A Collaborative Approach To AI Research: *Designing AI for Hepatocellular Carcinoma*

A presentation by Dr. Sunil Jeph and OpenVessel LLC

Dr. Sunil Jeph

Director of Computer Vision Research, CRRG

Assistant Professor of Radiology
Penn State College of Medicine

OpenVessel Presenters

Leslie Wubbel
CEO & Founder

Nathaniel Leies
Business Dev & Marketing

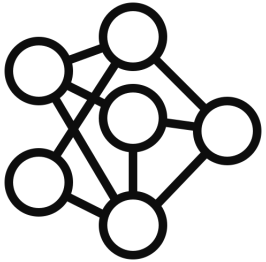
Alex Schweizer
Product Developer

OpenVessel

OpenVessel LLC is an AI research consultant.

We provide ***data science consultation and solutions to academic physicians*** to power their AI research.

OpenVessel | What We Do



Collaboratively ***engineer AI / ML solutions*** with investigators.

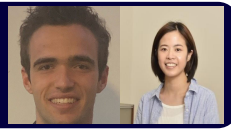
Collaboratively ***design experiments***, and oversee the entire experiment process.

Co-author impactful research and successful funding applications.

Establish a foundation of data-science research upon which ***our clients can build their success.***

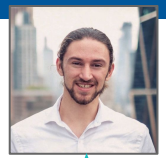
OpenVessel | Our Team

Interns
Programming



Architect

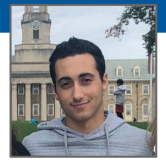
Business Development and Marketing



Intern
Graphic Designer

Dataset

Code Analysis



Performance

Interns
Data Science

Working with Penn State College of Medicine

December 2020

Clinical Radiology Research Group

Contacted CRRG | Offered pro-bono Medical AI project

Promise: Recruit and train PSU undergraduate interns

January 2021

Connected with Dr. Jeph | Identified clinical problem

January – February

Developed **OpenVessel Deep Learning AI Model**

Collaborated with Dr. Jeph: Biweekly meetings | Phone calls | Slack

March 2021

Demonstrate Deep Learning AI Model | Develop abstracts

Our Project Goal with Dr. Jeph

Engineer a model that can ***detect and classify*** hepatocellular carcinoma (HCC).

SCOPE OF WORK

- Train model with open-access liver CTs
- Record results
- Create abstracts
- Utilize model in publications, promoting PSU College of Medicine

