Group 9 Checkpoint (March 16): Predicting Hemorrhage Related Outcomes with CT Volumetry for Traumatic Hemothorax

- **Students:** Benjamin Albert, Chang Yan, Gary Yang
- **Mentors:** Dr. David Dreizin, Dr. Mathias Unberath

Brief Recap of Project

- Developing software tool to automatically quantify blood volume in a hemothorax (blood in the pleural cavity around lungs)
- To do this, segment hemothorax from chest CT scans
    - After segmentation, count the segmented voxels which have known units of volume
    - Segmentation methods are using deep encoder-decoder CNNs
        - Additionally experimenting with attention-based networks recommended by Dr. Dreizin
        - Also experimenting with causal model as recommended by Dr. Unberath

Brief Recap of Project Outcomes

Minimal: input chest CT scan to output hemothorax volume

Expected: (Minimal) with < 5% error on k-fold cross validation

Maximum:

- (Expected) that is also expresses a level of certainty and/or is interpretable
- Graphical application that wraps (Expected) to include CT scan visualizations with confidence level representations

Overview of what has been done

- Preprocessing
  - Removal of instances with corrupted metadata, crops to chest scan region, transformations to isotropic voxels, normalizations, downsampling, and padding
- Training
  - Training U-Net 3D [1] is currently running for k-fold cross validation
  - Implementation of DAF3D [2] is running concurrently (attention models suggested by Dr. David Dreizin)
  - Implementation of deep structural causal model (SCM) under development [3] (causal models suggested by Dr. Mathias Unberath)

[1] Çiçek Ö., Abdulkadir A., Lienkamp S.S., Brox T., Ronneberger O. (2016) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin S., Joskowicz L., Sabuncu M., Unal G., Wells W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science, vol 9901. Springer, Cham.

[2] Wang Y, Dou H, Hu X, Zhu L, Yang X, Xu M, Qin J, Heng PA, Wang T, Ni D. Deep attentive features for prostate segmentation in 3D transrectal ultrasound. IEEE transactions on medical imaging. 2019 Apr 25;38(12):2768-78.

[3] Pawlowski N, Castro DC, Glocker B. Deep structural causal models for tractable counterfactual inference. arXiv preprint arXiv:2006.06485. 2020 Jun 11.

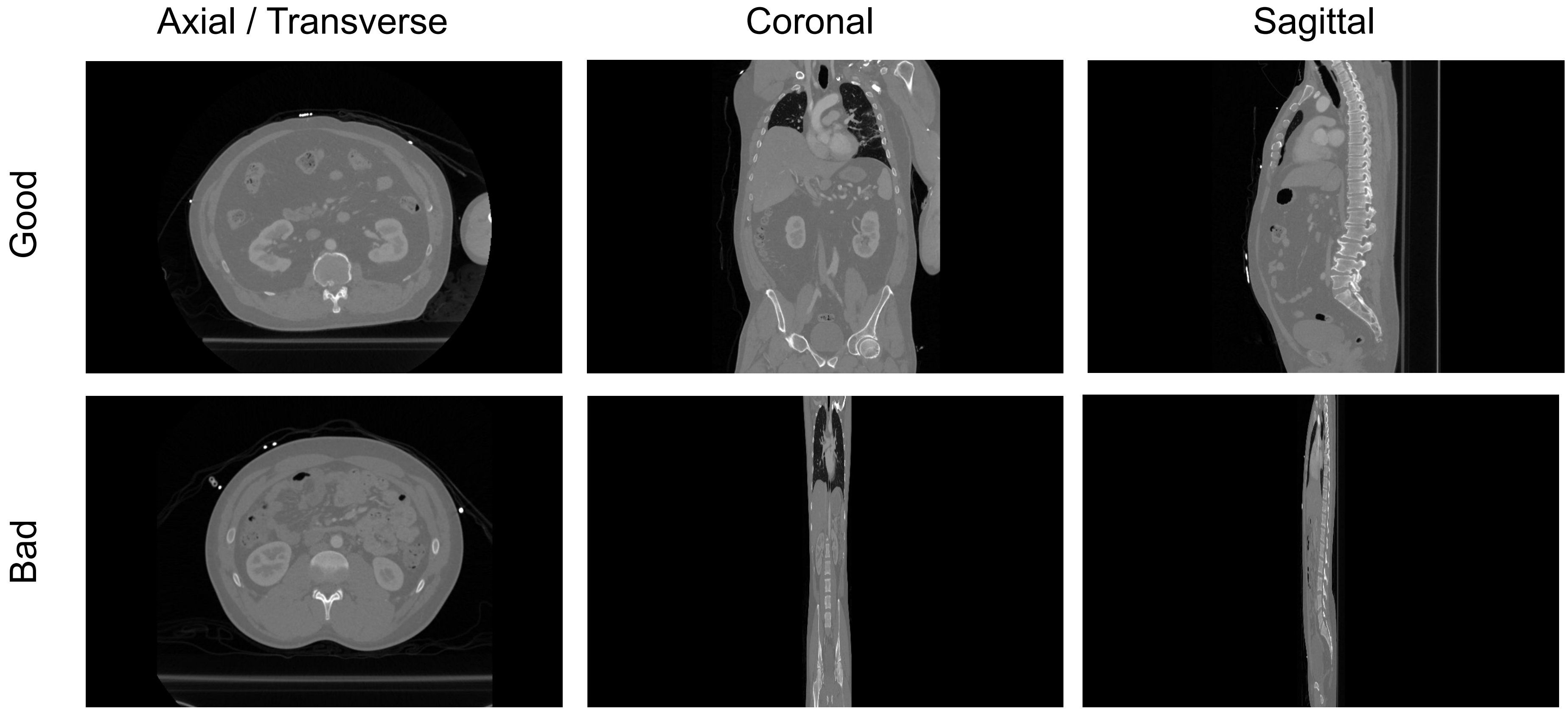
Preprocessing
&
Major problems we encounter

