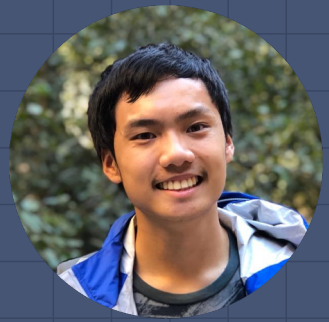
Pelvic Image Analysis and Geometry Reconstruction using Artificial Intelligence

Client: Dr. Mathias Brieu (Mechanical Engineering)

Advisor: Dr. Negin Forouzesh (Computer Science)

Meet the Team

Ralph Belleca
Project Lead, Front-end
rbellec@calstatela.edu



Robin Mok
Front-end
rmok2@calstatela.edu

Alejandra Olvera
Front-end
aolver14@calstatela.edu

Sabino Ramirez
Front-end
srami207@calstatela.edu

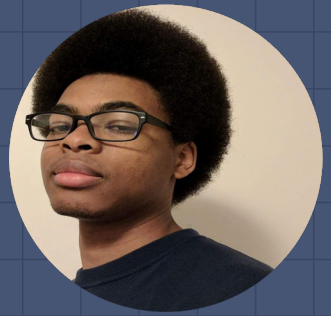
Mary Semerdjian
Front-end Lead
msemerd2@calstatela.edu

Meet the Team

Nicol Barrios
Back-end Lead
abarri53@calstatela.edu

Ted Kim
Back-end
tkim56@calstatela.edu

Silvano Medina
Back-end
Documentation Lead
smedin63@calstatela.edu

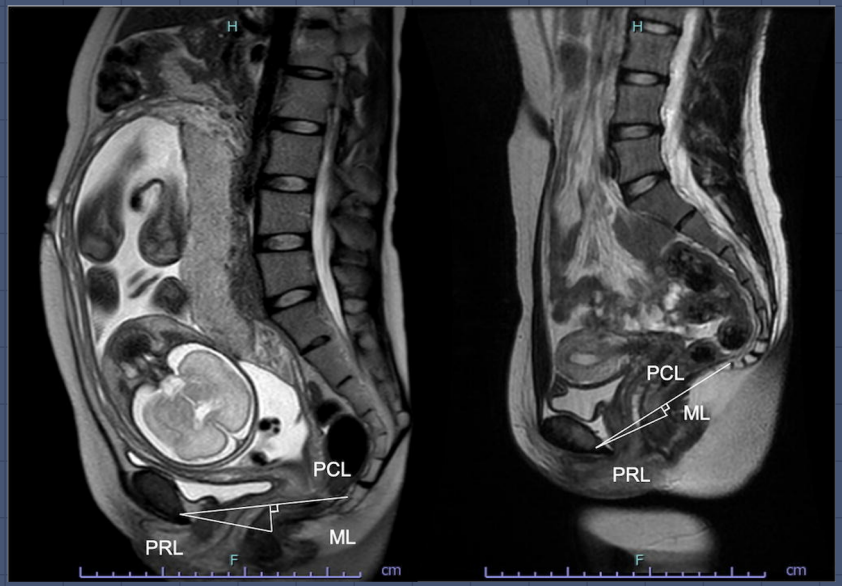


Demetrius Parker
Back-end
dparke11@calstatela.edu

Jason Tejada
Back-end,
Customer Liaison
jtejad12@calstatela.edu

Context

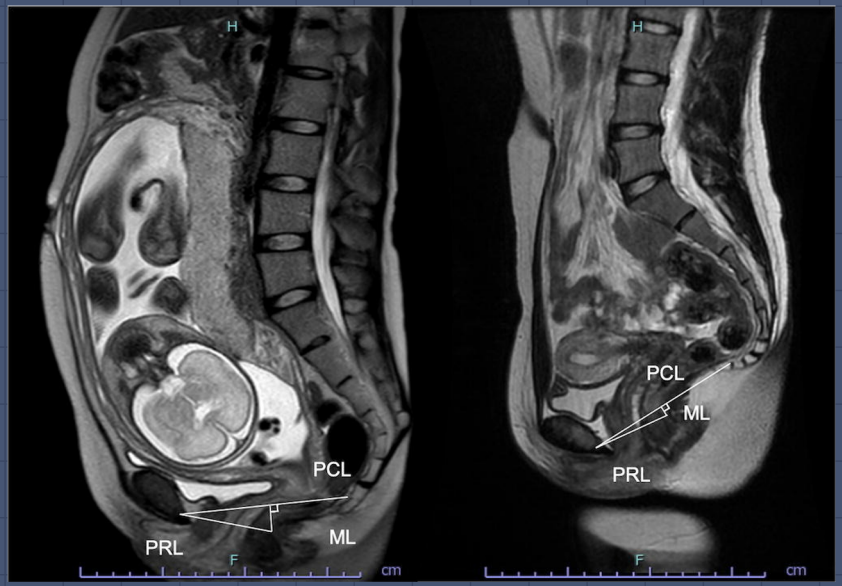
Example of Medical Imaging



MRI image taken during pregnancy

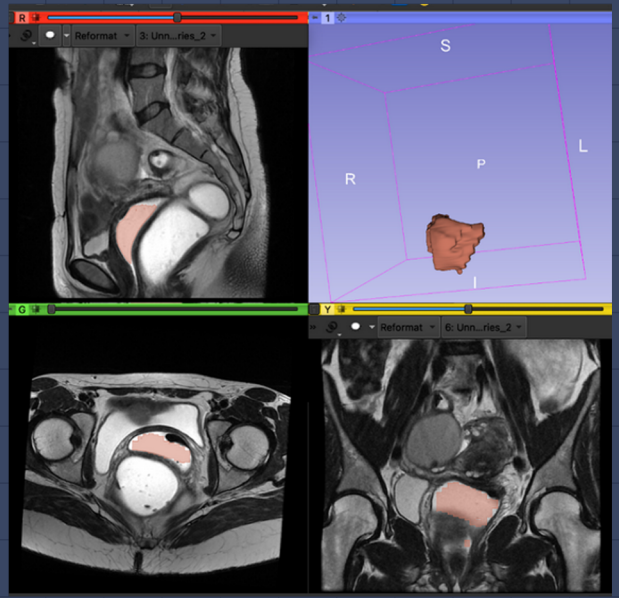
Context

Example of Medical Imaging



MRI image taken during pregnancy

Project's Focus: MRI Scan



Performing image segmentation on an MRI image with vagina as target organ

Ralph B.

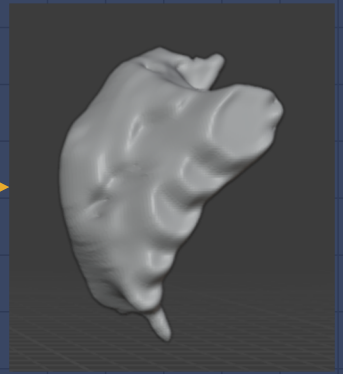
Problem

Traditional methodology is tedious and requires various repetitive manual procedures.

