempted.
- In the cGAN model, the image generator will be a U-Net model whilst the discriminator will likely utilize PatchGAN due to the spatial nature of blood flow.
- Loss: Hybrid between L1 + BCE (discriminator loss), with much larger emphasis on L1 due to importance of accuracy rather than "realism"

If approaches above don't produce needed results, ResNets and DenseNet-UNets may be looked into.

(Non deep learning methods such as support vector regression have been removed from consideration due to unknown relation between CBF and CVR, complicating choice of predetermined variables such as kernel transformation function)

A **vanilla U-Net** approach is generally preferred due to the focus on numerical accuracy rather than image realism, which is where **cGAN**s shine.

Methods:

I've worked with some of these models and frameworks on dummy datasets, gaining familiarity before applying it to the existing project dataset.

Pytorch—how I will create the model and train it on my data, then evaluating it on a test set

MONAI—the medical open network for artificial intelligence is an AI framework built on top of Pytorch for medical applications, specifically in medical imaging such as CT, X-Ray, and MRIs. Since CBF and CVR maps are fMRI-based, the framework will be useful. Can work with NIFTI and DICOM files.

FSL—program that deals with MRI files directly (NIFTI and DICOM) with pre-processing available. Already have been used to pre-process maps by skull-stripping, motion correction, and fitting to templates. This removes noise caused by inconsistencies caused by different orientations and skulls and focuses mainly on voxel-wise data

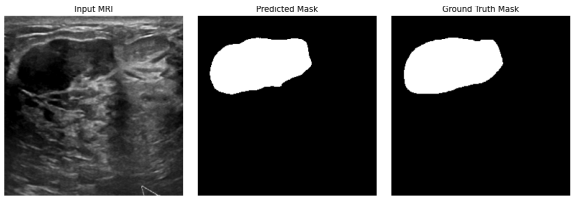
Obstacles:

Steep learning curve—took me a few weeks to gain familiarity with the medical terms, relationship between the physiological factors, and learning how to use these applications and methods (almost all completely new to me).

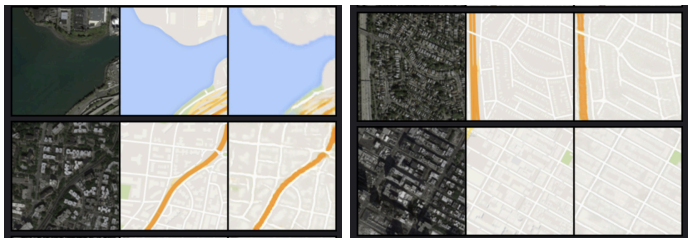
Second Update:

Since last week, I've implemented the aforementioned U-Net model and a Conditional General Adversarial Network using a U-Net for the generator and a PatchGAN binary classifier as my discriminator.

To gain familiarity with U-Nets, I created my own implementation of a U-Net module in Pytorch, tuning the encoding and decoding layers to best fit the application. I trained my U-Net application on a breast cancer segmentation dataset—yielding a dice score north of 0.92. An example is shown below



Since my main project deals with image-to-image translation, I decided to look further into pix2pix (conditional GAN) as they specialize in image translation in paired data. Similarly, I used almost the same U-Net implementation I had for the breast cancer segmentation but tweaked to fit the Satellite to Simple view translation dataset. I also implemented my own PatchGAN discriminator implementation—which helps make the output image less blurry (side effect of minimizing L1 Loss on U-Net output). Example outputs after training for 150 epochs:



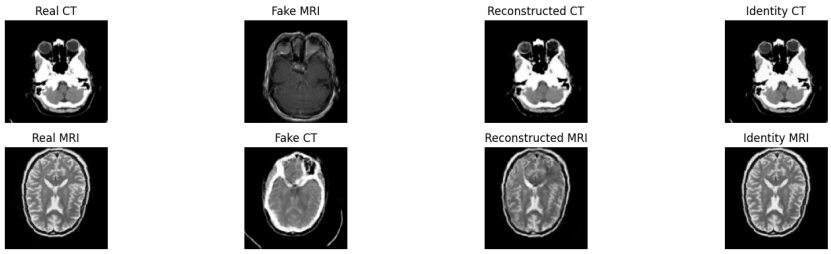
Currently, I'm working on a CycleGAN model which also specializes in image-to-image translation, but with unpaired data. I will next try to use CycleGAN to create a CT to and from MRI modality translation model.