Major problems we encounter

Preprocessing: Metadata error, manual cropping, Down sampling, etc.

Dependency problem: CUDA and pytorch

Preprocessing: Removed instances with corrupted metadata (15/94) cases that Dr. Dreizin manually segmented



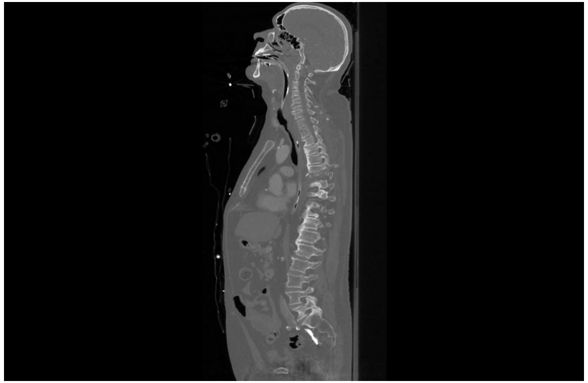
Preprocessing: Crops to chest scan region based on Dr. Dreizin's expertise

Axial / Transverse

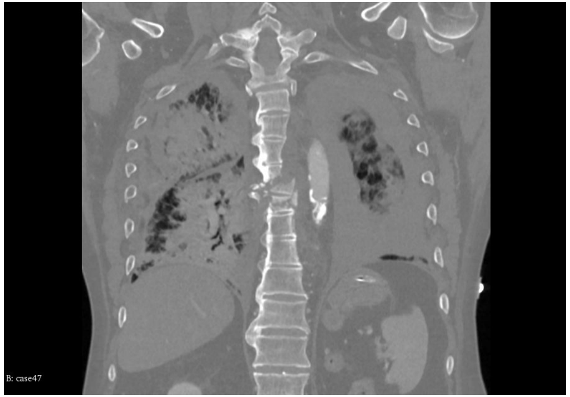
Coronal

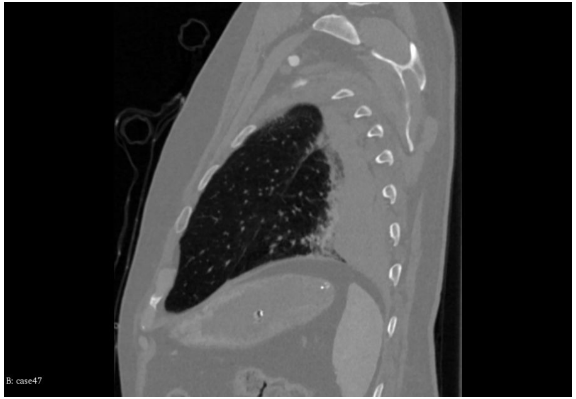
Sagittal

Original



Cropped





Preprocessing: transformations to isotropic voxels

- Transformed to 2mm cubic voxels from initially near 0.7 mm slices, downsampled from 1mm as suggested by Dr. Unberath
    - Transforming to 8mm cubic voxels for larger networks
- Needed frame transformations from NifTI space to numpy/pytorch space using meta data for quaternion transformations