MRI Input

Manual Thresholding

Other Manual Refinement



3D Object of Pelvic Organ (Vagina)

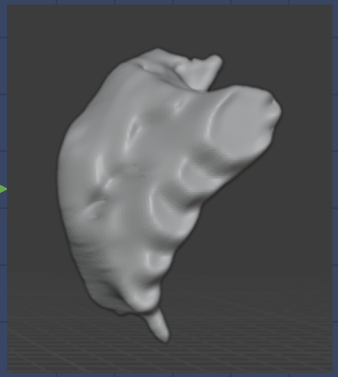
Goal

Streamline the process of converting MRI images of pelvic organs into 3-D models using Artificial Intelligence

MRI Input

NVIDIA AIAA

Semi-automatic thresholding using custom AI model

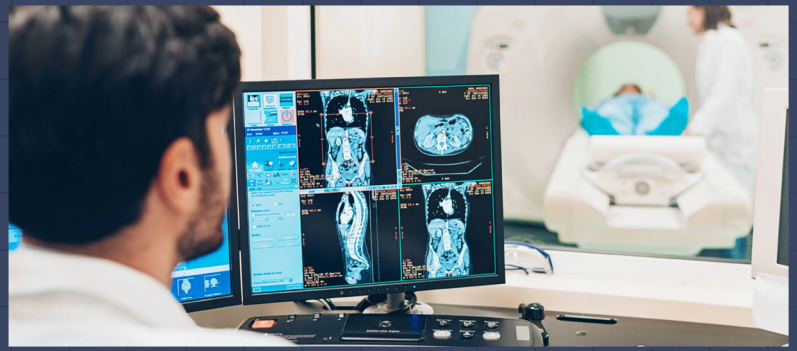


3D Object of Pelvic Organ (Vagina)

Ralph B.

Significance

Allow medical professionals to do image analysis in a faster and more effective way by simplifying the creation of 3-D models.



Approach for Spring Semester

System Requirements and Installation

Highlighting and labeling region of interest

Learns to create model using dataset

Model used to help create organ of interest in **3D Slicer**

Approach for Spring Semester

Use AI-Assisted Annotation to create pelvic organs using data in MRI scans given to the program

Use Clara Train to Create Custom 3D AI - Generated Model of pelvic organs

Start of Spring

Determined to use Deepgrow

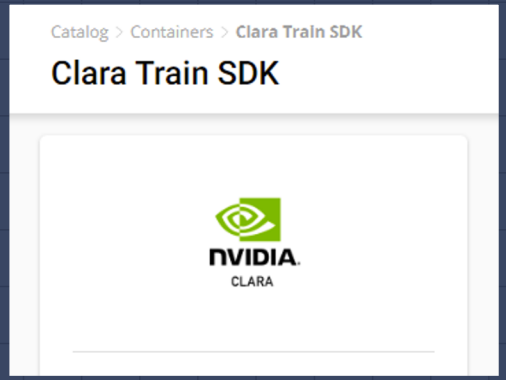
Goals:

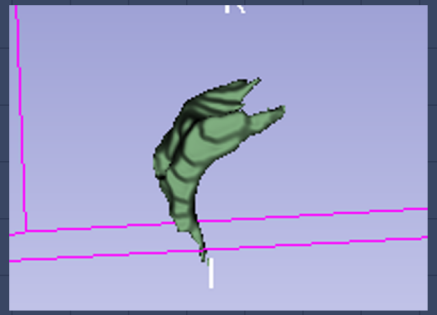
-Use Clara Train

-Find & mask Data Points

-Training & Creating Usable model

-Use Trained model with Deepgrow





Example of 3D model

Installation of Core Programs

-Installation

-Github

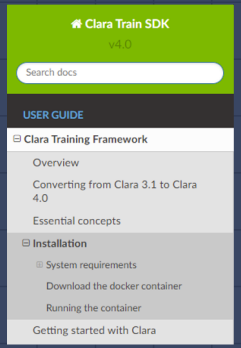
-NVIDIA's website

-Requirements

-Linux

-NVIDIA GPU

-Data Points



Clara Train SDK Documentations

Image source: https://en.wikipedia.org/wiki/Linux https://www.leadtek.com/p_images/zoom/10775_1M.jpg

Data Points

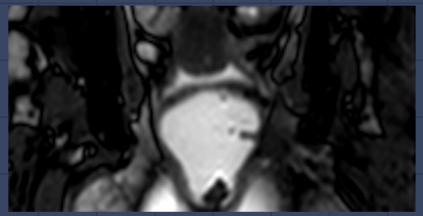
-MRI Scans

-Usable inputs

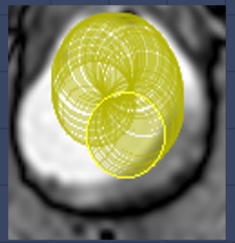
-Test Models

-Masking and labeling of Data Points

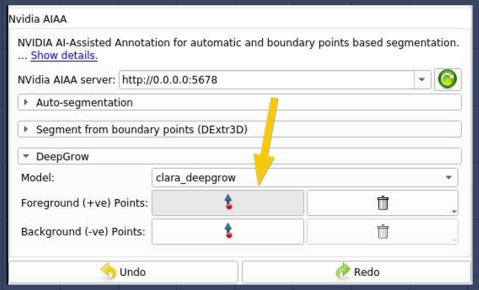
-Training



Examples of Scan

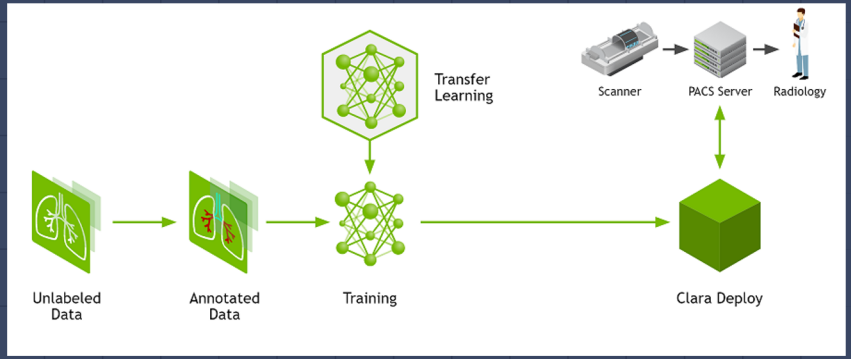


Masking/Labeling



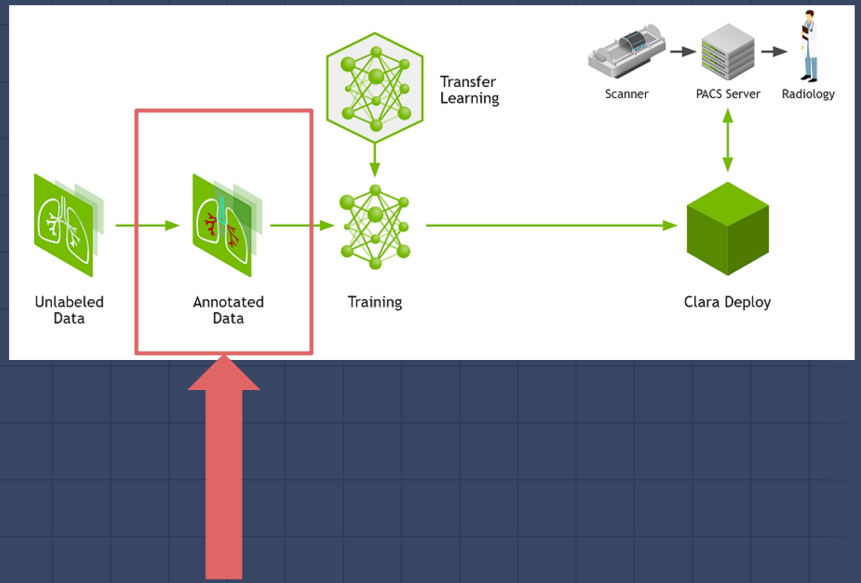
Clara Train (Version 4.0)

An application framework that allows users to quickly get started for annotating, training, and adapting AI models while providing pre-trained models.