Future Steps:

After finishing my CycleGAN application, I will work with T1w to FLAIR MRI modality translation until maintenance on the Baycrest server is done and we are able to anonymize the raw data. For privacy reasons, I can use an online notebook with GPUs such as Google Colab & adhere to HIPAA laws.

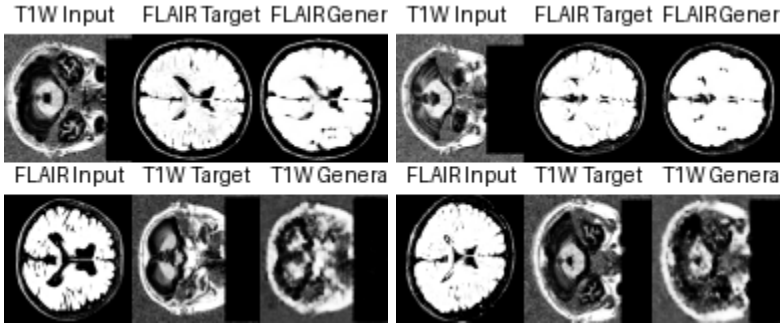
Third Update:

This week I completed my CycleGAN model on MRI and CT data with two U-Nets as my generators and two PatchGANs as my discriminators. I picked this approach as the MRI and CT data I acquired were unpaired. After training for 150 epochs, I got the following result:

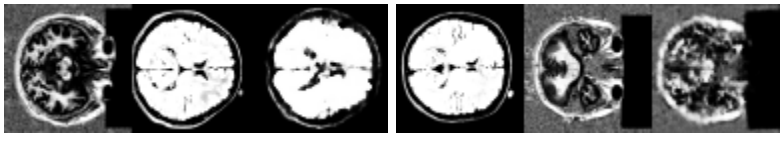


The data in this case was 2D slices, which were often taken from different planes. Given this, it was incredibly difficult for the model to generalize a translation when the slices were from fundamental different areas of the brain. However, training on 2D png images was a lot quicker than training 3D data.

When it comes to 3D data, I worked on translating between MRI T1w and FLAIR modalities. The most difficult part was acquiring and downloading the data. I used the Max Planck Institut Leipzig Mind-Brain-Body Dataset, which is tens if not hundreds gigabytes, many of which consisted of data that I didn't need. After extracting all the subjects MRI scans that had a pair of T1w and FLAIR data, I loaded that to my modified cGAN model which works with NIFTI filetypes and 3D images. The training was substantially longer, and I still have some tweaks to make to my model. However, some results of the latest version are shown below:



(training set — middle slice)

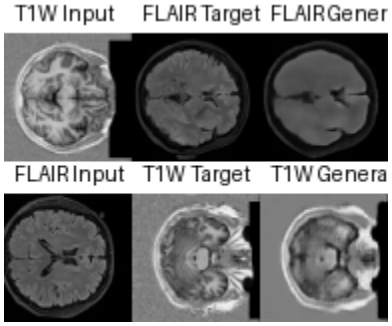


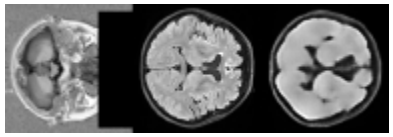
(val/test set — middle slice)

Future Steps: I will attempt to improve the generalizability of the T1w and FLAIR conversion model, and then afterwards work on the main objective of translation CBF and CVR maps.

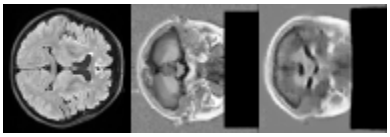
Fourth Update:

This week I continued working on my T1w FLAIR image-to-image translation model. Since last week, I managed to resolve the issue of shifted intensities—whether FLAIR images appeared to be too bright or T1w were too dark. By changing my image transformation pipeline to only scale the intensity of the image to [0, 1] and rewrite my generator to use a sigmoid activation function for the final layer, I was able to tone down the outputs to the following image.