Preprocessing: Normalizations

Implemented both min-max normalization and stddev standardization

(Only using min-max normalization as that is the “norm” in literature and discussing with Dr. Dreizin)

$$\frac{x - min(x)}{max(x) - min(x)} \qquad \frac{x - mean(x)}{stddev(x)}$$

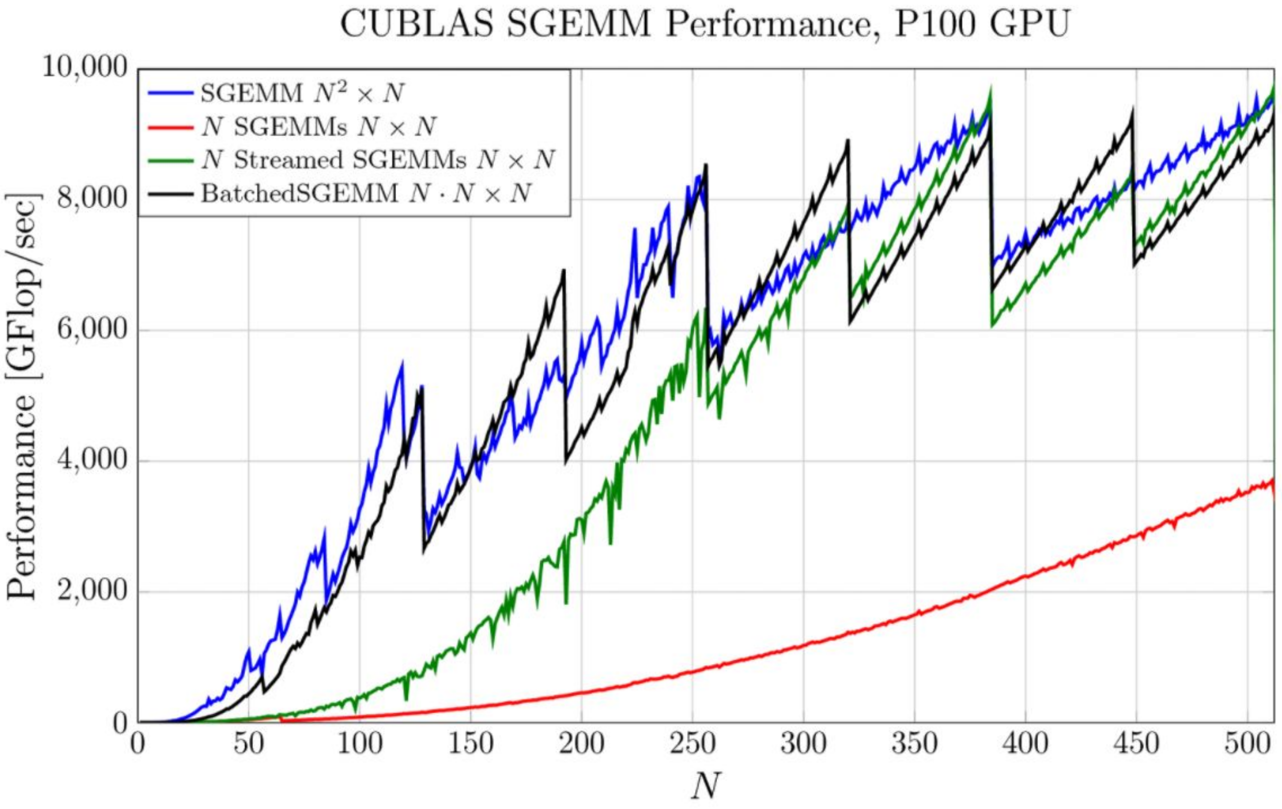
Preprocessing: Downsampling and Padding

Padded to 256x192x256 after downsampling

multiples of 64 are optimal for many CUDA functions, particularly for cuBLAS (Nvidia's BLAS impl.)