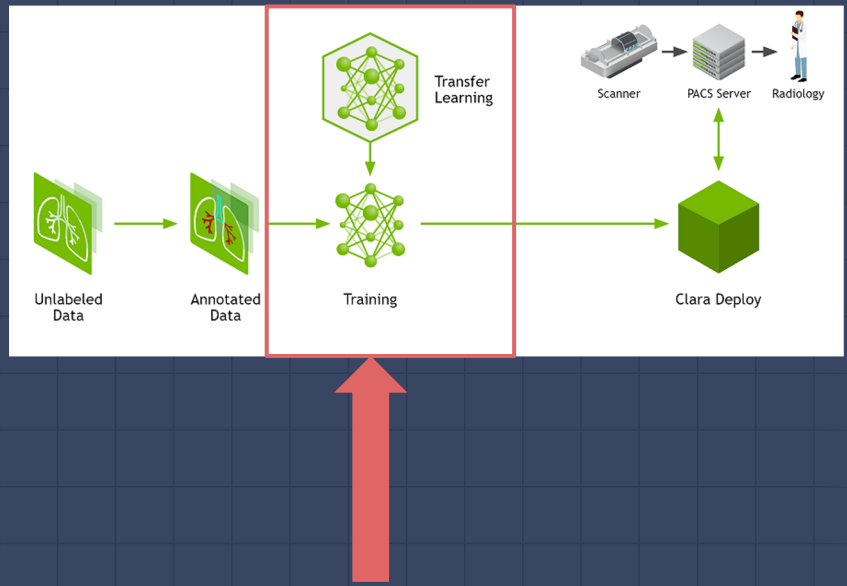
Clara Train (Version 4.0)

An application framework that allows users to quickly get started for annotating, training, and adapting AI models while providing pre-trained models.



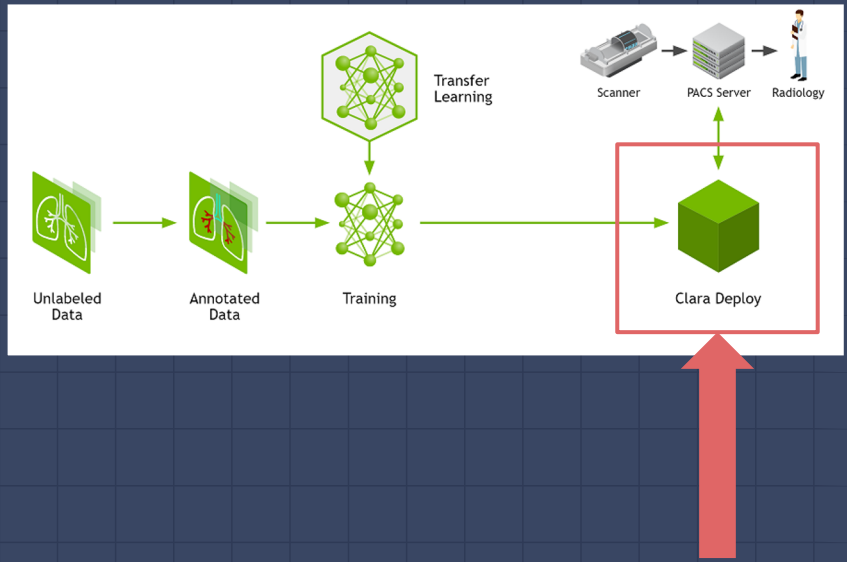
Clara Train (Version 4.0)

An application framework that allows users to quickly get started for annotating, training, and adapting AI models while providing pre-trained models.



Clara Train (Version 4.0)

An application framework that allows users to quickly get started for annotating, training, and adapting AI models while providing pre-trained models.

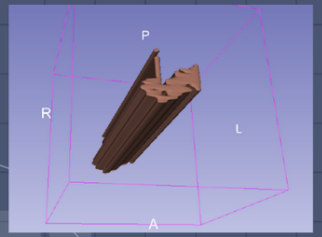


Masking and Binary Labels

Previously last semester...

DeepGrow

Most accurate but not reliable due to quality inconsistency and amount of time.



Our results from last semester were all based on a brain tumor model that the AIAA plugin provided for us. This semester we chose to use our own pelvic model that we had to create.

Masking and Binary Labels

In order to get the best model generated by Nvidia AIAA plug-in we need to create our own model.

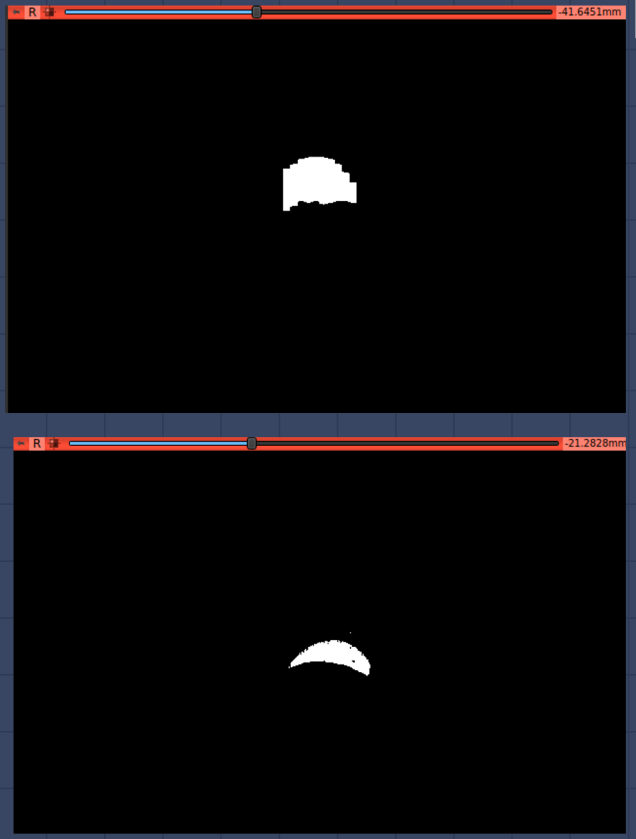
To do this we need to use the Nvidia Clara Train Framework to create a model of the organs we specify and train the A.I. on that model.

To specify the organ we create a binary labelmap using the masking process.

