

Pelvic Organ Image Analysis
and
Geometry Reconstruction
using
Artificial Intelligence

College of Engineering, Computer Science, & Technology
California State University, Los Angeles

Senior Design Advisor: Dr. Negin Forouzes (CS, ECST)
Senior Design Client: Dr. Mathias Brieu (ME, ECST)

Team:

Ralph Belleca
Nicol Barrios
Ted Kim
Silvano Medina
Robin Mok
Alejandra Olvera
Demetrius Parker
Sabino Ramirez
Mary Semerdjian
Jason Tejada

Table of Contents

1. Introduction	3
1.1. Background	3
1.2. Problem	3
1.3. Contributions	4
1.4. Solutions	4
2. Related Work and Technologies	6
2.1. Existing Research	6
2.2. Reused Technology	6
2.3 Project Approach	7
3. System Architecture	8
4. Results and Conclusions	9
4.1. Fall Semester Results	9
4.2. Spring Semester Results	12
4.3 Conclusion	15
5. References	17

1. Introduction

1.1. Background:

The development of radiology and the utilization of magnetic resonance imaging (MRI) in healthcare are both relatively new compared to other specializations in medicine. Since its discovery in the late 1970s, MRI has gone through a series of innovations backed by powerful computers and field-changing technologies. Today, medical imaging still plays an essential role in allowing medical professionals to provide accurate information about a patient's anatomy, especially when dealing with tumors. It is a technique of producing images of the body's interior noninvasively. The output image, i.e., the MRI image, is processed to be more practical using image segmentation and 3-D model construction. Image segmentation is a method of partitioning an image into sections based on specific characteristics such as color, light intensity, texture, and depth. There are multiple ways to conduct image segmentation: manual, automatic, and semi-automatic.

1.2 Problem:

Although image segmentation has been through phases of improvement over the last decade, this procedure of segmenting images and creating 3-D models of specific organs can still be tedious and repetitive. Although very rare, segmentation can require hours for a single case for some situations. The current methodology used by the team implements a manual approach to segmentation which requires users to input a threshold range. This creates a mask over the MRI image and determines what turns into a 3-D object. The process can be inaccurate because the threshold range depends on whether the organ of interest is fully covered by the mask, leading to multiple issues such as underfitting and overfitting. Overfitting happens when the mask covers

more than what needs to be and underfitting is the opposite. When either of these two things happens, another process called repair and refinement needs to be done, which makes it more laborious and unreliable. All these processes make up the traditional methodology and need to be more efficient. As the project suggests, the traditional methodology can be improved using advanced technology like artificial intelligence (AI).

1.3 Contributions:

In our group of 10, each team member was assigned different roles; however, these tasks changed as the project progressed. Roles such as project lead and client liaison remained the same. Half of the team was split into the front end and the other into the back end. While one researched solutions, the other figured out how to implement the solutions. These roles were dynamic throughout or project, and many members worked together on different parts. The client for the project is Dr. Mathias Brieu from the Mechanical Engineering department of the College of Engineering, Computer Science and Technology at Cal State LA. He is currently working on biomedical projects with medical professionals from Europe, and the outcome of this Senior Design project will help one of his projects.

1.4 Solutions:

The overarching goal of this research project is to streamline the process of converting MRI images of pelvic organs into 3-D model objects. The project was separated into two learning stages: understanding basic 3-D model construction and the creation of an AI model. The goal for the Fall 2021 semester was to learn the basics of 3-D modeling using 3D Slicer, a 3-D visualization software, and determine different ways to automate it. The team believed that to

streamline the process of 3-D model construction, it was critical to understand the fundamental procedures and all their intricacies. The team also experimented on Nvidia Artificial Intelligence-Assisted Annotation (Nvidia AIAA). The goal for Spring 2022 was determined during the latter part of the previous semester, where the team encountered an issue with Nvidia AIAA. Through the experiments conducted during the Fall 2021 semester, one of the first obstacles of using Nvidia AIAA was the absence of a pre-trained AI model for the team's target organs, which were female pelvic organs such as the bladder, rectum, and vagina. For this reason, it was decided that the Spring 2022 semester will be focused on training our own AI model.

