

Segmentation Results - nnUNet and our modifications

Selina Liu

Prez Outline: Review → Action → Results

Review

- Aim
 - Lung Tumor Seg
- Method
 - nnU-Net
- Concept
 - Automatic U-Net design
 - Encoder-Decoder Topology



Action

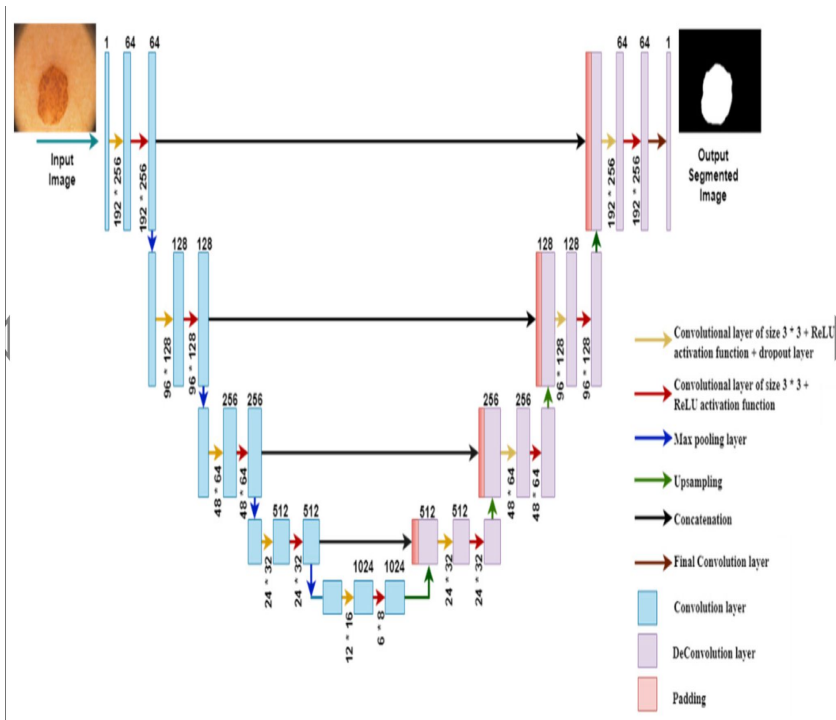
- Datasets
- Pre-Processing
- Pretrained Model
- Inference & Evaluation
- Customized Model
- Development & Inference & Evaluation



Results

- Seg Results Analysis
- Bugs & Solutions
- Formalized Workflow Notebooks

nnU-Net Review: automatic design of (encoder-decoder) U-Nets



U-Net:

$$f_{\theta}(x) = \hat{y}$$

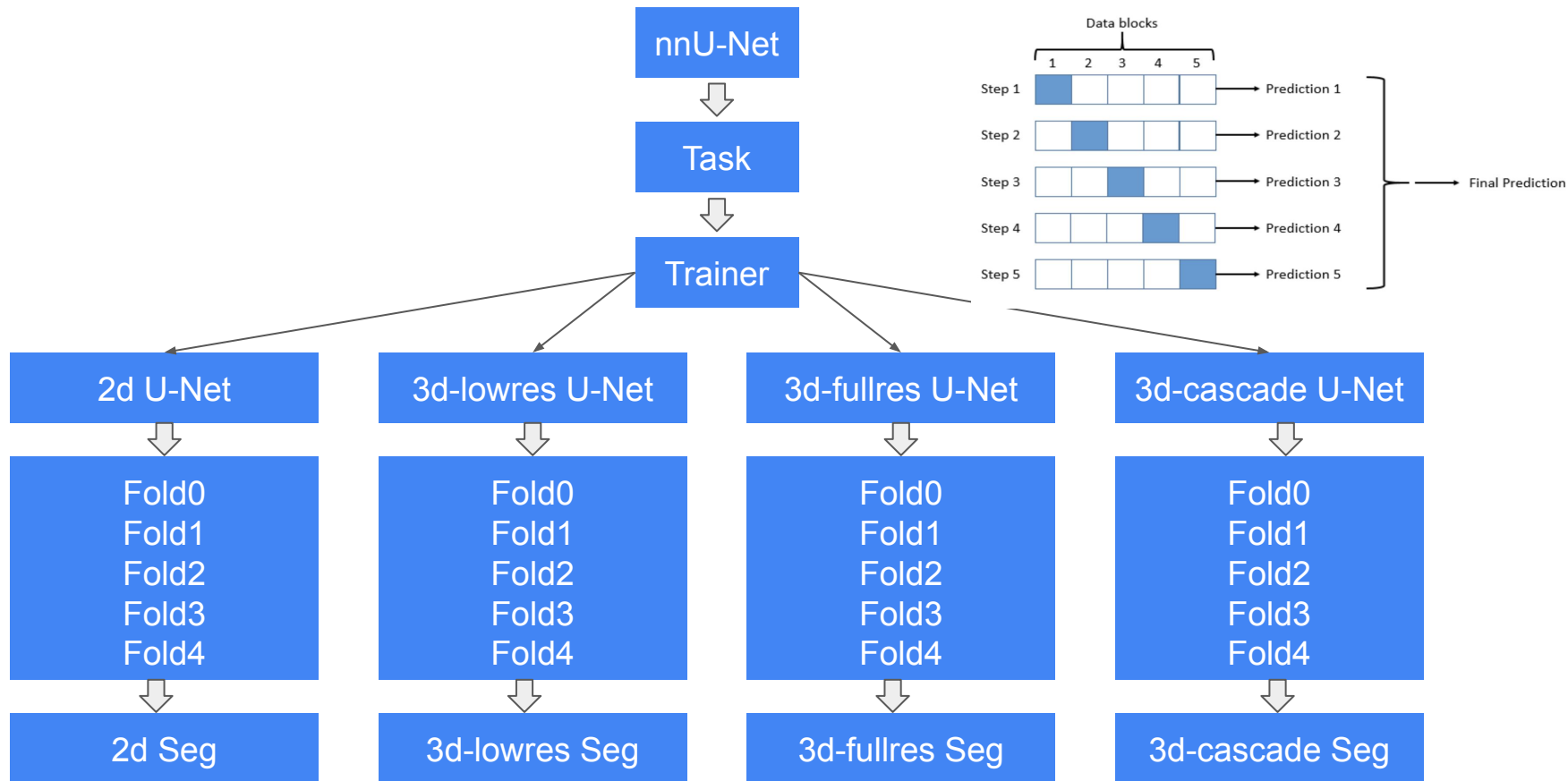
Hyperparameter (theta) tuning is a pain that we want to avoid.

nnU-Net:

$$g(X, Y) = \theta$$

A good start, but there is still room for further improvement!

nnU-Net Review: 4 Configurations & 5-Fold Structures



Action 1 - Datasets Investigation and Pre-Processing

Dataset 102

- **Train & Performance Test**
- NSCLC-Radiomics
- 414 3D CT Scans, 4 Masks
- [Nature Communications](#)
- Published 2014
- 100K Access & 2847 citation