Masking and Binary Labels

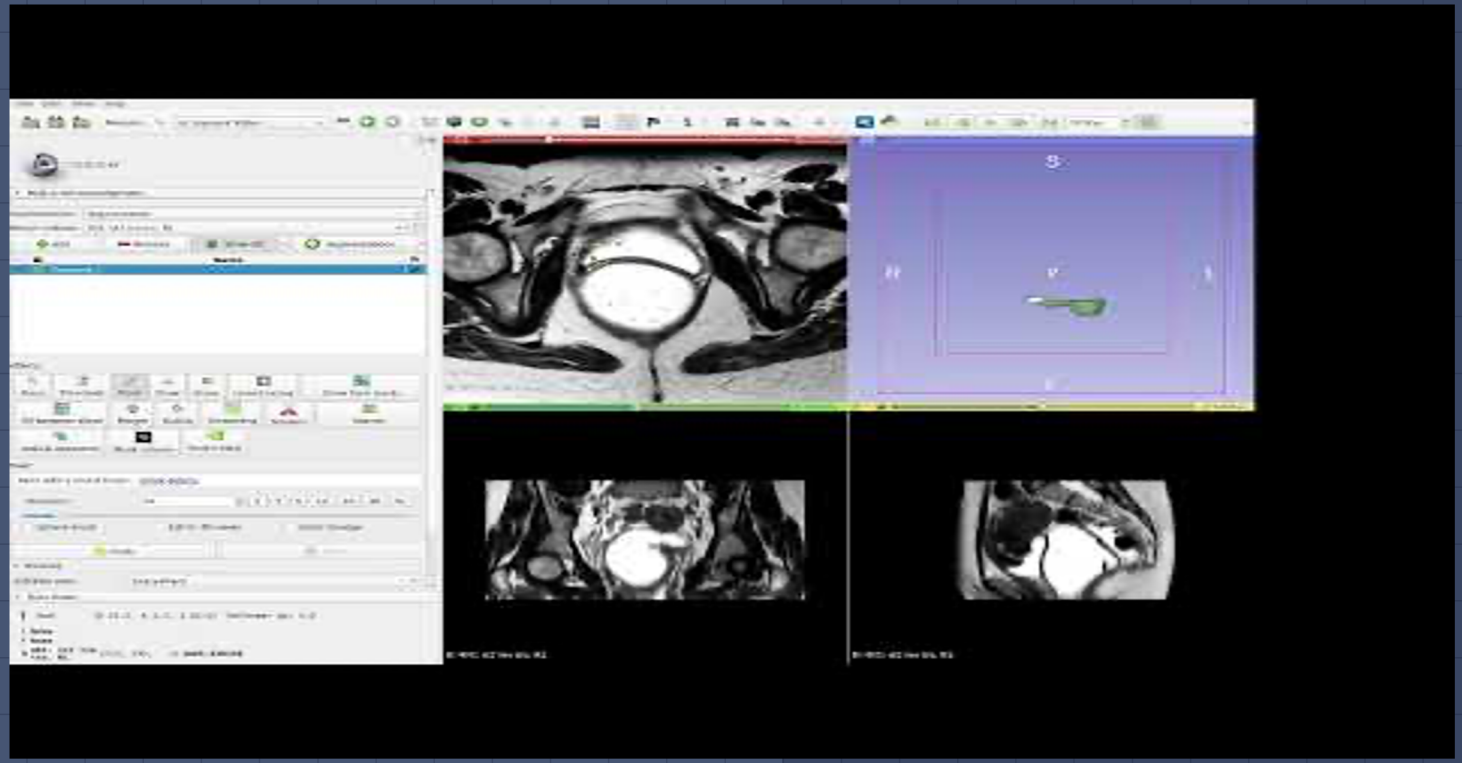
Binary labelmaps are created in 3D Slicer using a process called masking.

Masking is the process of blanking out a segment or area in a volumetric image to show only a selected organ. It can be used for creating a binary labelmap for registration, bias correction, etc.



Binary labelmaps for bladder(top) and vagina(bottom)

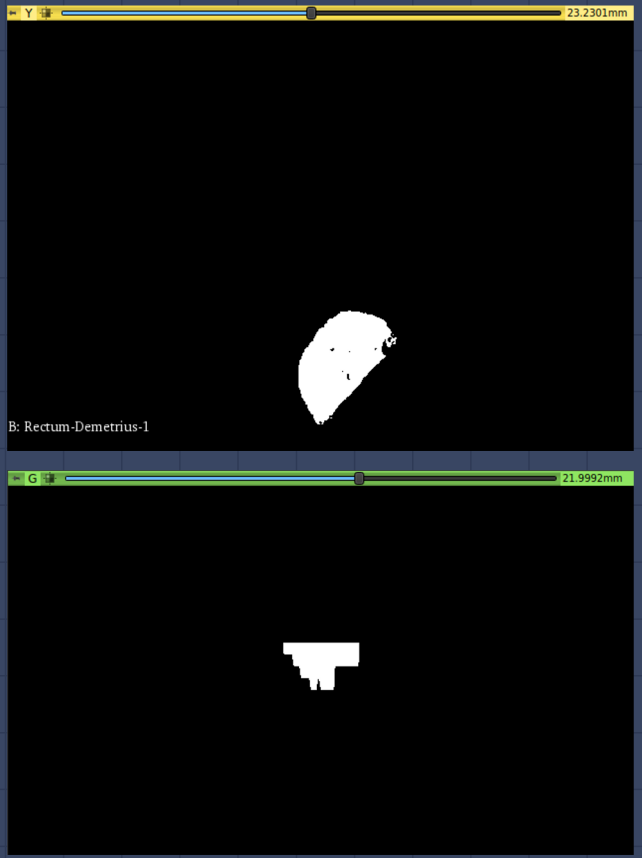
How-To Mask



Demetrius P.

Masking and Binary Labels

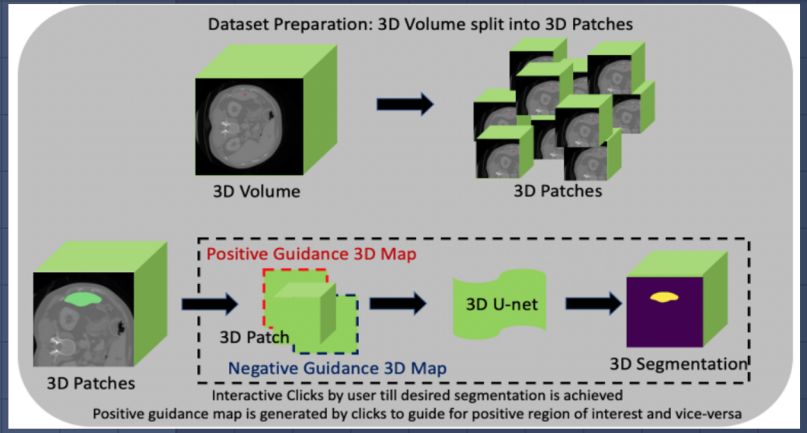
Once the labelmap have been created it is ready to be used by the Clara Train Framework.



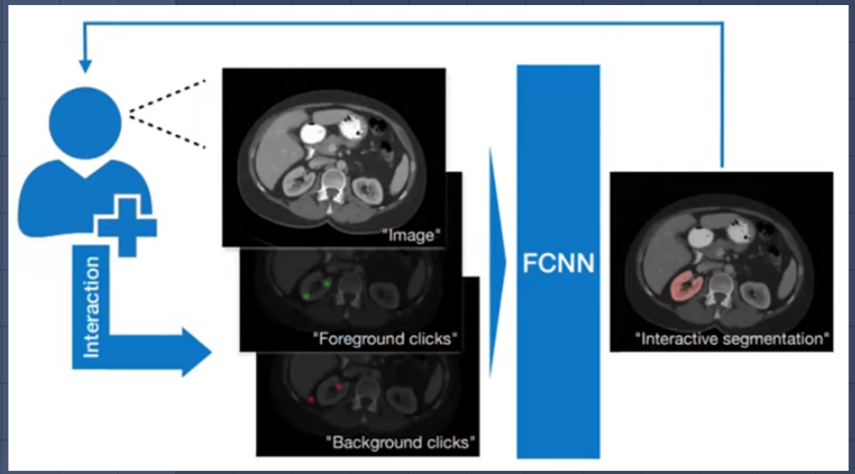
Binary labelmap for rectum(top) and bladder (bottom)

Nvidia DeepGrow

- A pretrained model from Nvidia that applies segmentation based on foreground and background clicks by the user.
- Built around the 3D U-Net convolutional neural network architecture.
- Output consists of a single channel containing a 3D semantic segmentation represented by a binary label system.