2. Related Work and Technologies

2.1 Existing Research

One essential aspect of this project is the use of graphical processing units (GPU). The GPU was initially designed to speed up the process of 3D graphics rendering but is now used in diverse applications such as digital automation, video rendering, medical image segmentation, and now artificial intelligence. There is a great number of research available regarding image segmentation. However, minimal research uses GPU-based medical image segmentation and specifically uses pelvic organs as their organ of interest. Most of the research is focused on the brain or lungs as their organ of interest. Although this was the case, there was research done by Southern Illinois University that helped understand the project. It was about analyzing brain tumor MRI using Nvidia AIAA, a GPU-based medical image segmentation. In a part of this research, Sajedul Talukder and his team extensively experimented on Nvidia AIAA. They demonstrated how better results were obtained using GPU-based processing compared to CPU-based. They showed that different algorithms designed explicitly for GPU generated up to 2-13 times better results.

2.2 Reused Technology

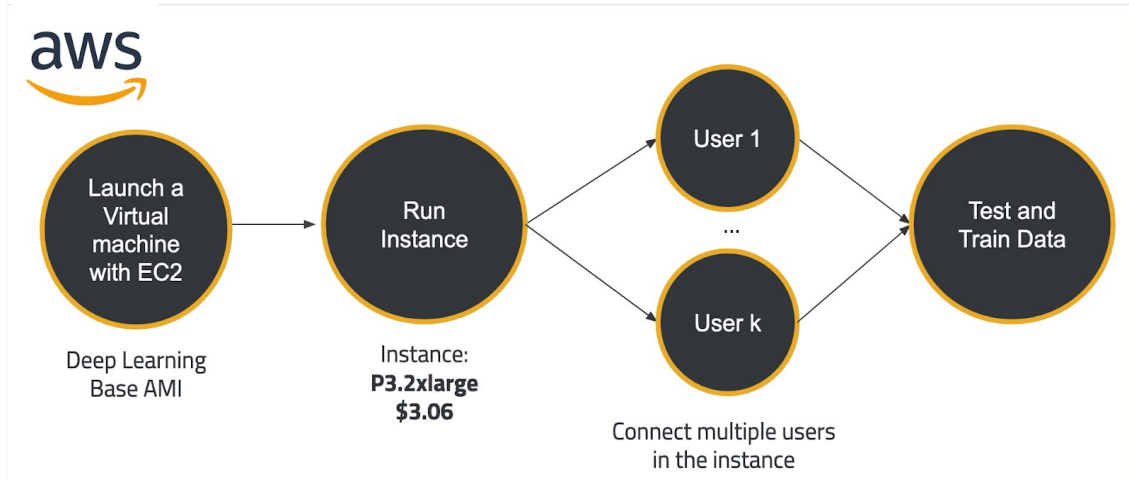
We utilized the existing software Blender 3D that is currently used to render 3D models. Still, we used it to mask our identified organs and used those masks to train our AI to automate this procedure.

2.3 Project Approach

The goal for the Fall 2021 semester was to learn the basics of 3-D modeling using 3D Slicer and determine different ways to automate it. Shortly after getting familiar with 3D Slicer, the team researched multiple ways to approach the problem of automation. After conducting some research, it was decided that a realistic and reasonable approach would be to use an existing application that uses AI. Consequently, the team learned about Nvidia Artificial Intelligence-Assisted Annotation (Nvidia AIAA), available within 3D Slicer, and tried implementing it. Nvidia is a technology company that creates graphics processing units (GPUs) and provides different applications such as the AIAA. Nvidia AIAA is an application programming interface (API) that allows users to create 3-D model objects using trained data conveniently. An API is analogous to an engine of a car, where one does not necessarily need to know the intricacies of the engine but turning the key in the ignition will run the car. Similarly, Nvidia AIAA allows users to generate 3-D objects by simply giving it an input, in this case, MRI images and a few manual operations.