Also helpful for network up/down sampling to maintain compatible tensor shapes

(PyTorch built for CUDA)



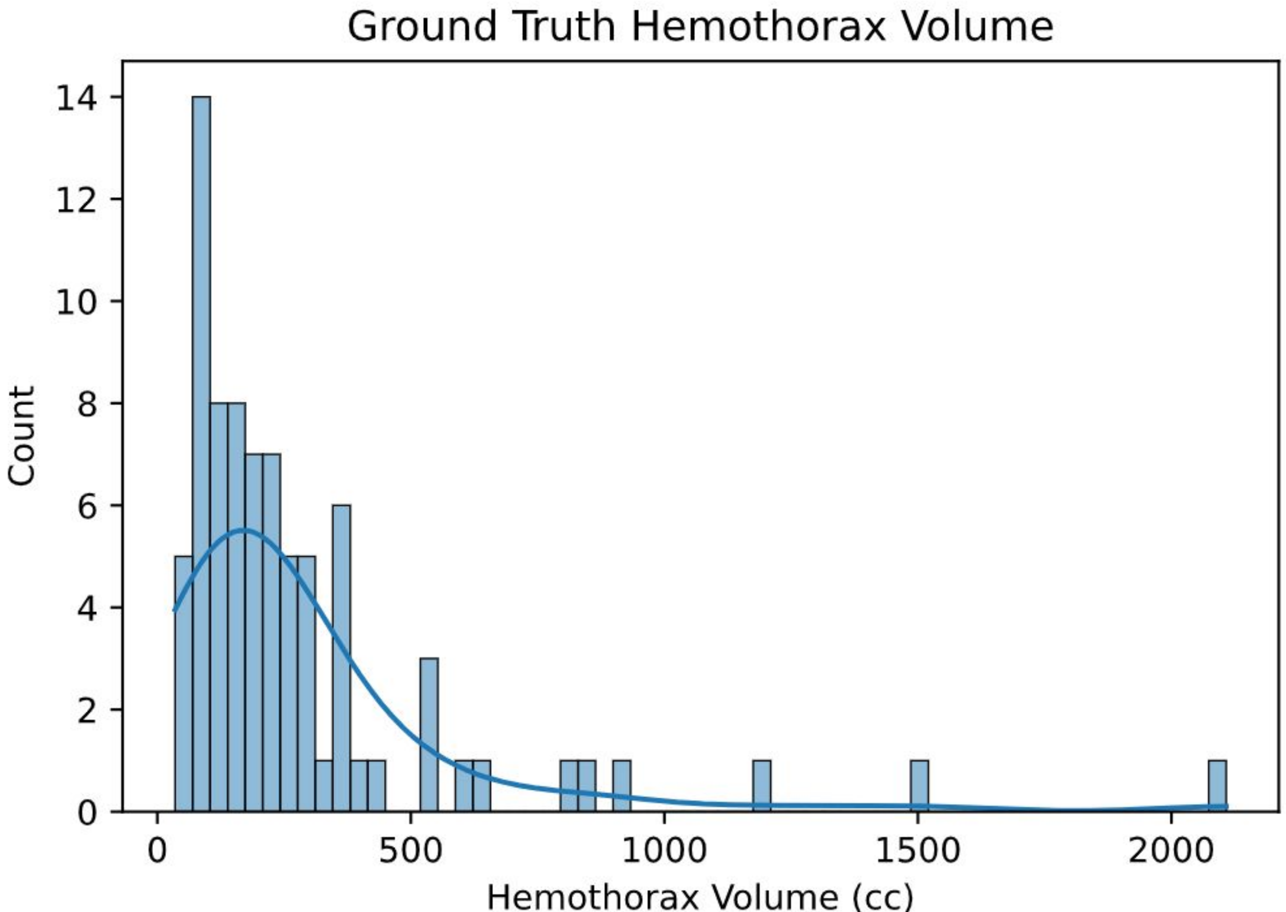
https://developer.nvidia.com/blog/cublas-strided-batched-matrix-multiply/