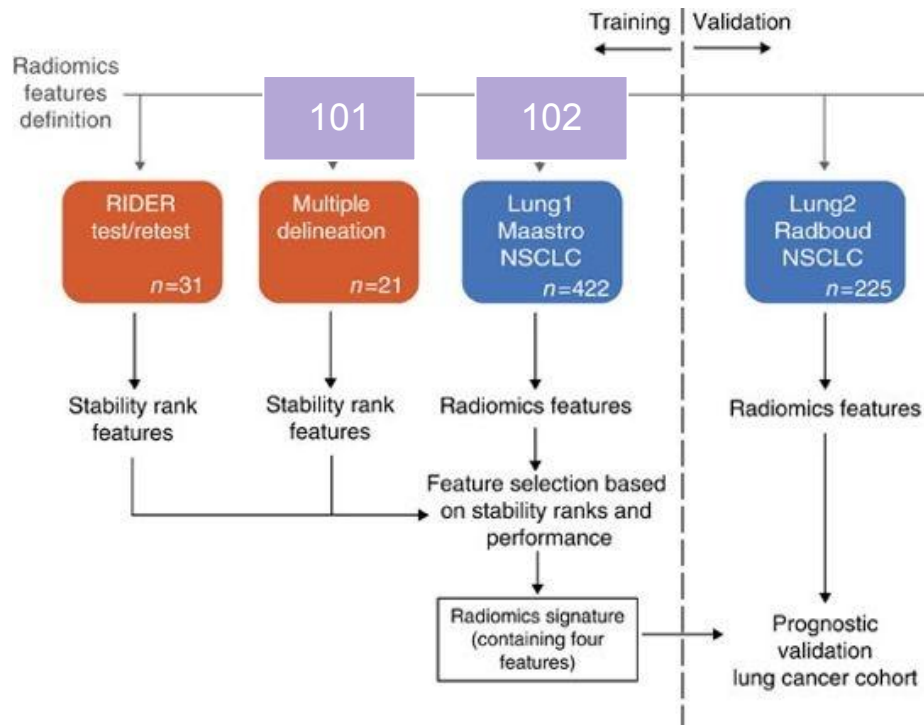
Dataset 101

- **Stability Test**
- NSCLC-Radiomics
- 20 3D CT Scans, 10 Masks
- [Nature Communications](#)
- Published 2014
- 100K Access & 2847 citation

Dataset 006

- **Performance Sanity Test**
- Medical Segmentation Decathlon
- 63 3D CT Scans, Single Mask
- [Nature Communications](#)
- Published 2022
- 30K Access & 93 citation

Dataset 101 & 102 - NSCLC-Radiomics: Closer Look



DataSet 101 → Stability Test

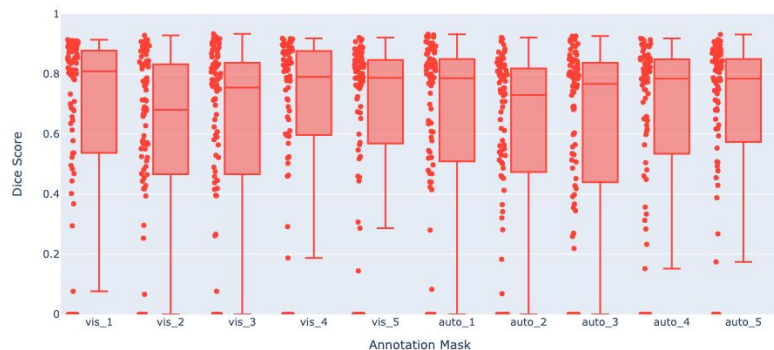
- 10 Masks, using all of them
 - GTV-1 auto1~5 are automatic seg corrected manually
 - GTV-1vis1~5 manually annotated ground-truth

Dataset 102 → Performance Test

- 4 Masks, only using GTV Mask
- Lung2 Test Set Unavailable
- 80% Train & 20% Validation

Action 2 - Pretrained Model Inference & Evaluation

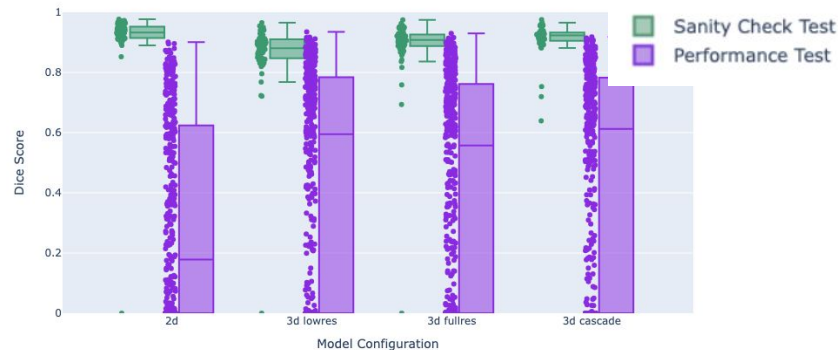
Pretrained Model 006 Stability Test on Dataset 101