Meanwhile, the focus for the Spring 2022 semester was to create an AI model. Through the experiments conducted during the Fall 2021 semester, one of the first obstacles to using Nvidia AIAA was the absence of a pre-trained AI model for the team's target organs: female pelvic organs such as the bladder, rectum, and vagina. For this reason, it was decided that the Spring 2022 semester will be focused on training our own AI model. This process was anticipated to be laborious. By the end of the Spring 2022 semester, the team was able to produce an AI model trained on pelvic organs that will facilitate a faster process for creating a better quality 3-D model.

3. System Architecture



When training and testing our data, we used Amazon Web Service. Per the Clara Train recommendation, we obtained the cloud GPU. The cloud GPU allowed people in the group to train and test our data regardless of whether or not they had a local GPU. To connect to the cloud GPU, we first made an AWS account. We then launch a virtual machine with EC2. Clara Train recommended the Deep Learning Base AMI, which provides a foundational platform for deep learning on AWS EC2 with NVIDIA CUDA. The instance used was p3.2xlarge, containing 1 GPU, 8vCPU, 61GB of memory, and up to 10GB of network performance. In theory, multiple people can access the instance and use its functions virtually by having a cloud GPU. Anyone who has access to the instance can proceed to train and test our data. To connect to the instance, it must first be launched, open the ssh client, and locate the private key. The command that is run after is `chmod 400<name of .pem file>` and connects to the instance using its public DNS. Once inside the instance, we can train and test our data.

4. Results and Conclusion

4.1 Fall Semester Results

After conducting experiments during the Fall 2021 semester, it was found that all Nvidia features did not run exactly the same on each execution. Some Nvidia features had runtime errors, a computational failure that can only be detected after the program runs, which was suspected of being caused by hardware or operator errors. The team's approach to fix this error was to re-implement the feature using different operating systems: Windows, macOS, and Linux. As expected from its pre-experiment complication, the auto-segmentation feature did not produce a significant result. After running multiple experiments, it was decided that the auto-segmentation feature will need to be re-examined once an AI model, trained using a substantial amount of MRI image data of pelvic organs, is created.

In comparing the results, the research focused on each methodology's time efficiency and quality accuracy. In terms of time efficiency, the results from each Nvidia feature on different operating systems were calculated by getting the average of elapsed time. All experiments implementing the manual thresholding feature were successful. As shown in Table 1, the average elapsed time of manual thresholding for trials 1 and 2 was substantially close. It was good to notice that Trial 2 in all experiments was quicker by almost a minute, and it was suspected to be caused by memorability bias. Since members already ran Trial 1, it was highly likely that the members might have done the second trial in a much faster phase.

Manual Thresholding		
Experiment No.	Elapsed Time (Trial 1)	Elapsed Time (Trial 2)
1	0:02:02	0:01:08
2	0:02:13	0:01:21
3	0:02:07	0:00:57
4	0:02:15	0:00:51

Table 1: Elapsed time (hh:mm:ss) to conduct manual thresholding using 3D Slicer.

In the same manner, experiments implementing the DEXTR3D feature were successful. As presented in Table 2, the data followed the same trend as the manual thresholding, where Trial 2 has a shorter elapsed time. The DEXTR3D methodology has an average elapsed time of 2 minutes 20 seconds, almost longer than a minute compared to manual thresholding.

DEXTR3D		
Experiment No.	Elapsed Time (Trial 1)	Elapsed Time (Trial 2)
1	0:01:20	0:01:03
2	0:03:44	0:03:00
3	0:01:45	0:01:42
4	0:03:14	0:02:48

Table 2: Elapsed time (hh:mm:ss) to conduct Nvidia DEXTR3D feature using 3D Slicer.

However, not all experiments implementing the DeepGrow feature were successful due to various factors such as hardware and operator issues. As seen in Table 3, there was at least one

successful experiment. Given that the AI model for pelvic organs has not been trained yet, it was decided that the DeepGrow feature was still a worthwhile feature to experiment more on in the future.