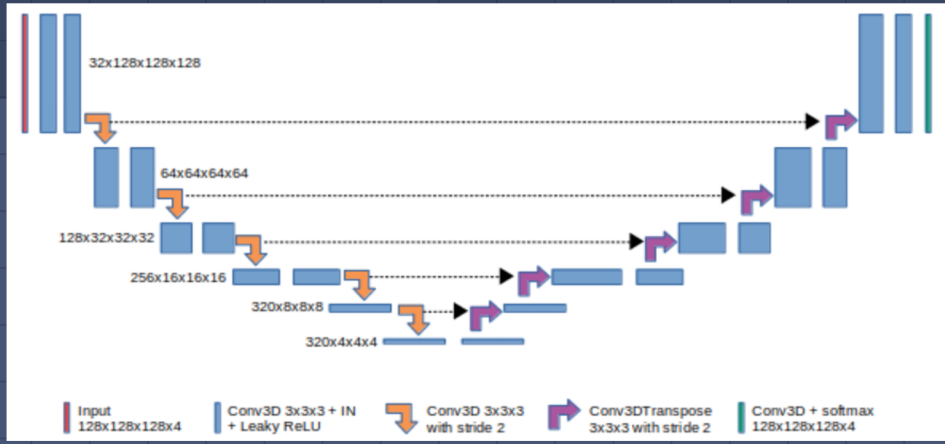


Top: From Nvidia's site showing data preparation. Bottom: annotation workflow from user's perspective.



3D U-Net

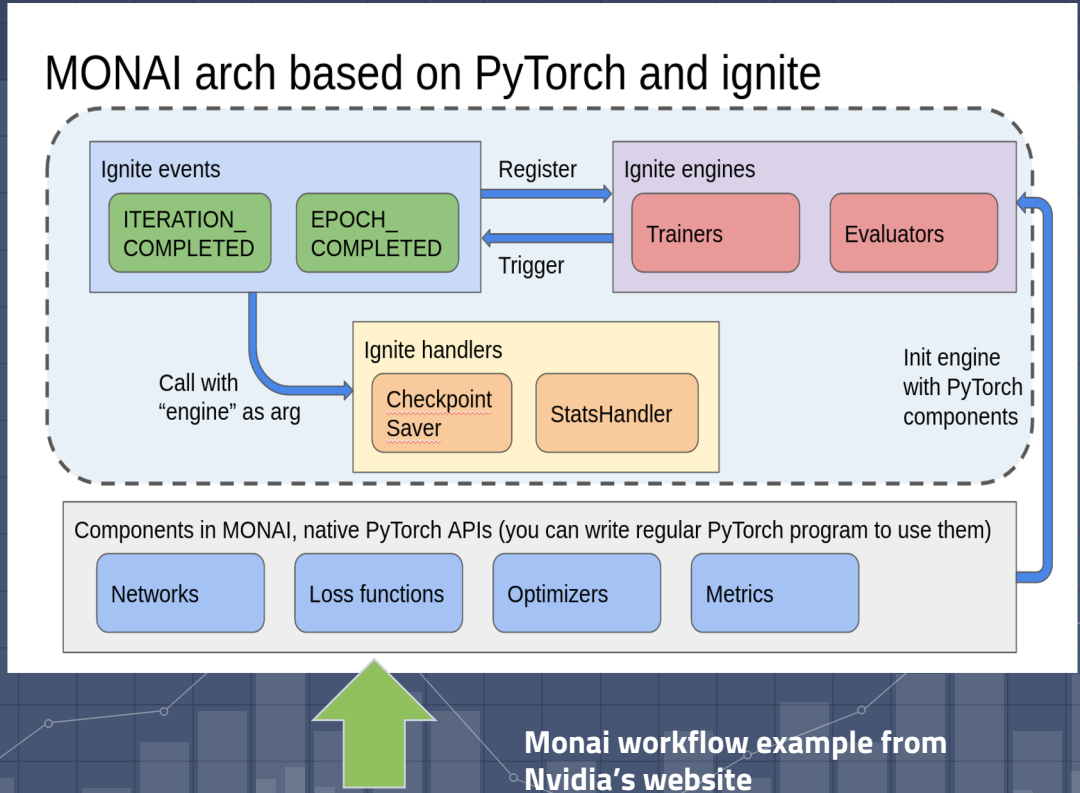
- An encoder-decoder network that can return accurate and detailed segmentations.
- The **Encoder** uses convolution and max pooling layers to downsample the image.
- The **Decoder** combines info from the bottom of the "U" with high res. Feature maps output by the encoder levels via skip connections.
- Does not require a large dataset.



Top: Graphical representation of untet architecture. Bottom: example of down sampling

Monai Framework

- PyTorch-based open source framework for deep learning in medical imaging.
- The main components we see are Loss Functions, Optimizer, Metrics, Transforms, and Handlers.
- For instance, the default loss function for Clara Train 3D DeepGrow is DiceLoss which will provide useful metrics.
- Engine is a trainer, validator, or evaluator and initiate a loop. Within the loop, events get triggered and the attached handlers get called.



Monai workflow example from Nvidia's website

Training With Clara Train

- The 3D DeepGrow MMAR provides scripts in the “commands” directory.
- **./train.sh** is a script that begins the training work flow set up by the preceding configuration steps.
- The configuration is also responsible for initializing any Monai components to be used throughout the training process.

```
"DATA_ROOT": "/",
"DATASET_JSON": "/workspace/senior_design/data/tmp4dg2d/dataset_0.json",
"PROCESSING_TASK": "segmentation",
"MMAR_EVAL_OUTPUT_PATH": "eval",
"MMAR_CKPT_DIR": "models",
"MMAR_CKPT": "models/model.pt",
"MMAR_TORCHSCRIPT": "models/model.ts"
```

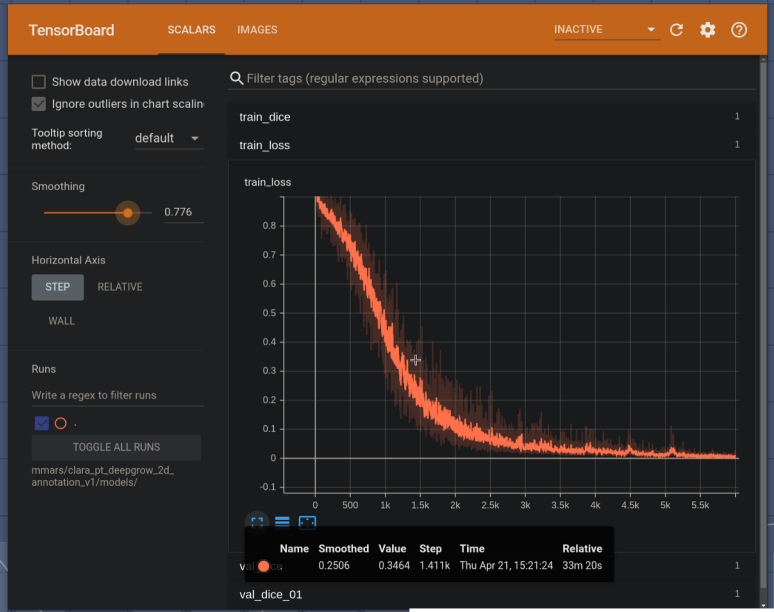
Sabino Screenshot of environment.json config

```
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: 0.0181
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
```

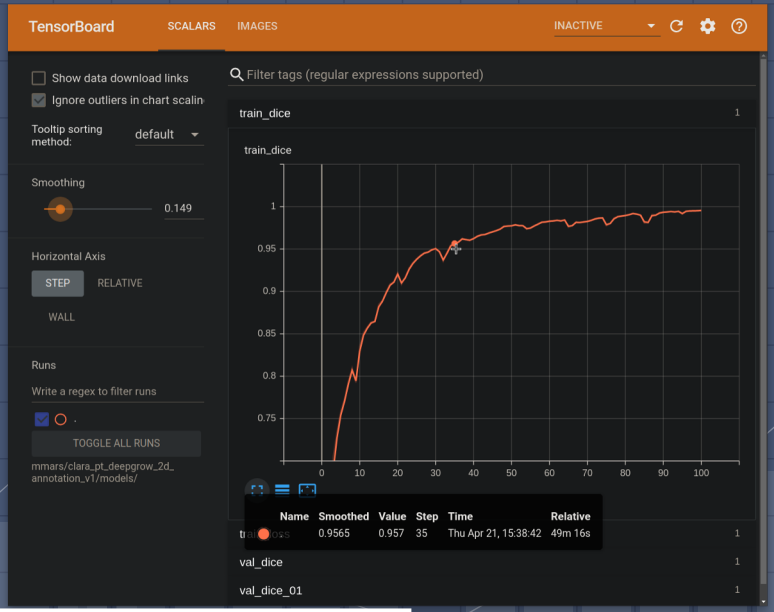
Screenshot of the terminal during training process.