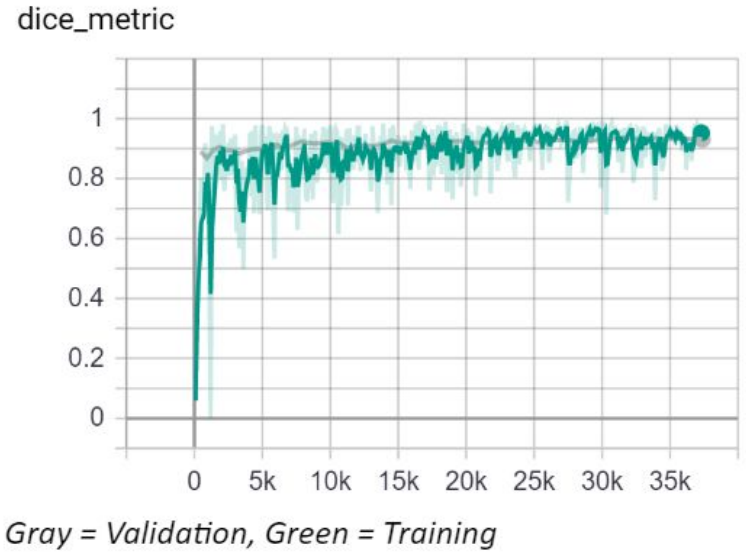
OpenVessel

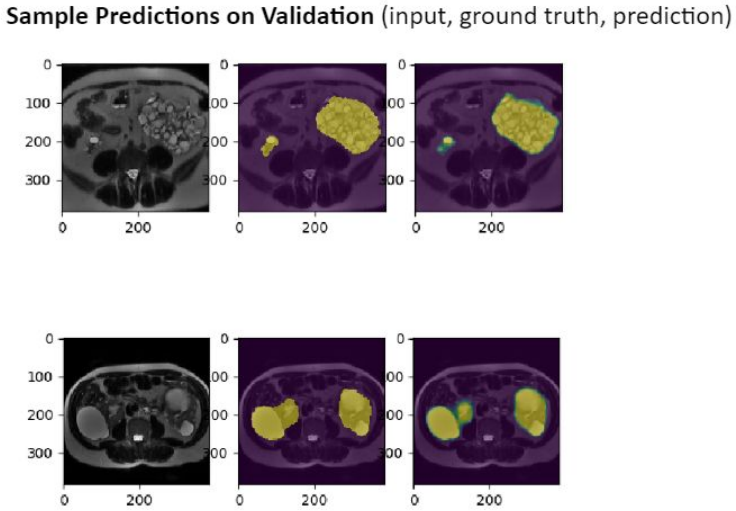
Timelines & Deliverables

Dr. Jeph's Previous AI Research

Automatic Segmentation of Autosomal Dominant Polycystic Kidney Disease Using Deep Learning with Multi-institutional Validation

Akshay Goel MD, Sunil Jeph MD, Sadjad Riyahi MD, Martin Prince MD



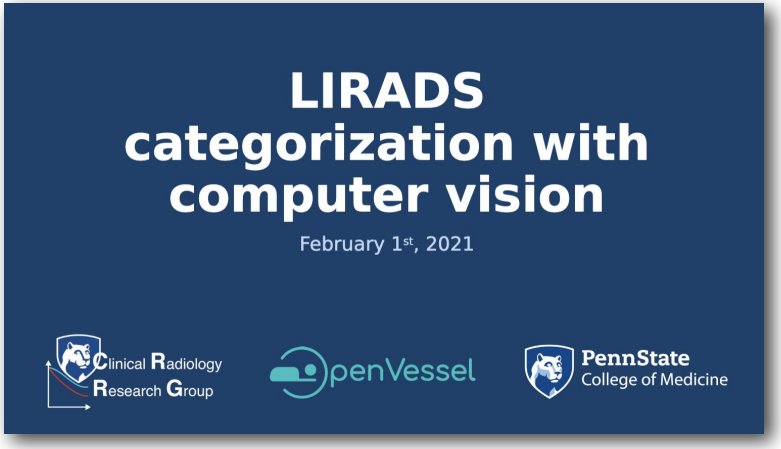


Clinical Problem

Dr. Jeph requested that we address **liver research.**

“Primary liver cancer is the second most common cause of cancer death” [1].

Opportunity to introduce **multiphase CT** data from Penn State to the OpenVessel model.



Above: Dr. Jeph introduced OpenVessel team to LIRADS classification and the multiphase CT modality in a collaborative meeting.

OpenVessel AI / ML Consultation

Client Ask

Create a data model that will continue Dr. Jeph's research on segmentation/data modeling. Interested in AI for the liver.

What OpenVessel Offered

Building a deep learning model to segment and classify liver cancer. Access to openly available liver CTs and AI codebases provided a starting point.

Liver Tumor Segmentation (LiTS) Dataset

We used the LiTS Dataset, available through the LiTS Challenge, an AI competition launched in 2017.

SPECIFICATIONS

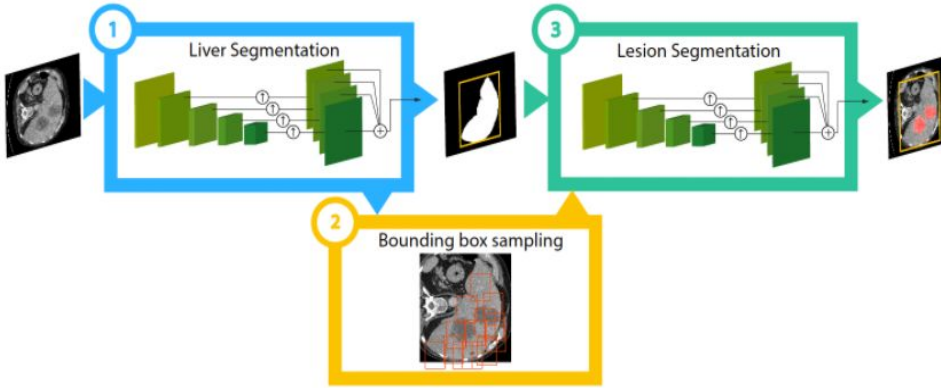
- 130 CT Scans
- 57.6 GB
- Contains pixel volumes and labels
- Variance in contrast levels