Dependency problem: CUDA

PyTorch binaries for CUDA compute 3.5 removed from global repo to save space

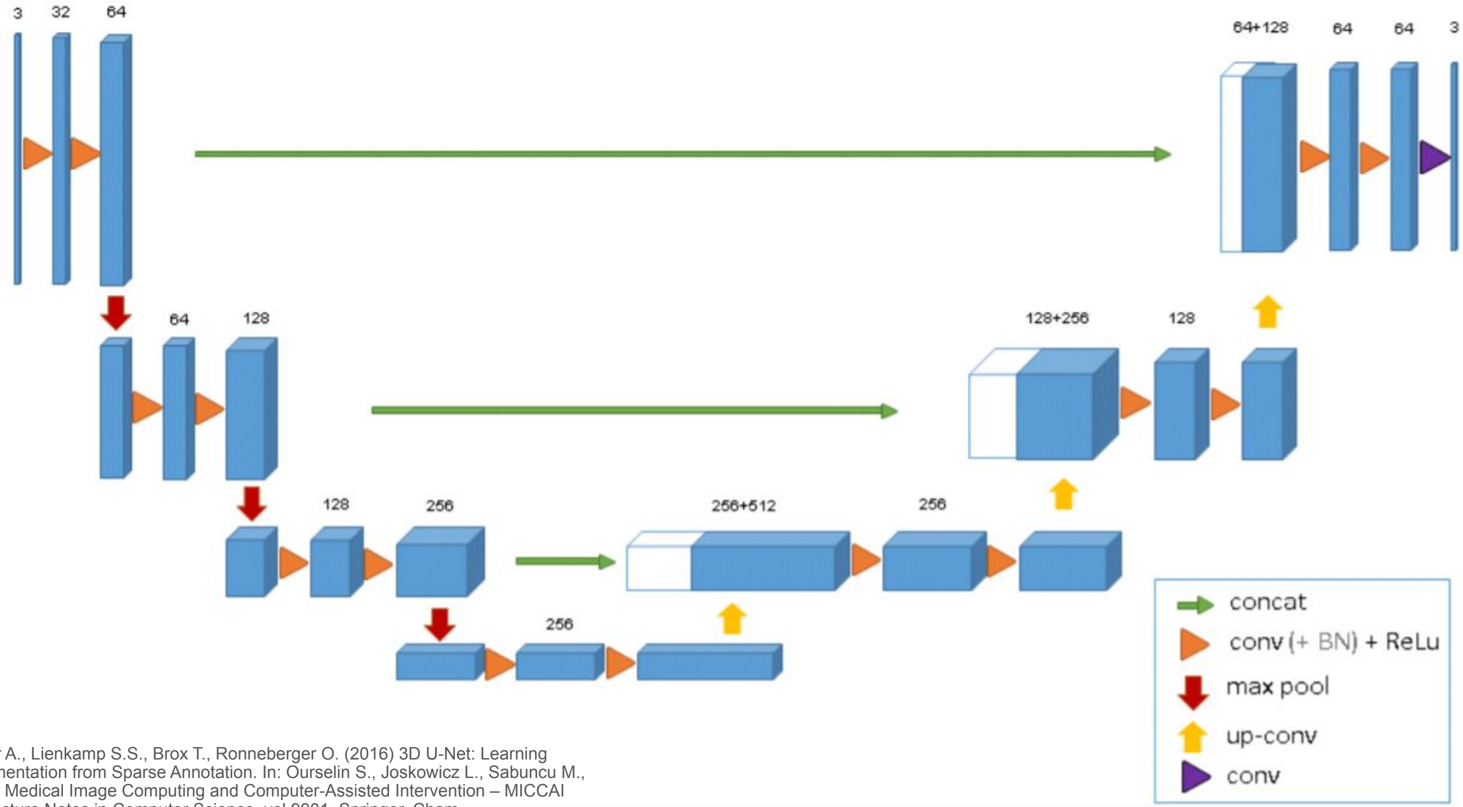
- Solution: built PyTorch from sources
  - Converted workspace from using pip to shared conda env for easier building from sources
  - In total, added about 3 days of debugging