DeepGrow		
Experiment No.	Elapsed Time (Trial 1)	Elapsed Time (Trial 2)
1	6:45:00	5:18:00
2	-	-
3	12:55:00	50:49:00
4	19:37:00	20:13:00

Table 3: Elapsed time (hh:mm:ss) to conduct Nvidia DeepGrow feature using 3D Slicer.

Based on Figure 3, manual thresholding is the most time-efficient out of all the three methodologies tested. Although this is the case, the time difference between manual thresholding and Nvidia DEXTR3D is not that significant because it is believed that once these features are used on a larger scale, there will be hundreds of MRI images being scanned, this tiny difference will be negligible [8]. Manual thresholding also requires more human interaction than Nvidia features. Hence it is premature to conclude that traditional thresholding is more efficient overall. The data obtained from the DeepGrow methodology was not included in Figure 2 because it was inconsistent and had extreme values that will affect the visualization of the difference between the other methodologies.

Methodology Time Implementation Comparison

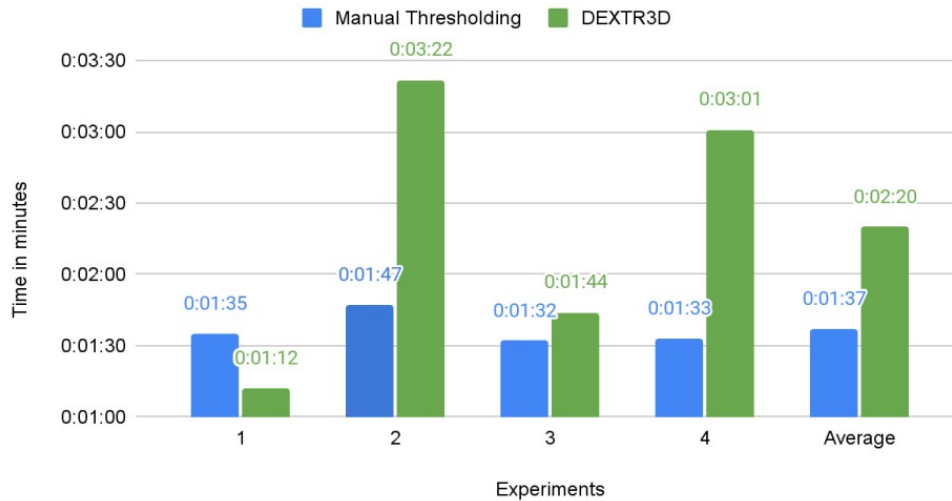


Figure 3: Time implementation comparison (hh:mm:ss) between manual thresholding and DEXTR3D

4.2 Spring Semester Results

First, the team created binary masks, which are essential in training our own AI model. As shown in figure 3, binary masks are 2-D drawings consisting of 1's and 0's laid on an MRI image that determines the shape of the target organ. The image pixels of the target organ are marked by 1's, and the surrounding parts are marked as 0's.

The team was also able to learn and use virtual GPUs through Amazon AWS. During the final stretch of the project, the team used the virtual GPU as part of the training process. As shown in Figure 4, the training process takes multiple iterations of calculations. The virtual GPUs can be thought of as virtual access to a computer that lives on one of Amazon's servers. This customized computer has all the hardware required in the project, specifically the Nvidia GPU, and can be used by multiple users simultaneously virtually.

Finally, the accuracy of the AI model was calculated through TensorBoard using the Dice Loss Function shown in Equation 1. As depicted in Figure 5, the loss was cut in half after 1,000 iterations and approached close to zero after 3,000 iterations.