ABNORMALITIES

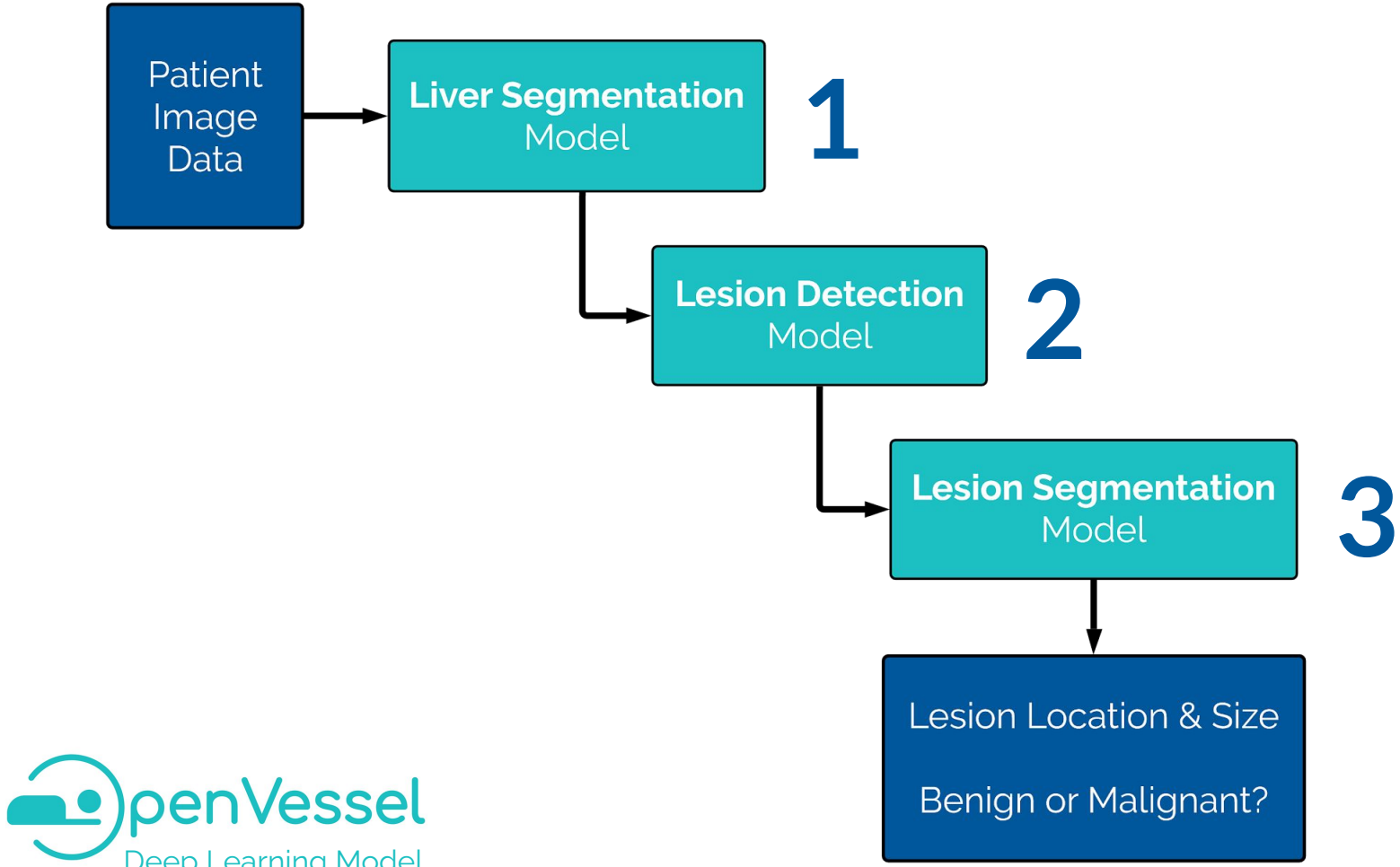
- Primary & secondary cancers
- Variance in lesion shapes and sizes
- Variance in number of lesions per study

OpenVessel Deep Learning Model

- Started with Barcelona Tech Deep Learning Model (2017), an open source code base
- We validated the model by rebuilding it
- Automated manual processes
- Debugged extensive problems
- Arrived at OpenVessel Deep Learning Model (2021)