Preprocessing: Dataset Overview



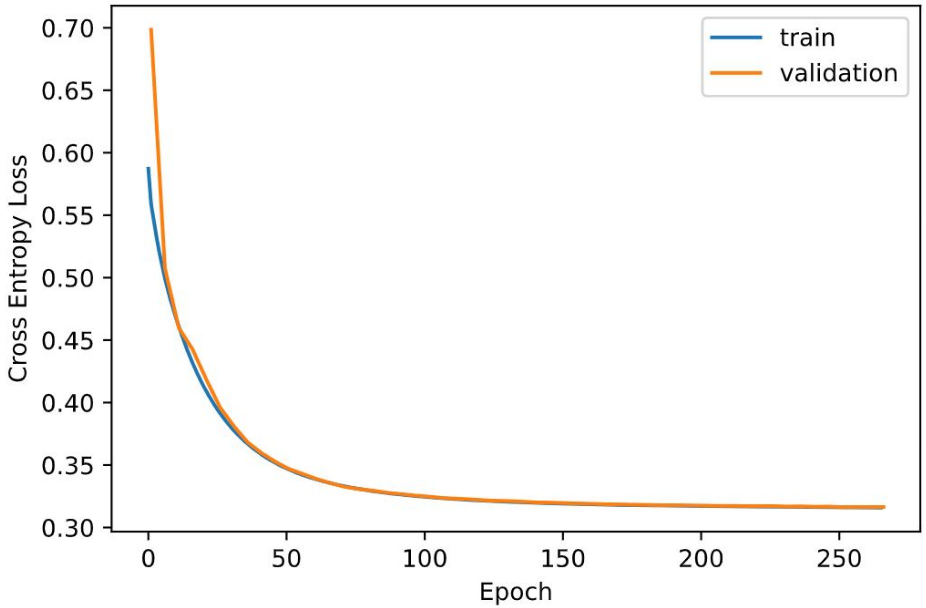
Current Benchmarking Result
Unet 3D

UNet 3D



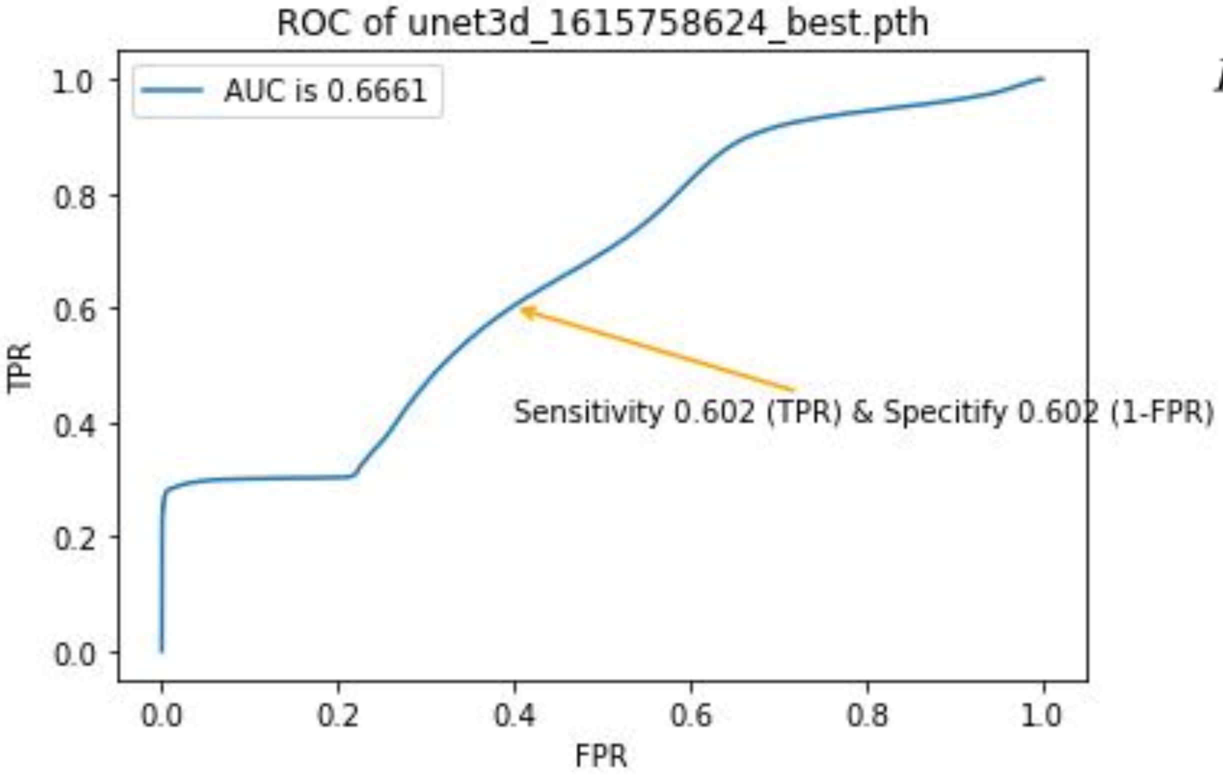
[1] Çiçek Ö., Abdulkadir A., Lienkamp S.S., Brox T., Ronneberger O. (2016) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin S., Joskowicz L., Sabuncu M., Unal G., Wells W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science, vol 9901. Springer, Cham.