(T1w to FLAIR validation) (training set)



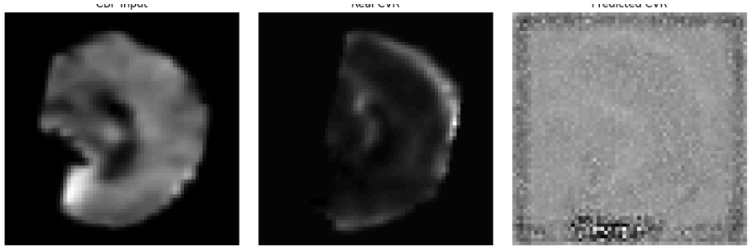
(FLAIR to T1w validation)

It is evident that the intensities are a lot more accurate. Additionally, the general anatomical structure seems to be a lot more consistent. However, probably due to high emphasis on minimizing L1 loss, a blurring/smoothing effect lowers the amount of details in the generated outputs. The next step for this part is to either lower the L1 lambda weight or to add complexity to the generator in order to get greater spatial detail and lower the blurring effect.

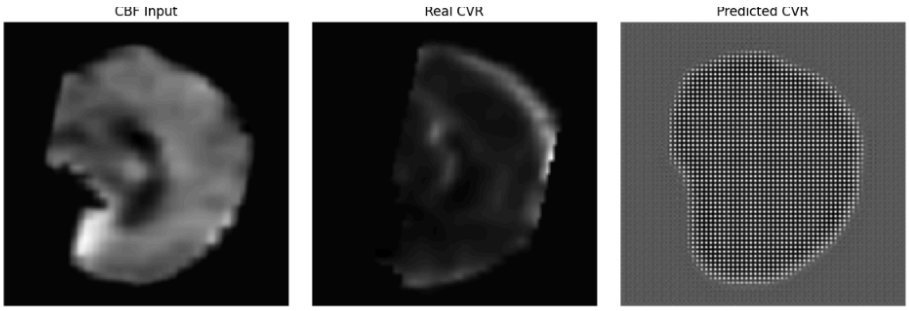
<u>MAIN PROJECT:</u>

This week I also made some progress in my main project of converting CBF maps to CVR maps. After making sure the data is anonymised and normalized to the MNI space, I uploaded the CBF & CVR paired data to train using an online notebook. The main caveat—however—is the existence of merely 14 pairs of NIFTI files, making it incredibly difficult to train an accurate model.

Due to the emphasis on numerical accuracy rather than image realism, I opted to begin using a simpler model architecture—such as U-Nets and ResNets. In my first attempt, I used a vanilla U-Net architecture with channels going from 64 to 512. After training for 100 epochs, the results are as follows:

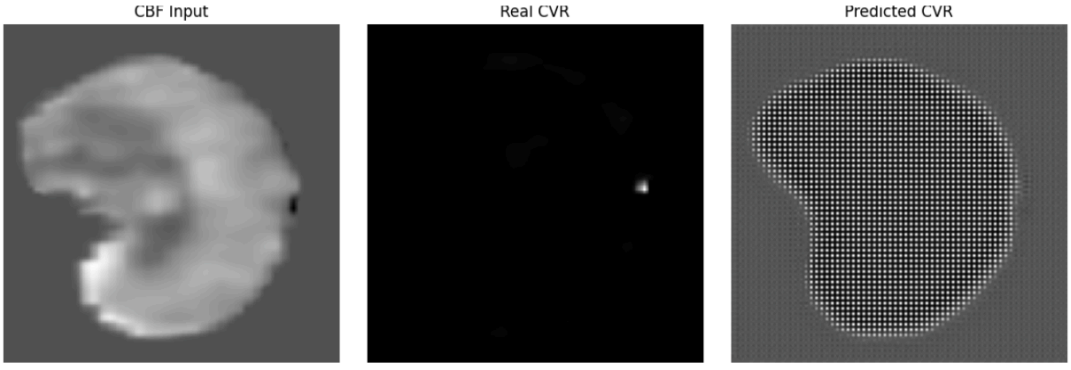


This indicates that the U-Net is failing to generalise. For this reason, I tried using a model Attention U-Net, which should focus on relevant regions in the brain scan. Using similar configurations to the original U-Net—however changed the image spatial size from 64 cubed to 128 cubed—I saw the following result after 300 epochs:

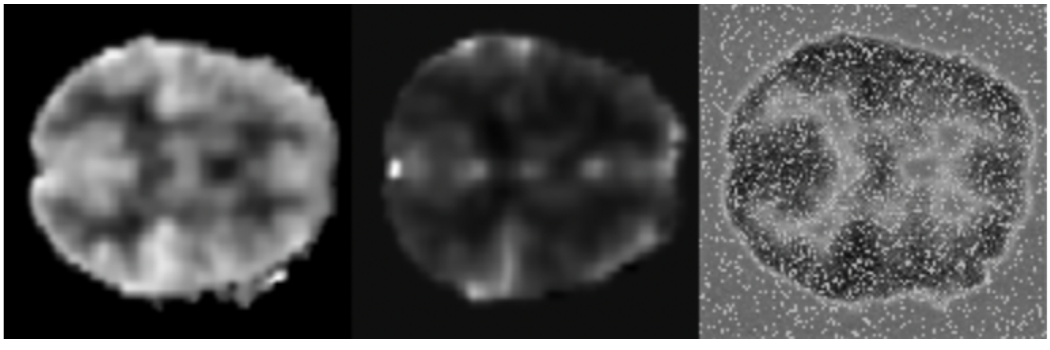


Similarly, this checkerboard pattern is a consequence of the upscaling section in the decoder of the U-Net which could result due to little information. For this reason, I wanted to use a deeper, more complex model.

In my third attempt, I used a Dynamic U-Net. These types of U-Net are a lot more computationally expensive, but with the amount of data that is available, it is more feasible to use. I used similar configurations to the last two units with the same amount of filters (channels). After training for 200 epochs, I got the following (similar to Attention U-Net):



Given these results, I decided to use a cGAN instead. In my cGAN, I kept a Dynamic U-Net, as it theoretically should yield more accurate results, with a PatchGAN as my discriminator. After training for 140 epochs, this is the following result:



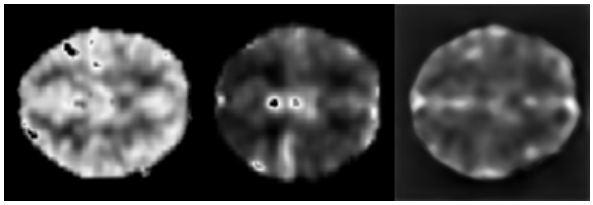
Despite the existence of obvious artifacts, the results are already a lot more promising. The general structure is a lot more apparent and there is some notion of anatomical structure compared to the usual checkerboard pattern.
The discriminator had a loss of 0.16 for epoch 140, which indicates that it is overpowering the generator.
The next steps:

1. Change the generator to something more powerful, such as a SegResNet
2. Weaken the discriminator by removing layers and/or add a holdoff for a few epochs
3. Add more layers to the generator, closing the "power gap" with the discriminator

Fifth Update:

I worked some more on the CBF CVR project. Given the early plateauing from yesterday, I retrained the model after slightly increasing the learning rate. Also, in this iteration, I began to save metrics as the training went on, such as the Pearson correlation coefficient, R2, PSNR, etc.

After training for 300 epochs, I saw the following result in the validation set:

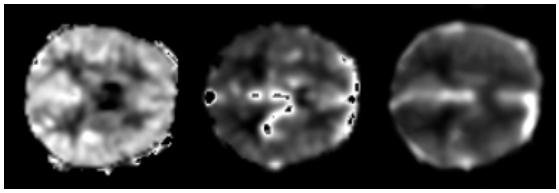


Despite a visually reassuring output, the metrics told a different story. At epoch 300, the r value was around 0.45, having peaked at 0.6 around epoch 40. Moreover, the R^2 value started at -10, and steadily increased to -3 by epoch 300. This result tells me that my model is performing worse than the worst possible least-squares solution. After further inspection, it became apparent that the main issue was due to scaling.