Above: Barcelona Tech Model

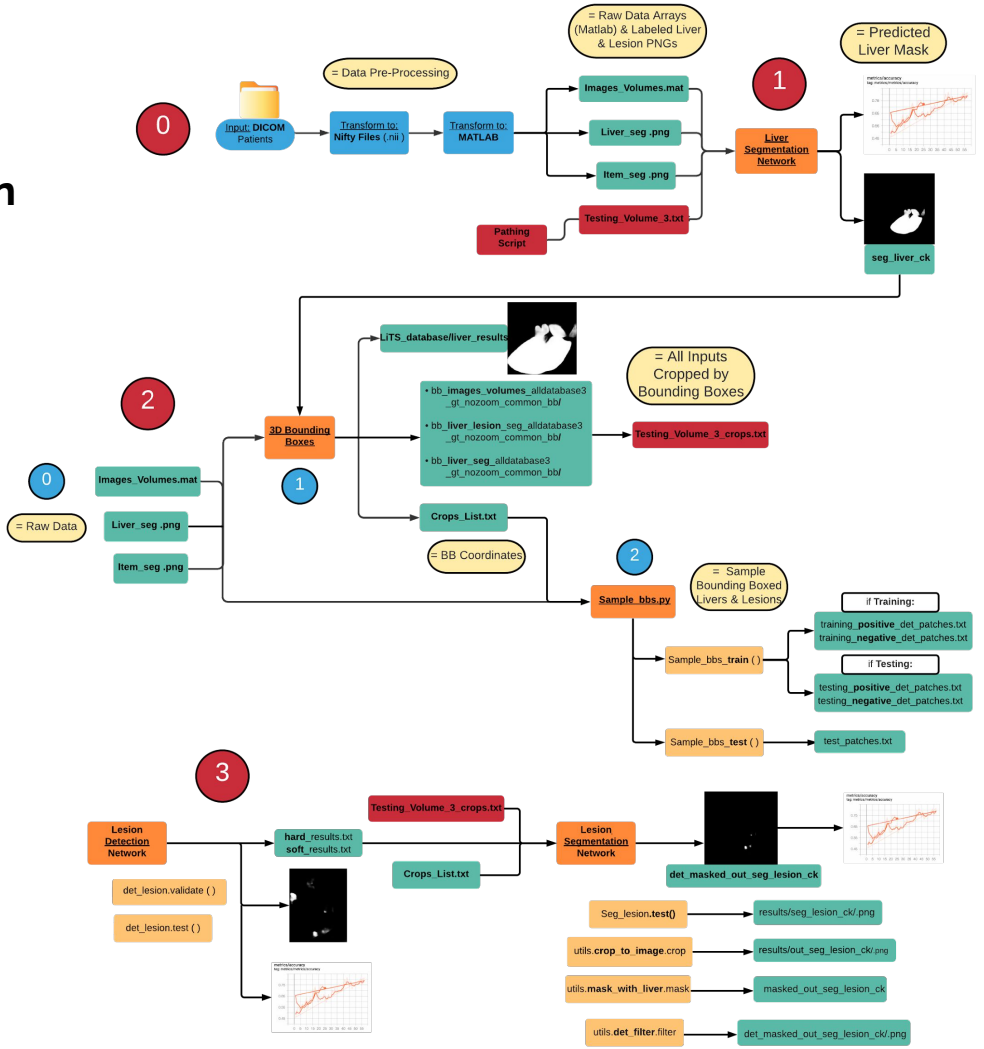


OpenVessel
Deep Learning Model

1 Preprocessing and Liver Segmentation

2 Lesion Detection

3 Lesion Segmentation



Work Timeline | 2 Month Progress

GitHub

[OpenVessel Github Repository]

Estimated Total Hours
1,680

Number of FTEs
6

Average employee contribution per week
35 hours

OpenVessel

Bringing AI closer to healthcare.