Results

Train_loss



Train_dice



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

Spring Semester Challenges

Creating our own AI model

- Since Nvidia AIAA doesn't have any model trained with Pelvic organs (Nvidia has model for brain, lungs, spleen, etc.), we had to train our own models.

Nicol B.

Spring Semester Challenges

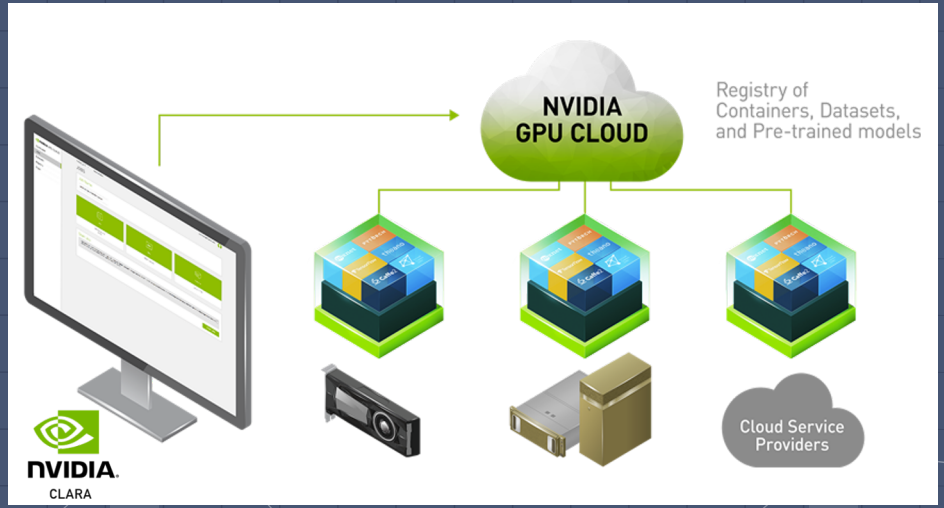
Configuration → Masking → Training → Integration to 3D Slicer

Local GPU

Virtual GPU

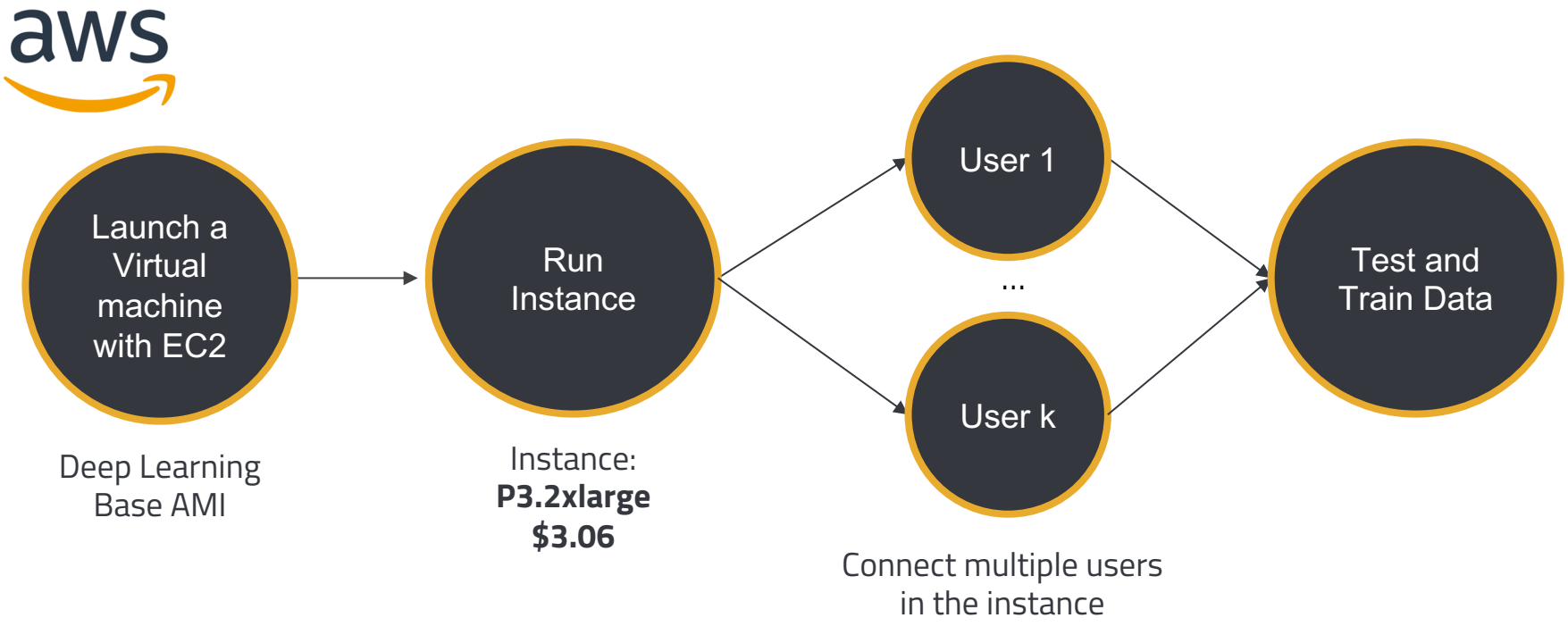
Cloud GPU

- In order to use Clara Train SDK which is built on top of MONAI, we need and NVIDIA GPU.
- The Cloud GPU is required for training and testing our data.
- Requirements for GPU:
  - V100 Tensor Core
  - 16GB or 32GB Memory
  - 8 vCPU