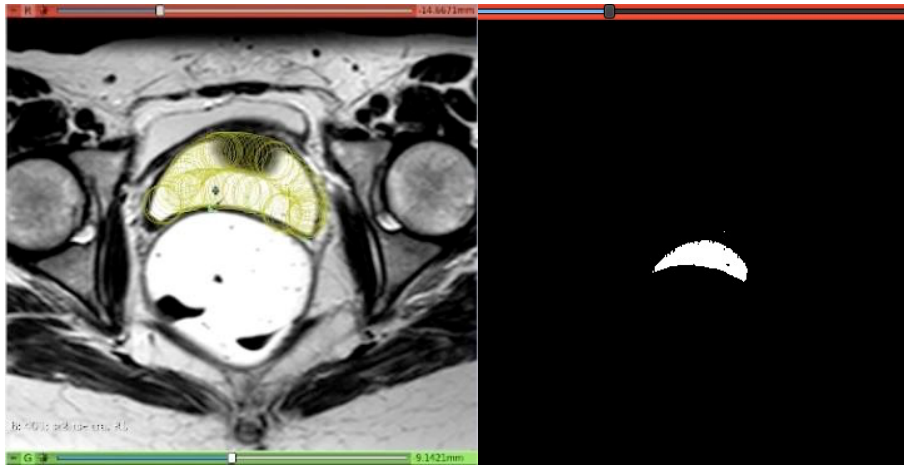


Figure 3: An MRI image highlighting the vagina (left); Binary label map of vagina (right)

```

2022-04-22 00:04:12,062 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 43/60 -- train loss: 0.0061 [220/887]
2022-04-22 00:04:13,337 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 44/60 -- train loss: 0.0089
2022-04-22 00:04:14,646 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 45/60 -- train loss: 0.0039
2022-04-22 00:04:15,950 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 46/60 -- train loss: 0.0034
2022-04-22 00:04:17,264 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 47/60 -- train loss: 0.0057
2022-04-22 00:04:18,557 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 48/60 -- train loss: 0.0054
2022-04-22 00:04:19,835 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 49/60 -- train loss: 0.0079
2022-04-22 00:04:21,136 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 50/60 -- train loss: 0.0066
2022-04-22 00:04:22,448 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 51/60 -- train loss: 0.0092
2022-04-22 00:04:23,663 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 52/60 -- train loss: 0.0075
2022-04-22 00:04:24,930 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 53/60 -- train loss: 0.0041
2022-04-22 00:04:26,184 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 54/60 -- train loss: 0.0092
2022-04-22 00:04:27,503 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 55/60 -- train loss: 0.0065
2022-04-22 00:04:28,775 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 56/60 -- train loss: 0.0080
2022-04-22 00:04:30,080 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 57/60 -- train loss: 0.0053
2022-04-22 00:04:31,371 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 58/60 -- train loss: 0.0090
2022-04-22 00:04:32,605 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 59/60 -- train loss: 0.0076
2022-04-22 00:04:33,563 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 97/100, Iter: 60/60 -- train loss: 0.0050
2022-04-22 00:04:33,564 - ignite.engine.engine.SupervisedTrainer - INFO - Got new best metric of train dice: 0.995022177696228
2022-04-22 00:04:33,564 - ignite.engine.engine.SupervisedTrainer - INFO - Current learning rate: 0.0001
2022-04-22 00:04:33,564 - ignite.engine.engine.SupervisedEvaluator - INFO - Engine run resuming from iteration 0, epoch 96 until 97 epochs
2022-04-22 00:04:36,881 - ignite.engine.engine.SupervisedEvaluator - INFO - Got new best metric of val dice: 0.9762508273124695
2022-04-22 00:04:36,881 - ignite.engine.engine.SupervisedEvaluator - INFO - Epoch[97] Metrics -- val dice: 0.9763
2022-04-22 00:04:36,881 - ignite.engine.engine.SupervisedEvaluator - INFO - Key metric: val dice best value: 0.9762508273124695 at epoch: 97
2022-04-22 00:04:36,882 - deepgrow.handler - INFO - Epoch[97] Metrics -- Region: 01, val dice: 0.9763
2022-04-22 00:04:36,918 - ignite.engine.engine.SupervisedEvaluator - INFO - Epoch[97] Complete. Time taken: 00:00:03
2022-04-22 00:04:36,921 - ignite.engine.engine.SupervisedEvaluator - INFO - Deleted previous saved final checkpoint: model_final_iteration=7.pt
2022-04-22 00:04:36,952 - ignite.engine.engine.SupervisedEvaluator - INFO - Train completed, saved final checkpoint: model_final_iteration=7.pt
2022-04-22 00:04:36,952 - ignite.engine.engine.SupervisedEvaluator - INFO - Engine run complete. Time taken: 00:00:03
2022-04-22 00:04:36,997 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch[97] Metrics -- train dice: 0.9950
2022-04-22 00:04:36,997 - ignite.engine.engine.SupervisedTrainer - INFO - Key metric: train dice best value: 0.995022177696228 at epoch: 97
2022-04-22 00:04:36,997 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch[97] Complete. Time taken: 00:01:22
2022-04-22 00:04:39,180 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 1/60 -- train loss: 0.0088
2022-04-22 00:04:40,490 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 2/60 -- train loss: 0.0069
2022-04-22 00:04:41,833 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 3/60 -- train loss: 0.0054
2022-04-22 00:04:43,175 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 4/60 -- train loss: 0.0079
2022-04-22 00:04:44,524 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 5/60 -- train loss: 0.0053
2022-04-22 00:04:45,849 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 6/60 -- train loss: 0.0066
2022-04-22 00:04:47,148 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 7/60 -- train loss: 0.0086
2022-04-22 00:04:48,484 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 8/60 -- train loss: 0.0044
2022-04-22 00:04:49,815 - ignite.engine.engine.SupervisedTrainer - INFO - Epoch: 98/100, Iter: 9/60 -- train loss: 0.0029