https://www.dev.openvessel.org/ openvessel@openvessel.org

Repositories 9 | Packages | People 14 | Teams 1 | Projects | Settings

Find a repository... | Type: All | Language: All | Customize pins | New

DataScienceDeck

Private

Python 0 2 1 0 Updated 3 hours ago

liverseg-2017-nipsws

Forked from imatge-upc/liverseg-2017-nipsws

Detection-aided Liver Lesion Segmentation

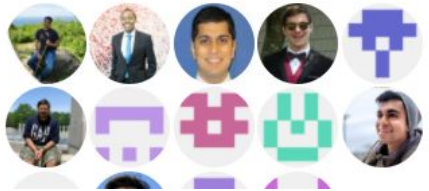
Python MIT 47 1 4 0 Updated 3 hours ago

Top languages

Python JavaScript HTML

People

14 >

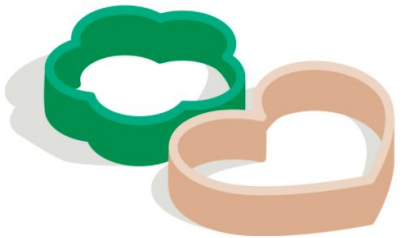


TERMINAL DEBUG CONSOLE PROBLEMS 8 OUTPUT

```
(MyVenv) PS D:\L_pipe\greg_branch\liverseg-2017-nipsws> python .\main.py
Running step: seg_liver_test

Started loading files...
Done initializing Dataset
```

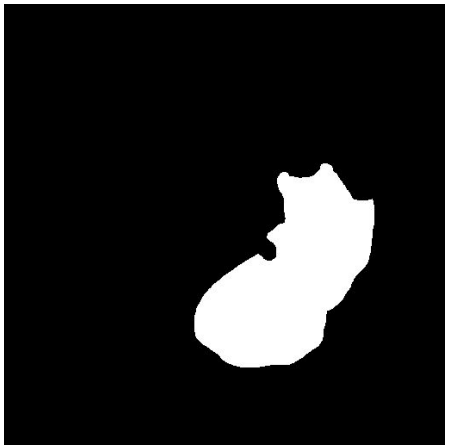
RESULTS

Terms to Know: Segmentation & Mask



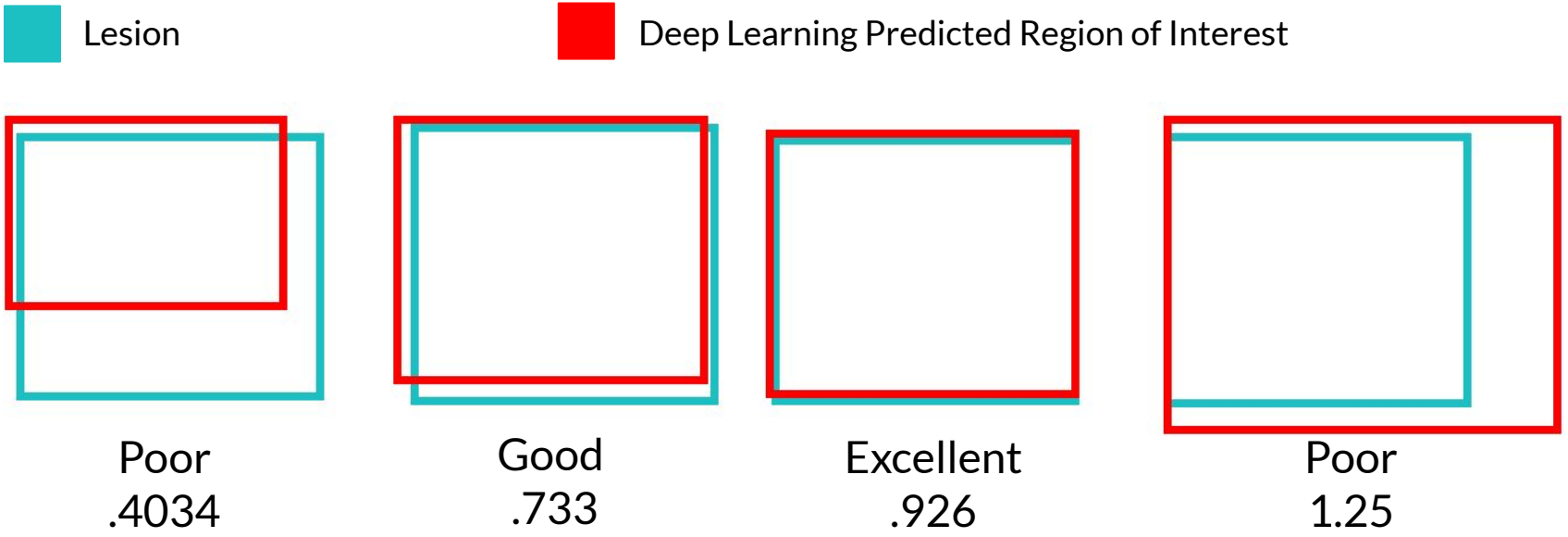
Segmentation of a Mask-

- Divides images into ***parts or sections***
- Generates a **mask** that ***outlines an object***
- By analyzing the HU values