Stability Test

- **CONSISTENT CHECK**
- **Only Care About Consistency**
- Relatively consistent performance across all 10 ground-truth segs
- Pretrained Model 006 is Stable

Pretrained Model 006 Performance Test on Dataset 006 [Sanity Check Test] and 102 [Performance Test]



Performance Test

- **Closer to Top, the Better**
- **Only Care About Accuracy [Dice Score]**
- Bad Performances on **Our Dataset of Interest**
- Pretrained Model 006 is Bad [for our purpose]

Action 3 - DataSet Preprocessing Bug Fix & Naive Models

Model 006

Developer:

- nnUNet Group

Train Set:

- Dataset 006

Stability Check:

- PASS

Features:

- NONE



Multi-Mask Debug

Shape Mismatch &
random errors

→

Crack nnUNet Code
[~ 1 week]

→

Our Dataset have
Multi-Mask

→

Pre-Processing to
fix multi-mask
[~ 3 days]



Model 102

Developer:

- UCSF RBVI

Train Set:

- Dataset 102

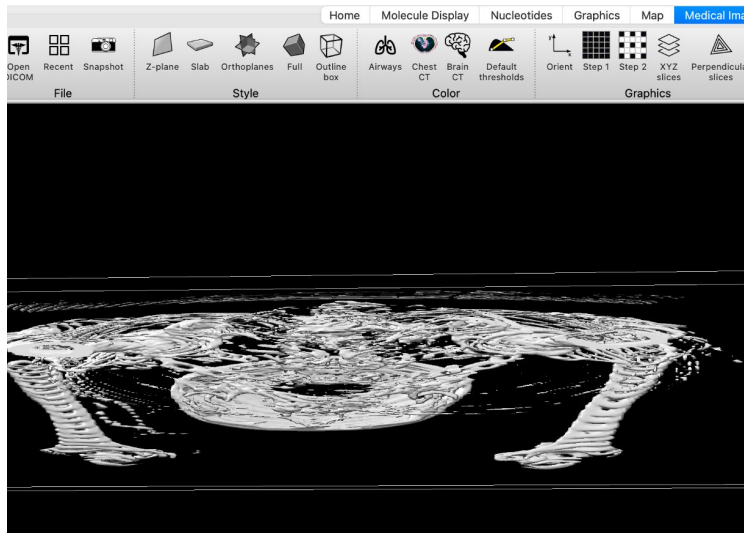
Stability Check:

- PASS

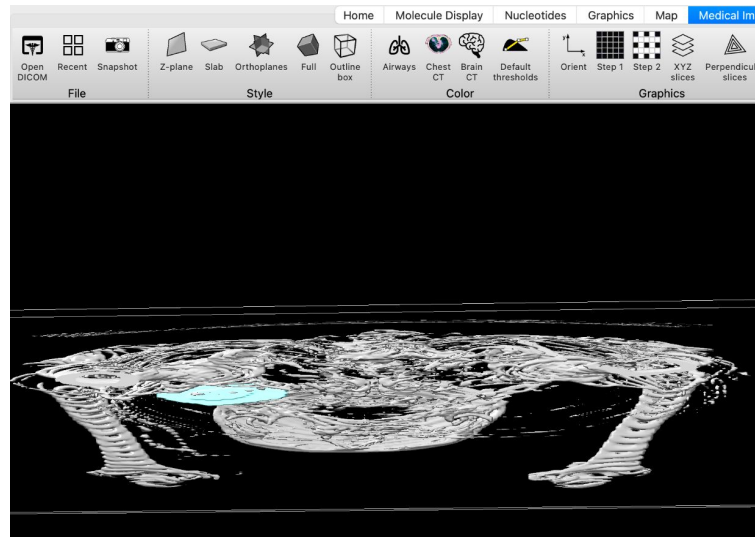
Features:

- Standard
nnUNet
pipeline

Where Are We Doing Good? In-Depth Case-Study



Pretrained Model 006

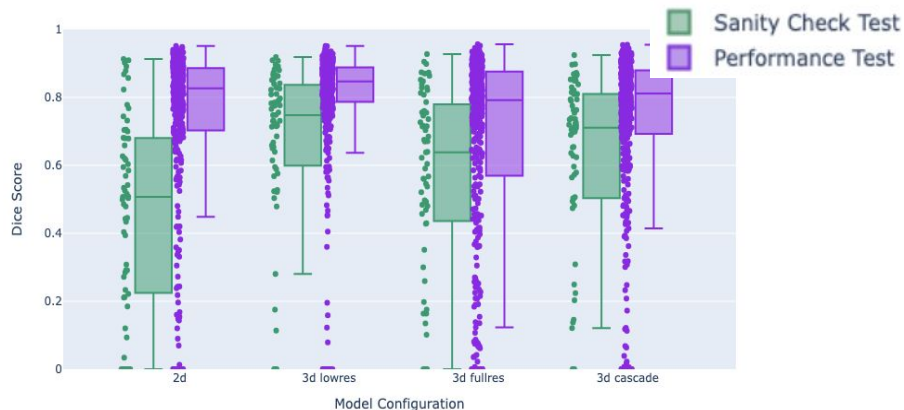


Our Model 102

Our Model 102 successfully detect tumor that pretrained model 006 missing with ~90% accuracy

Transfer Learning - Frozen Layers & Customized Initial Parameters

Model 102 Performance Test on Dataset 006 [Sanity Check Test] and 102 [Performance Test]



Key Observation:

- Good Performance on Dataset 102 [Test Set]
- Not good for Dataset 006 [Sanity Check]

Propose:

- Transfer Learning from pretrained Model 006

Concept:

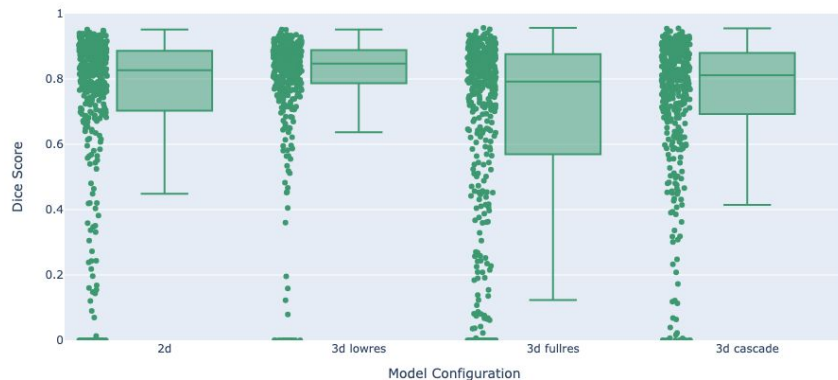
- Inherits the pre-learned features from Model 006, and then learns additional, task-specific features from dataset 102.
- Froze initial layers of Model 006, and only adjusting weights of later layers based on Dataset 102

Ideal Improvement:

- Good Performance on Dataset 102 [Test Set]
 - Better Performance on Dataset 006 [Sanity Check]
- More comprehensive Seg Model

Manual Splits - Customized Train-Test Splits

Model 102 Four Model Configurations All Have Difficult Cases



- **Accuracy [Higher, the Better]**

Key Observation:

- Our Model 102 struggles for certain “difficult cases”

Propose:

- Manual Split the train and test

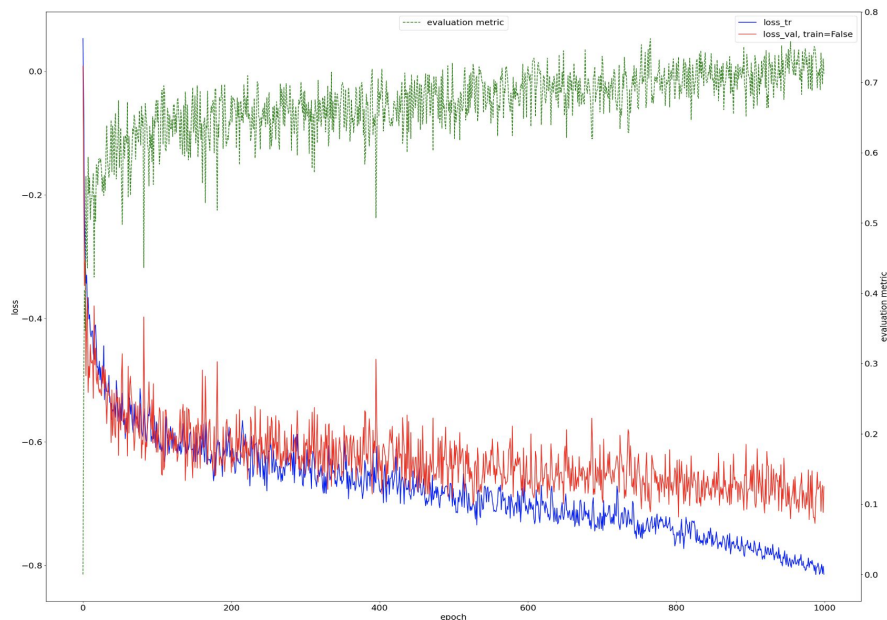
Concept:

- Define “difficult cases” to be cases with Dice Score < 0.1
- Manually force these “difficult cases” into the training set for our model to learn

Ideal Improvement:

- Our model gains more predictive power towards these ‘difficult cases’
→ More comprehensive Seg Model

Max Epoch - Increase From default 1000 to 2000



Key Observation:

- The green Dice Scores are still increasing at epoch 1000

Propose:

- Increase Max Epoch to 2000

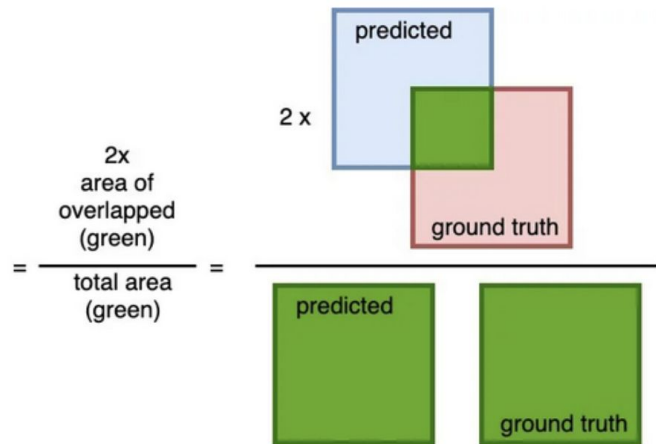
Concept:

- By default, the training are stopped at epoch 1000
- If we allow our model to learn information up to epoch 2000, it could learn more useful information

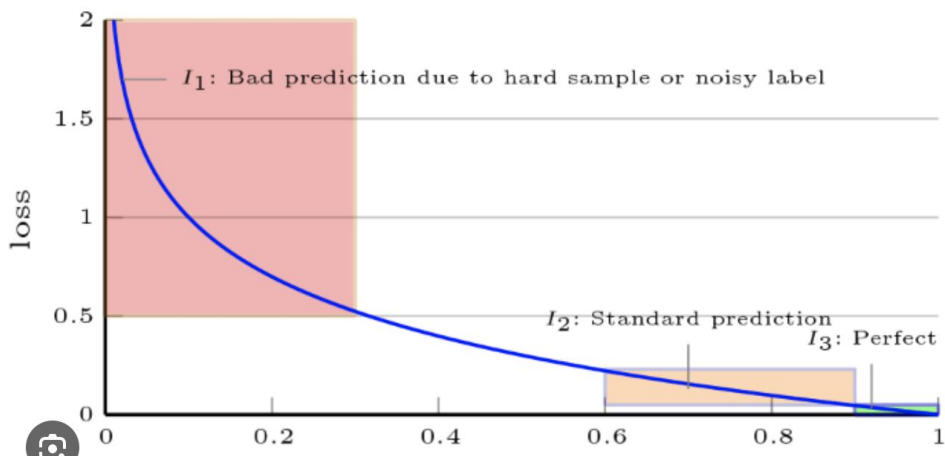
Ideal Improvement:

- Our model gains better performance on average
→ More comprehensive Seg Model

Loss Function - Cross Entropy Loss & Dice loss



- The overlap between predicted seg and ground truth
- Only care about “**Correctness**” of prediction
- Model 102 Version 2



- Dissimilarity between the true label distribution and the predicted probabilities.
- care about “**Correctness**” & “**Confidence**”
- Model 102 Version 3

Action 3 - Customized Models Features Summary

Model 102 V1

Naive Model

- 20 folds

Minsky [3 days / fold]

Train Set:

- Dataset 102

Stability Check:

- PASS

Features:

- Standard nnUNet pipeline

Model 102 V2

Fine-Tuned Model

- 20 folds

Wynton [2 ws+ / fold]

Train Set:

- Dataset 102

Stability Check:

- PASS

Features:

- Transfer Learning
- Manual Splits
- Max epoch 2000

Model 102 V3

Fine-Tuned Model

- 20 folds

Wynton [2 ws+ / fold]

Train Set:

- Dataset 102

Stability Check:

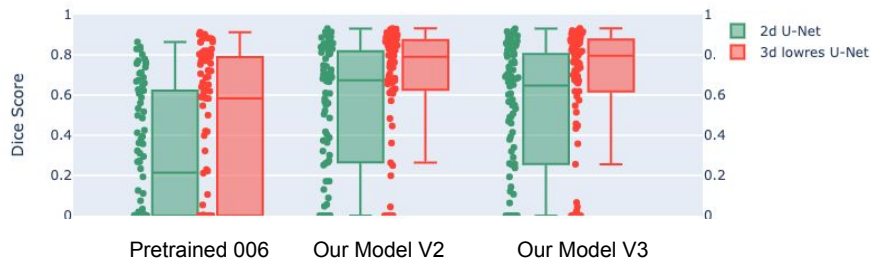
- PASS

Features:

- Transfer Learning
- Customized Loss function
- Manual Splits
- Max epoch 2000

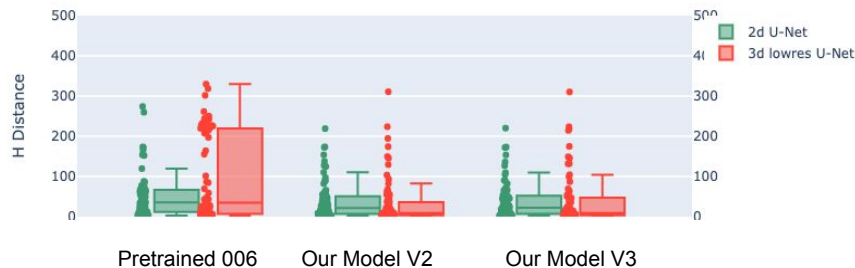
Result 1 - Our Models [Half-Trained] are LOT better

Cross Comparison of Model Performance [M 104 & 105 only half trained]



- Dice Score [Overlap between prediction and ground-truth in xy plane]
- **Higher the Better**
- Pretrained M006 \rightarrow 0.583
- Our M V2 \rightarrow 0.790
- Our M V3 \rightarrow 0.796

Cross Comparison of Model Performance [M 104 & 105 only half trained]