```

Figure 4: A screenshot of terminal command showing the individual steps taken during the training process

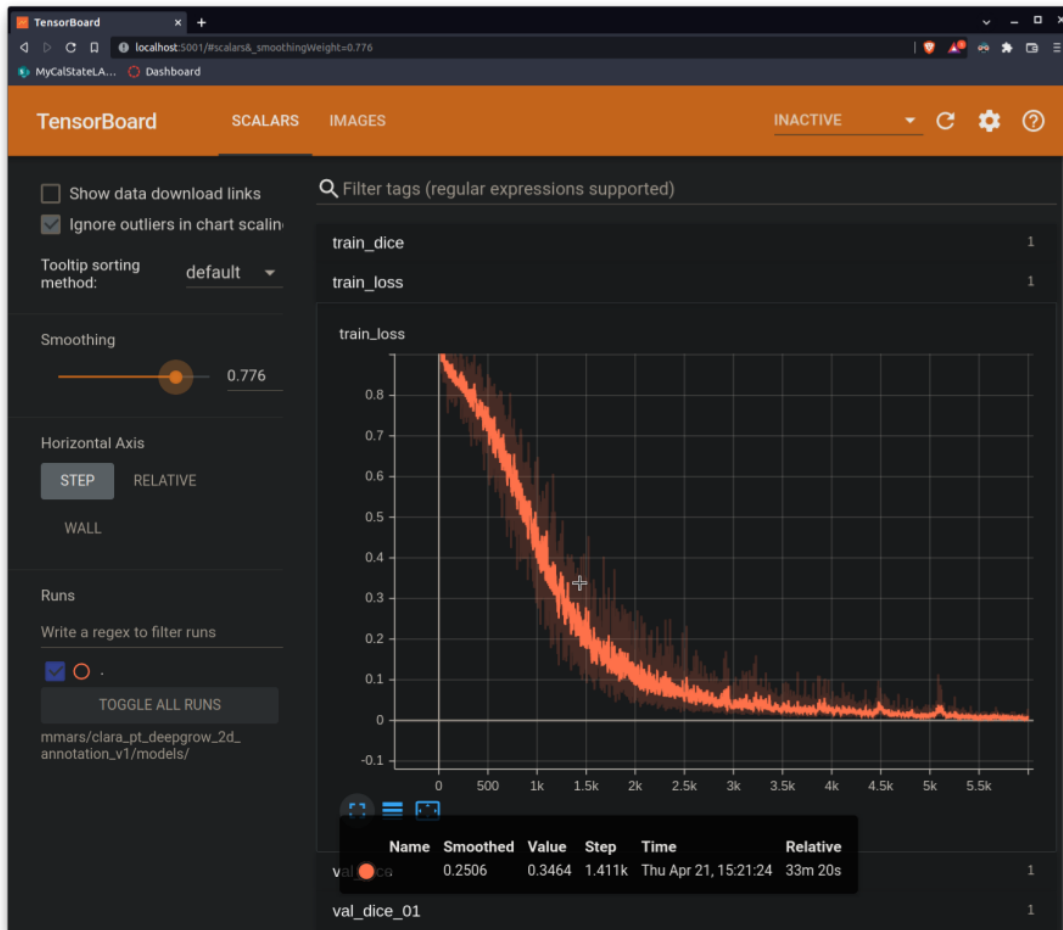


Figure 5: A graph showing accuracy loss of the AI model using TensorBoard

$$Dice = \frac{2 \times TP}{(TP + FP) + (TP + FN)}$$

Equation 1: The Dice Loss Function

4.3 Conclusion

The biggest challenge in this project was creating our own AI model. In the attempt to train our own model, the team came across multiple technical issues as well as non-technical challenges. On top of the expected learning curve, the creation of the AI model had myriad requirements, both in hardware and software. The team inevitably purchased a virtual GPU through AWS to be able to train our model. Aside from that, we also encountered challenges in communicating within the team and with other people outside the group. As a team, we struggled to get everyone engaged and comfortable speaking out during meetings, especially during the early stages of the project. As the project lead, I needed to step up and keep everyone engaged by constantly coming in prepared with agendas during meetings and holding everyone accountable for their tasks.

The team achieved its second-semester goal of producing its first pre-trained AI model of pelvic organs. As demonstrated throughout the project, the work necessary to automate the process of creating 3-D models using artificial intelligence can be tedious but very satisfying. On top of starting this project from scratch, as a team, we have accomplished the following: learning and applying basic 3-D model construction using 3D Slicer, familiarizing ourselves with the female pelvic organs and identifying them given an MRI image, experimenting on Nvidia AIAA and its features, and ultimately, getting closer to having our own pre-trained AI model for the pelvic organs.