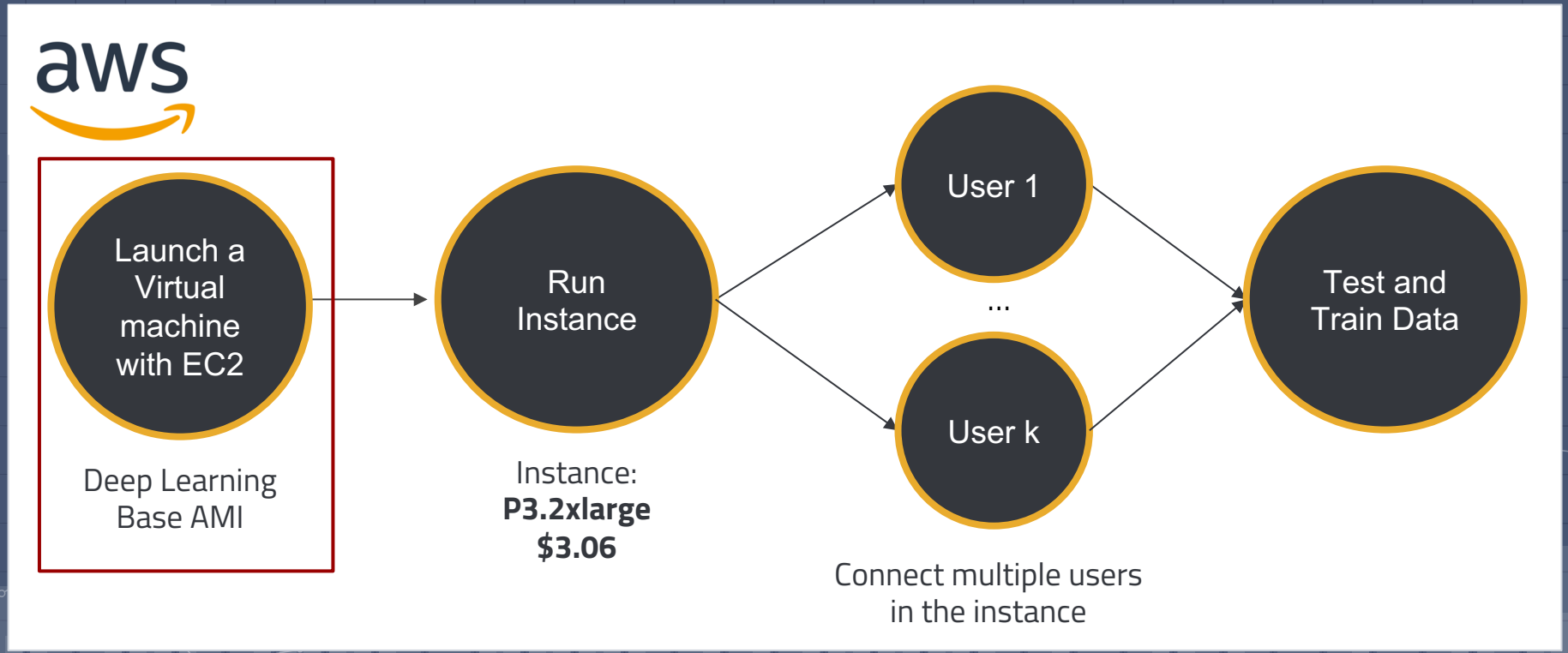


Cloud GPU Diagram from NVIDIA Website

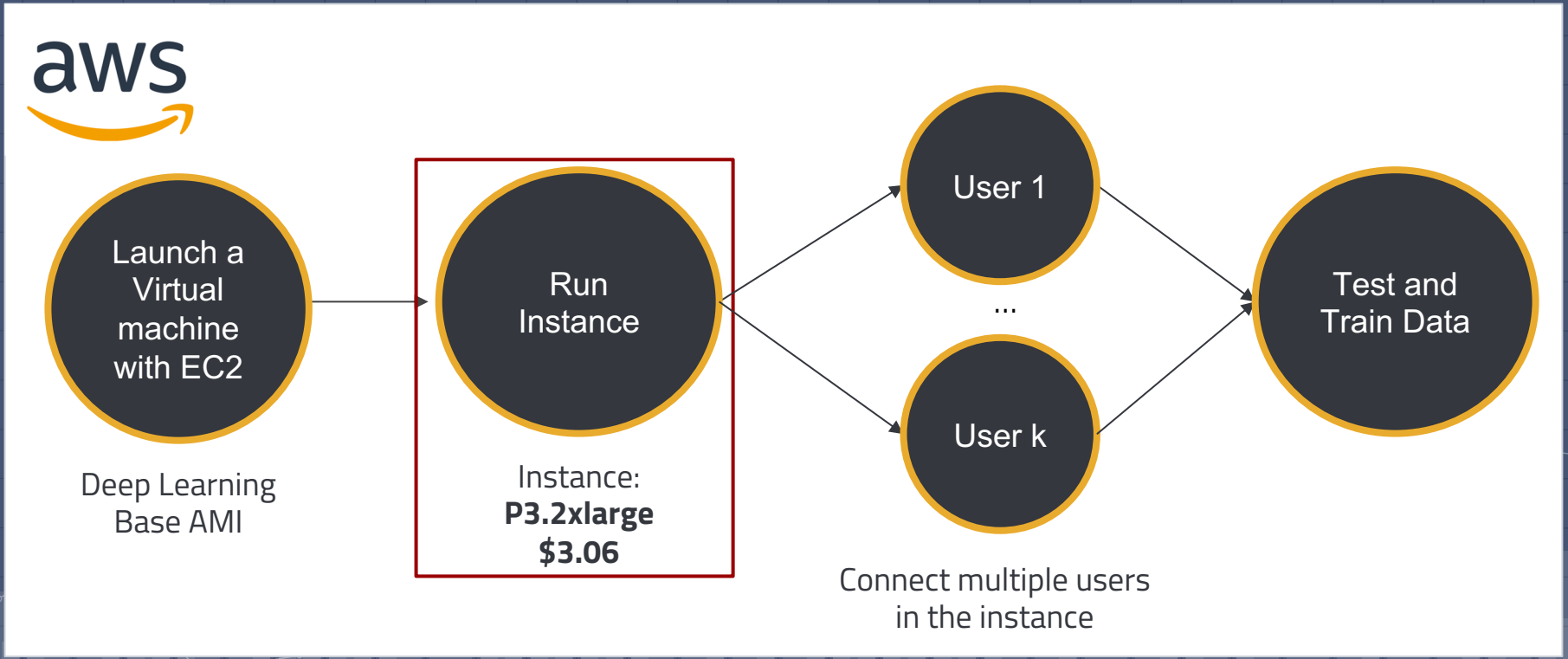
Amazon Web Services



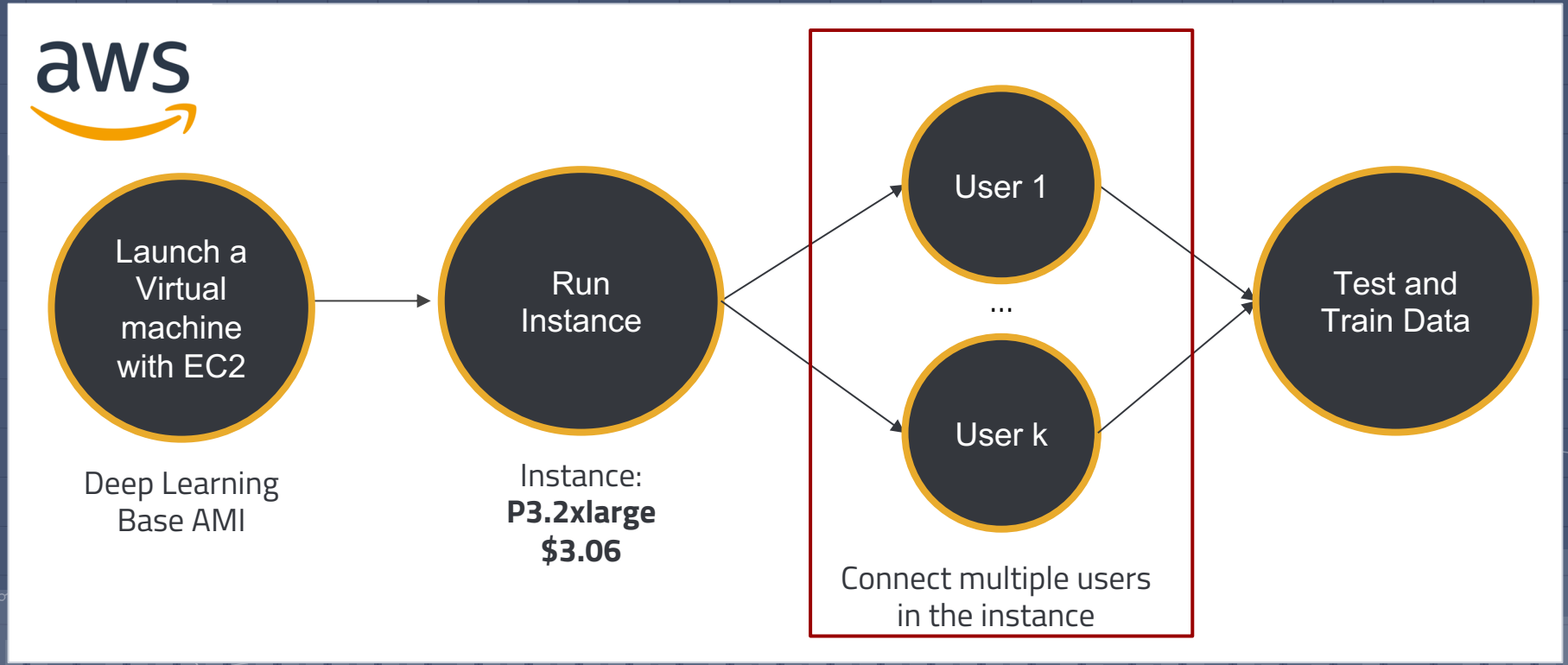
Amazon Web Services



Amazon Web Services



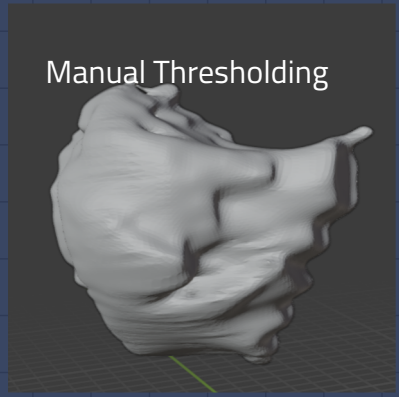
Amazon Web Services



Fall Semester Achievements

Researching & Understanding NVIDIA AIAA

- Traditional approach
- Decided to use NVIDIA AIAA to automate the modeling process
- Needed a model trained on pelvic organ data points



Jason T.

Spring Semester Achievements

Four Stages & New Model

- Determine & Setup the right configuration
- Creating binary masks
- Training the model
- Alternative to physical GPU

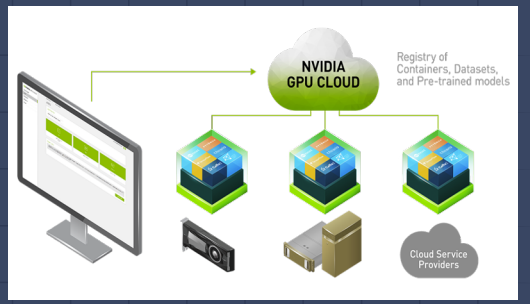
Configuration

Masking

Training

Integration to 3D Slicer

Binary labelmap for vagina



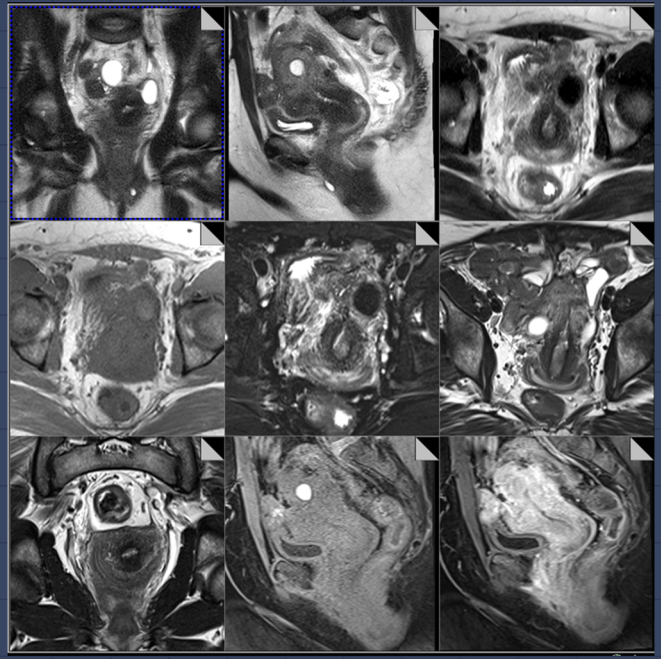
Next Steps

Short Term