Training: Current Results of 3D U-Net



Çiçek Ö., Abdulkadir A., Lienkamp S.S., Brox T., Ronneberger O. (2016) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin S., Joskowicz L., Sabuncu M., Unal G., Wells W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science, vol 9901. Springer, Cham.

Training: current metrics of 3D Unet



$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Dice: 0.6021

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Jaccard: 0.4307

ROC_AUC: 0.6661

Timeline adjustments
&
Deliverable/Dependency Status

Dependency status

Dependency	Need	Status	Followup	Contingency Plan	Planned	Hard	Check
Computing Power	train many models	Orthrus cluster	N/A	Google Cloud Credits	1/21	2/22	✓
CT Scans	train any models	Uploaded to Orthrus	Interpolating to 1mm, building 3 views	N/A	2/18	2/24	✓
CUDA **developer.nvidia.com/cuda-zone**	GPU interface	Installed	N/A	N/A	1/28	2/22	✓
PyTorch **pytorch.org**	setup environment	Built from sources	N/A	Tensorflow tensorflow.org	2/15	2/22	✓
3D Slicer **slicer.org**	visualize scans	Installed	N/A	Use python packages	2/18	2/22	✓
Open-Source Models	Benchmark and ensemble	Accessed Implementing	Run and evaluate	N/A	3/10	3/15	✓

Deliverables Status

	Activities	**Results/Deliverables**	**Due**	**Status**
Minimum	Literature survey for model selection	Draft a list of open-source models with code or architecture description	2/22	
	Preprocess CT scans (interpolate, make 3d slices)	Interpolate CT scans and convert data to PyTorch tensor type	3/1	
	Complete pipeline and I/O APIs for the project	Build a network framework consists of Python classes	3/1	
	Benchmark open-source models	Benchmark existing open-source models measured with Dice/Jaccard	3/21	**In progress**
Expected	**Research on and implement several Deep Structural Causal Models (New)**	**Several implemented SCM that outperforms the benchmark models (New)**	3/26	**In progress**
	Design and implement an ensemble algorithm	A documented program that estimates blood volume (inputs: CT axial scans; output: predicted hemothorax volume)	4/4	**In progress**
	Improve the ensemble algorithm	A documented program outperforms the benchmark (inputs: CT axial scans; output: predicted hemothorax volume)	4/11	Planned
Maximum	Incorporate certainty level into our algorithm	A documented program visualizes confidence (input: CT axial scans; output: 3D heatmaps)	4/30	Planned
	Implement a GUI-program for visualization	A documented program incorporates the framework (inputs: CT axial scans; output: 3D segmentation)	4/30	**Plan to Cancel**

Newly planned: Deep Structural Causal Models

Base on the suggestion from Dr. Mathias Unberath, we decide to add the deep structural causal models into our benchmarking models

- it will also be used as a submodel of the ensemble.
- The Deep Structural Causal Models are a new type of AI which focus on the causation instead of simply correlation.

Newly planned: Deep Structural Causal Models

ML is great at finding correlations in data, but not causation, and classic ML algorithms has a risk of

- Spurious correlations
  - Because of over-parameterization
- Over-fitting
- Lack of explainability
- Lack of generalization
  - Especially for data of different distribution

Newly planned: Deep Structural Causal Models

In our Hemorrhage prediction here, the data actually has strong causation:

- The hemorrhage should start at some points, and then spread into 3D space, constrained by many biological factors
- Thus, this process can be explained and generalized among patients
- This fact makes the SCM a good candidate to learn the causation behind pattern

Time estimates that were good

Overall, building the overarching framework and testing took as long as expected

- Some “pieces of the puzzle” are incrementally implemented
    - E.g. abstract model class implemented to parallelize development of evaluation modules before inherited models were actually implemented
- File I/O and data handling classes
    - Implemented and tested, already being used
- Evaluation module
    - Implemented and tested, already being used
    - faster than anticipated, which compensated for longer preprocessing

Time estimates that were bad

Preprocessing was more involved than anticipated

- Many procedures originally not considered (e.g. manual cropping, frame transformations)
- Problems found with first attempts (e.g. model size is too big without downsampling)