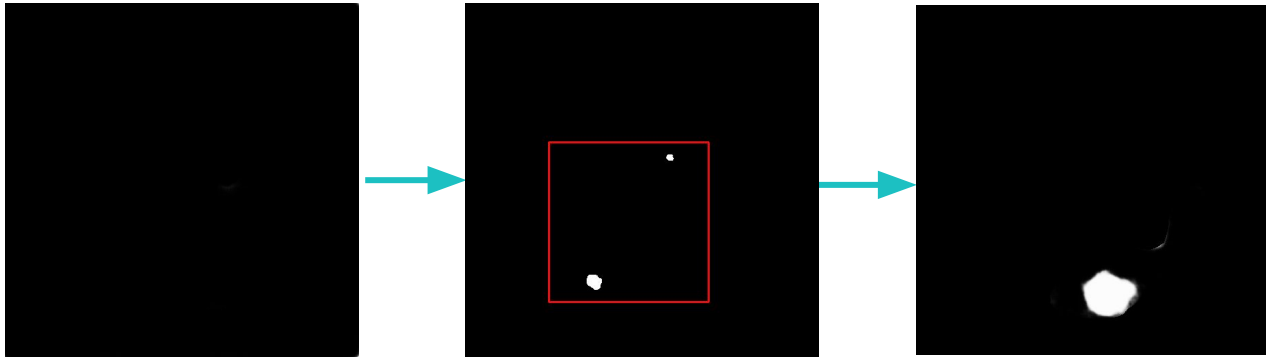
Understanding Evaluation Metrics | DICE Score

DICE is a metric describing how well a model identifies a lesion. Computed through overlap.



OpenVessel
Deep Learning Model

VISUAL OUTPUTS

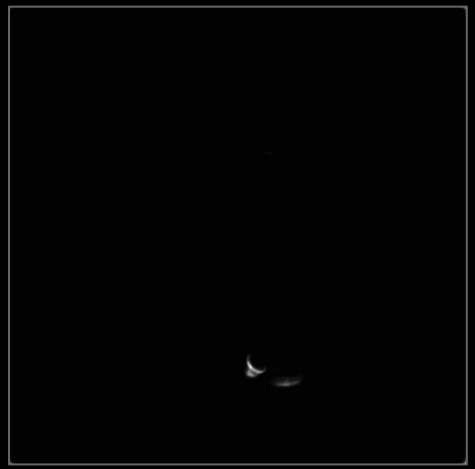


Predicted Liver Mask | Detected Lesions | Predicted Lesion Mask

	Technical Description	DICE SCORE	Time
Liver Segmentation	Liver mask predicted by the model & segmented out of the region of interest (ROI) in the patient.	0.8233	4.11 m
Lesion Detection	Network classifies areas of interest in liver mask as positive (containing lesions) or negative.	0.68	1.90 m
Lesion Segmentation	The positive lesions are segmented out as results.	0.71	4.36 m