In my current iteration of the training, I opted to use a scaling transformation which set the minimum to 0, the maximum to 1, and adjusted the remaining values accordingly. This type of scaling was irreversible, and calculating the R^2 value in this transformed system explained the negative numbers.

As a result, I had to adjust the transformation pipeline to standardise the data using z-scores instead, as that transformation is easily reversible given the population mean and standard deviation.

After making the adjustment to the pipeline, incorporating normalisation and denormalisation throughout training, and replacing the Sigmoid activation with Identity due to standardising the data, I trained my model for 150 epochs and saw the following:


(val @ epoch 150)

The model started with an R^2 value of around 0.3, slightly increased to 0.58, then started going back down until -0.1 by epoch 150. On the contrary, the Pearson coefficient had shown a value of 0.75 by the end. This indicates that the “bright” and “dark” areas are conceptually correct (same pattern/relationship), but the numerical scale is off.

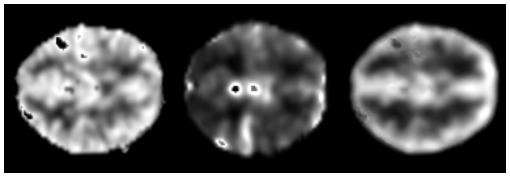
For this reason, I slightly changed my generator architecture to using an affine activation function. Similarly to the last iteration of the model, my initial gamma (slope) and beta (y-int) values were 1.0 and 0.0 respectively, replicating an identity activation. The difference—however—is that the gamma and beta values adjust during training to ensure that the numerical scale of the output matches what’s expected, as we know the general pattern is correct.

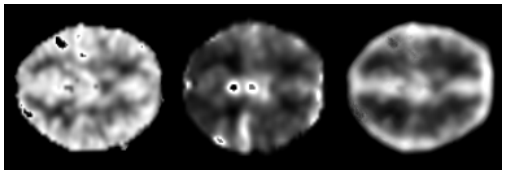
Before I retrained the data, I also started saving more checkpoints of the model parameters at different points in training, namely: epoch where R^2 (calibrated with gamma and beta) was max, epoch where val loss was min, epoch where PSNR was max, and the latest epoch trained.

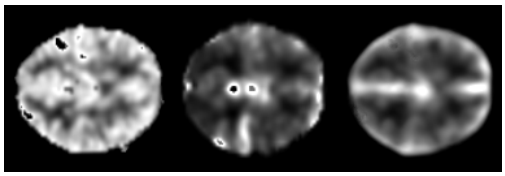
I trained my model for 300 epochs, and found the following statistics (all on validation set):

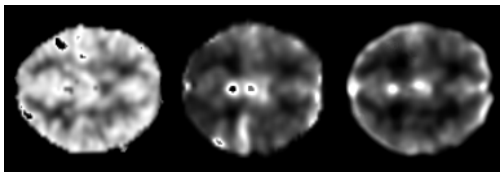
- Calibrated R^2 was max with a value of 0.6826 @ epoch 51
- Validation loss was min with a value of 5.445 @ epoch 20
- PSNR was max with a value of 24.0465 dB @ epoch 11

Despite these maximum values being hit at lower epochs than expected, it makes sense as 14 image-pairs is prone to overfitting.


val epoch 11 (max PSNR)


val epoch 20 (min val loss)


val epoch 51 (max R^2 calibrated)


val epoch 300 (last epoch)

The images above may not directly reflect the seen metrics for a few reasons.

1. I arbitrarily picked the first subject in the validation set to take a visualisation of; however, the metrics are logged as **an average of all the subjects** in the val set
2. I'm only looking at the middle slice of the brain, whereas the metrics are logged for the entire **3D volume**
3. The visualisation rescales the output data to [0, 1] from MNI space—the metrics are calculated in MNI before the rescaling so intensities can be misleading

The metrics logged are the objective measure, and the visualisations can be deceptive.

- The visualisations are **purely illustrative**