The team envisioned creating a more user-friendly web application to automate the MRI image conversion that is powered by artificial intelligence. Since Nvidia AIAA was not 100% compatible when we started the project, the team spent most of the Spring 2022 semester

creating our AI model. This task of building a web application will be entrusted to the next group that will handle this project.

5. References

1. Edelman, R. R. (2014, October 23). The history of MR imaging as seen through the pages of radiology. Retrieved May 7, 2022, from <https://pubs.rsna.org/doi/10.1148/radiol.14140706>
2. Sakinis, Tomas, et al. "Interactive segmentation of medical images through fully convolutional neural networks." arXiv preprint arXiv:1903.08205 (2019).
3. Zwanenburg*, A. (2020, March 10). The image Biomarker Standardization Initiative: Standardized quantitative Radiomics for high-throughput image-based phenotyping. Retrieved May 7, 2022, from <https://pubs.rsna.org/doi/full/10.1148/radiol.2020191145>
4. Vadmal, V., Junno, G., Badve, C., Huang, W., Waite, K. A., & Barnholtz-Sloan, J. S. (2020, April 14). MRI image analysis methods and applications: An algorithmic perspective using brain tumors as an exemplar. National Library of Medicine. Retrieved May 7, 2022, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7236385/>
5. Talukder, S., & Noyes, N. (2021). Analyzing brain tumor MRI to demonstrate GPU-based medical ... - cs.siu.edu. Retrieved May 7, 2022, from https://www2.cs.siu.edu/~stalukder/publications/GPU_Based_Image_Segmentation_for_Medical_Field.pdf
6. 3D Slicer Documentation. URL <https://slicer.readthedocs.io/en/latest/>
7. DicomCleaner Documentation. URL <http://www.dclunie.com/ixelmed/software/webstart/DicomCleanerUsage.html>
8. NVIDIA Clara Train SDK. URL <https://catalog.ngc.nvidia.com/orgs/nvidia/collections/claratrainframework>
9. NVIDIA AI-Assisted Annotation. URL <https://docs.nvidia.com/clara/clara-train-sdk/aiaa/index.html>