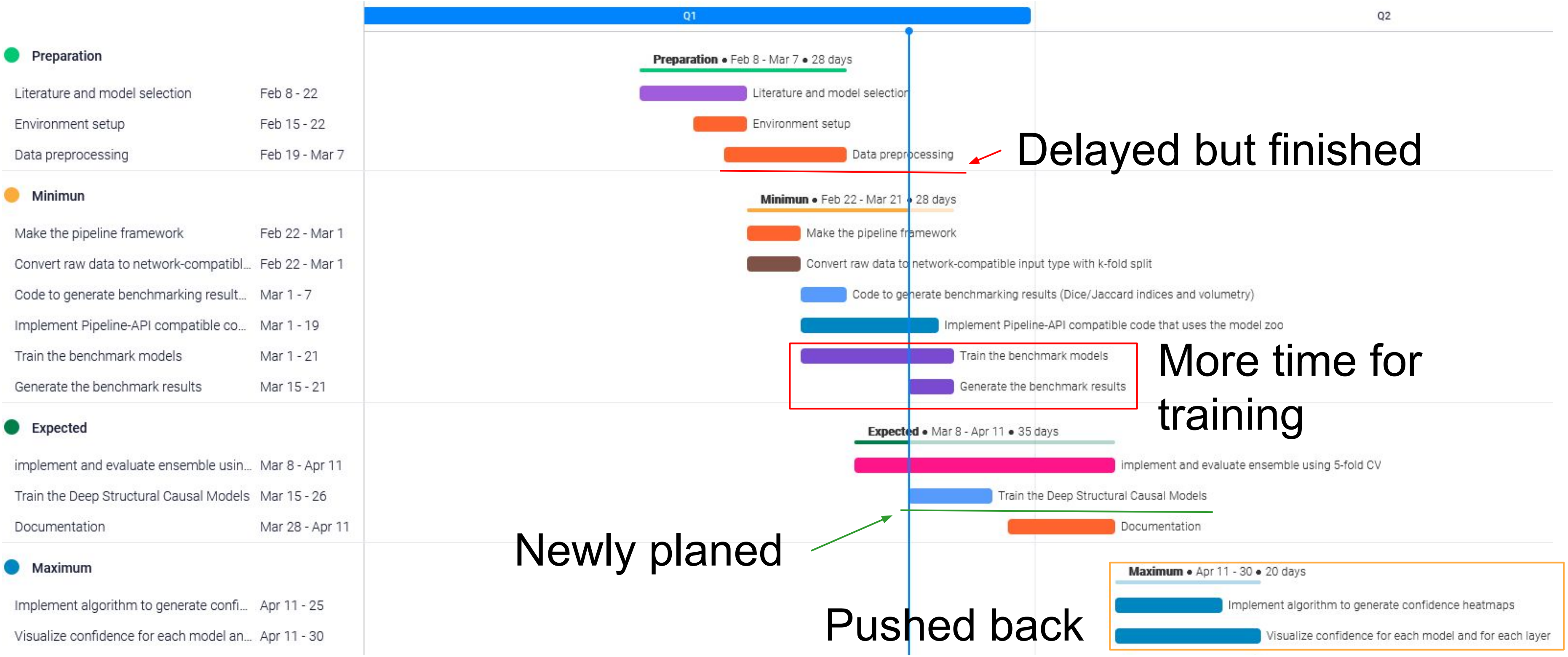
To compensate, developer time reallocated to preprocessing and getting baseline models running

Consequently, based on our adaptive software development strategy, the GUI (Maximal) needs to be sacrificed, and the original (Maximal) of having confidence expression is kept.

- **Expected:** (March 21 - April 11)
  - Build an ensemble program that works
    - ~~Mar 8 - Mar 15~~ Mar 21 - Apr 4
    - Status: Pushed back to reallocate development time to getting baseline mode
  - Finish the ensemble program for blood volume quantification
    - ~~Mar 15 - Mar 29~~ Mar Apr 4 - Apr 11
    - Status: Planned

- **Maximum:** (March 29th - April 30th)
  - An algorithm that computes/visualize certainty for segmentation
    - ~~Mar 29~~ Apr 11 - Apr 30
    - Status: Planned
  - ~~A GUI program that incorporates the framework + models~~
    - ~~Apr 12 - Apr 30~~
    - ~~Status: Planned~~

Adjustment to Timeline



Thank you for your attention!