1 *Visual Results*
Liver Segmentation Network

Normal Patient CT Scan



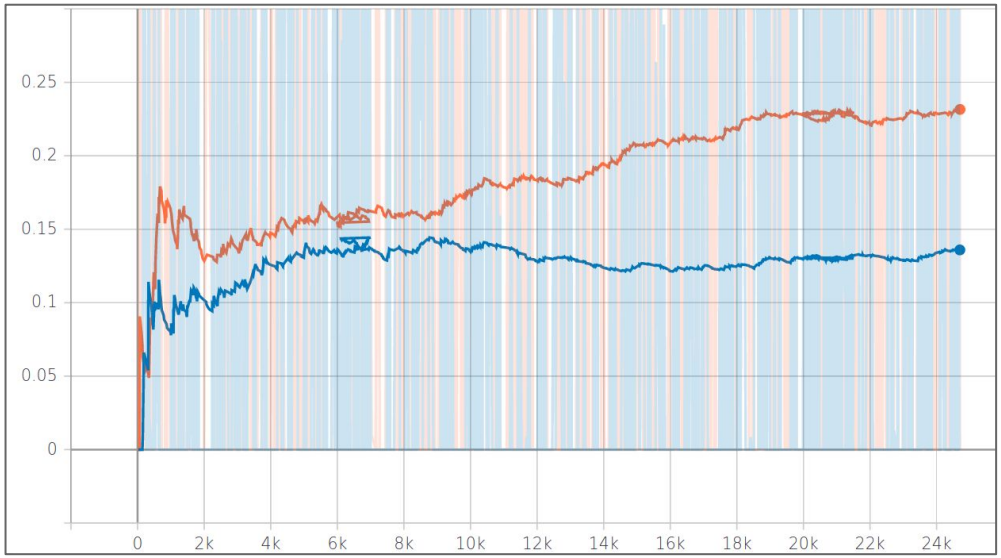
Deep Learning Predicted Result

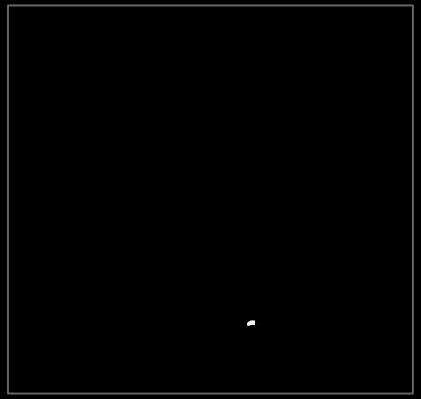
TERMINAL DEBUG CONSOLE PROBLEMS 8 OUTPUT

(MyVenv) PS D:\L_pipe\greg_branch\liverseg-

1 *Liver Segmentation Network*
Metrics

DICE SCORE - 0.8233 | TEST TIME - 4.11 Minutes

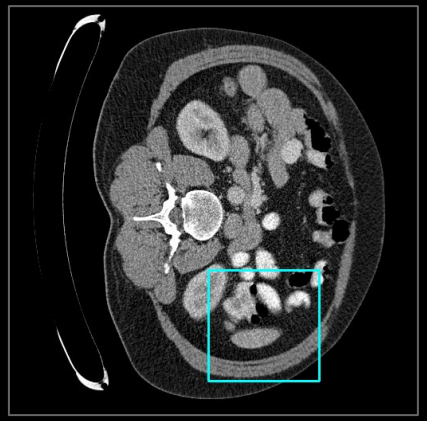




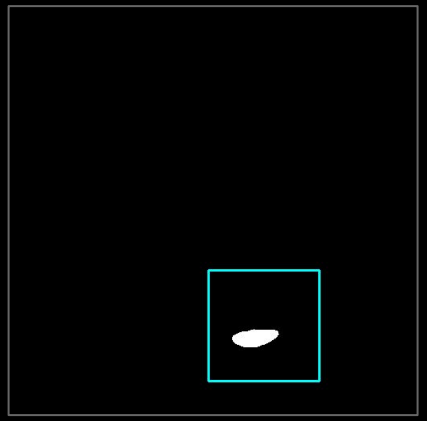
Liver identify by Radiologist

Deep Learning Predicted Result

2 Lesion Detection Network



Normal Patient CT Scan

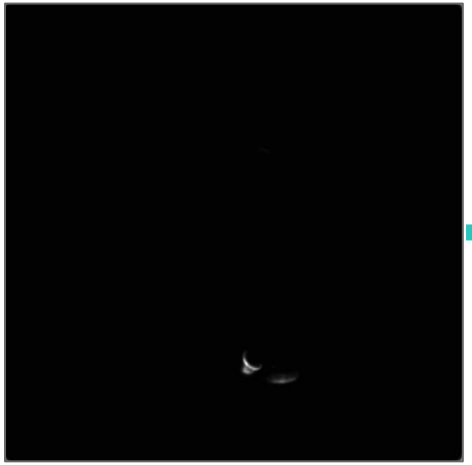


Deep Learning Predicted Result

```
TERMINAL  DEBUG CONSOLE  PROBLEMS 8  OUTPUT
png')

Done step: compute_bbs_from_gt_liver

Time taken: 71 seconds or 1.0 minutes 11s to

Running step: sample_bbs_test


Done step: sample_bbs_test

Time taken: 6 seconds or 0.0 minutes 6s to ru

Running step: det_lesion_test

Started loading training files...
Started loading validation files...
Started loading testing files...
Done initializing Dataset
INFO:tensorflow:Restoring parameters from D:\
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
```