- H Distance [Distance between prediction and ground-truth in z-axis]
- **Lower the Better**
- Pretrained M006 \rightarrow 34.8
- Our M V2 \rightarrow 9.24
- Our M V3 \rightarrow 8.86

Result 2 - Bugs, Debugging Notebooks, Solutions & Timeline

Major Bug	Debugging	Solutions	Timeline
nnU-Net env Pytorch Compile	Github Issues	nnU-Net Guide	3 days
Shape Mismatch Multi-Masks Dataset	Crack nnU-Net code Pre-processing Fix	Multi-Mask Pre-processing	2 weeks
Customized Trainer Not recognized	Crack nnU-Net code Trainer set up	nnU-Net Trainer	2 weeks
Minor Bugs	Github Issues	Debug Notes	1 weeks

Result 3 - Formaralized nnU-Net Workflow Notebooks

- [nnU-Net Virtual Environment Setup](#)
- [Step A - Data Format Convert, \(Multi-Mask Segmentation\) DICOM to Nifti](#)
- [Step B - Pretrained Model Inference and \(Multi-Mask Shape Mismatch\) Evaluation](#)
- [Step C - Training Standard nnUNet Model, Inference, and Evaluation](#)
- [Step D - Transfer Learning With nnUNet Variants & Manual Split & Max Epoch](#)

- [Debug - nnUNet Debug Notes.ipynb](#)

nnU-Net Summary:

Actions

- Investigated NSCLC-Radiomics Dataset & Fix Multi-Mask 'Bug'
- Evaluated Performance of Pre-Trained Model 006 → Not Good Enough
- Customized Our nnU-Net Version 1 ~ 3

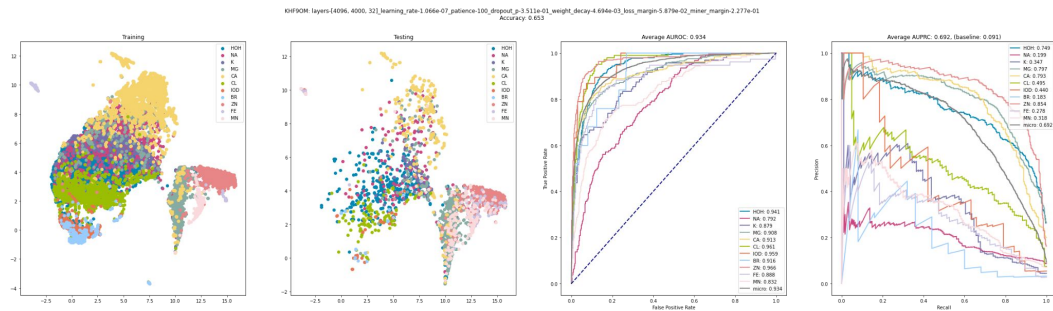
Results

- Our Models [Half Trained] have better performance [Dice ~ 0.8]
- Collected Major Bugs & Annotated Solutions
- Formalized nnU-Net Workflow for our users

Additional Projects: CryoEM MIC & Universal Model

CryoEM MIC:

- Predict the identity of unknown densities in CryoEM images
- Major Models available on github [Not Public Yet]



Universal Model:

- State of Art Machine Learning Algorithm to segment
- Recently Accepted by [ICCV](#)

Next: Deploy nnU-Net, MIC, Universal Model on ChimeraX

- nnU-Net Wrap Up
 - Wait for our models to be fully trained
 - Inference and Evaluation on remaining Model Configuration [3d_fullres & 3d_cascade]
 - Pick the BEST Model [Version X + Model Config X] to deploy
- Deploy nnU-Net on
- UCSF Private Data Validation
 - For gpu users: deploy on AWS / other web services ...
 - For cpu users: enable nnU-Net cpu prediction
 - Website including Tutorials, Examples and FAQ ...
- Deploy CryoEM MIC
 - Setup Command Line prediction ...
 - Deploy as a bundle
- Deploy Universal Model
 - Test on UCSF Private DICOM
 - nnU-Net Similar Deployment

THANK YOU