- Gather more MRI data
- Improving our AI

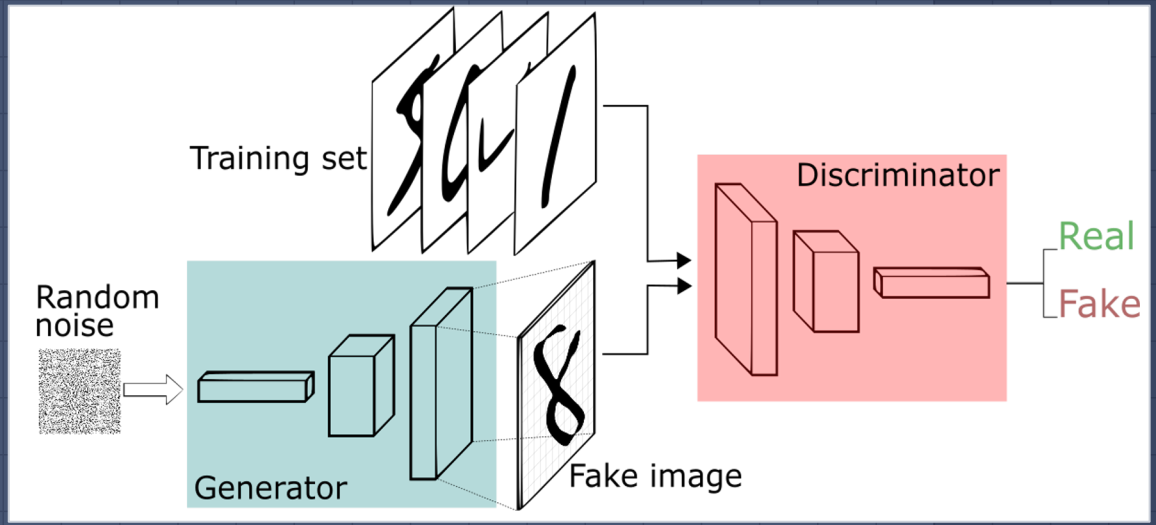
Long Term

- Create a Web Application



GANs

Generative Adversarial Network



Works Cited

NVIDIA Clara Imaging.

https://developer.nvidia.com/clara-medical-imaging (2022).

Image Source: https://www.iinn.com/wp-content/uploads/sites/4/2019/10/Insight-Medical-Campus-_-Blog-Difference-Between-MRI-and-CT.jpg

Pelvic Fracture

X-ray of the pelvis, pre-operative, displaying a variety of pelvic fractures by Dr. Brent Burbridge MD, FRCPC, University Medical Imaging Consultants, College of Medicine, University of Saskatchewan is used under a CC-BY-NC-SA 4.0 license.

https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394

Thalles Silva

Image Source: https://www.iinn.com/wp-content/uploads/sites/4/2019/10/Insight-Medical-Campus-_-Blog-Difference-Between-MRI-and-CT.jpg

Thank you for listening!

Any Questions?