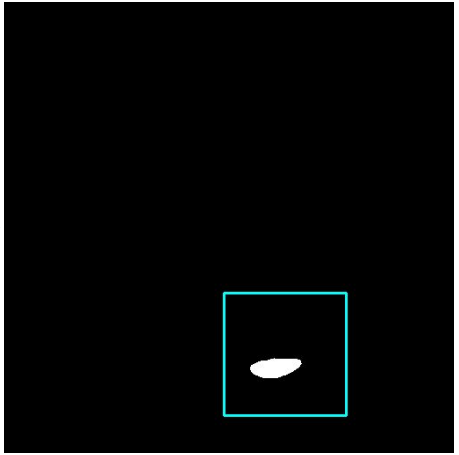
2 Lesion Detection Network

512 x 512 Liver Mask



Zoomed Liver Mask

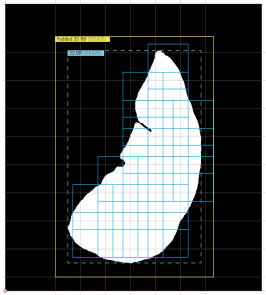
Sampled Liver Mask



Sample = Inputs to Lesion Detection Network

Why samples?

- ***Simpler*** inputs for ***faster*** computation & better ***accuracy***
- Can create ***extra data*** among scarcity



What is a “sample”?

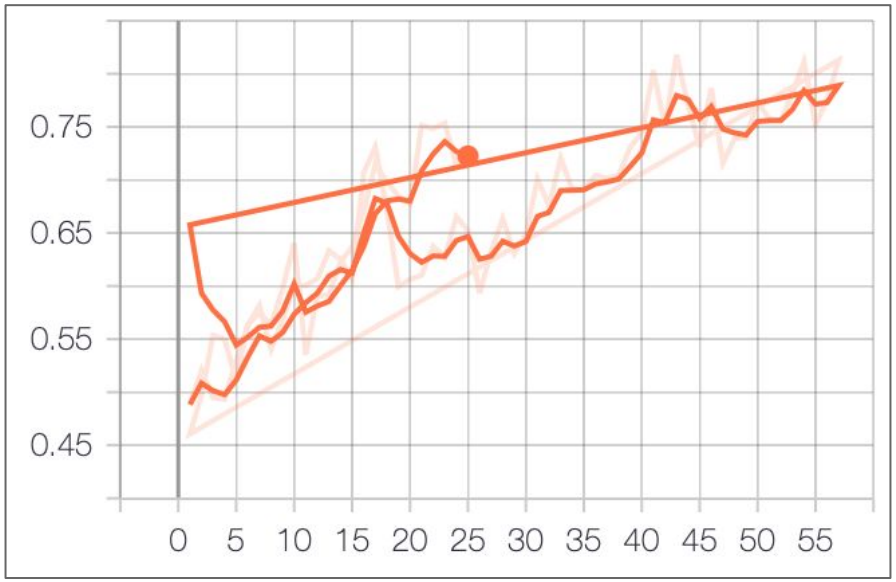
- 50 x 50 sample boxes w/ 15 pixel ***margins of error***
- A **positive** sample > 50 lesion pixels
- A **negative** sample < 50 lesion pixels

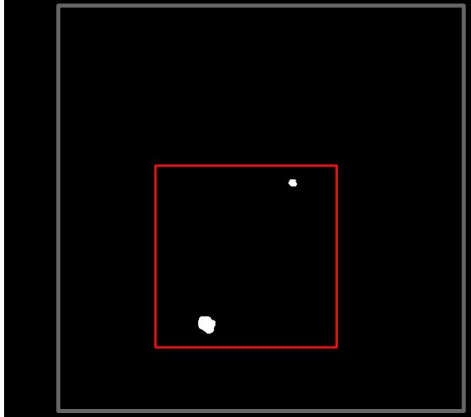
How do we determine which predicted lesions are segmented?

- If probability of lesion > 50% = a ***hard result***
- If probability of lesion < 50% = a ***soft result***

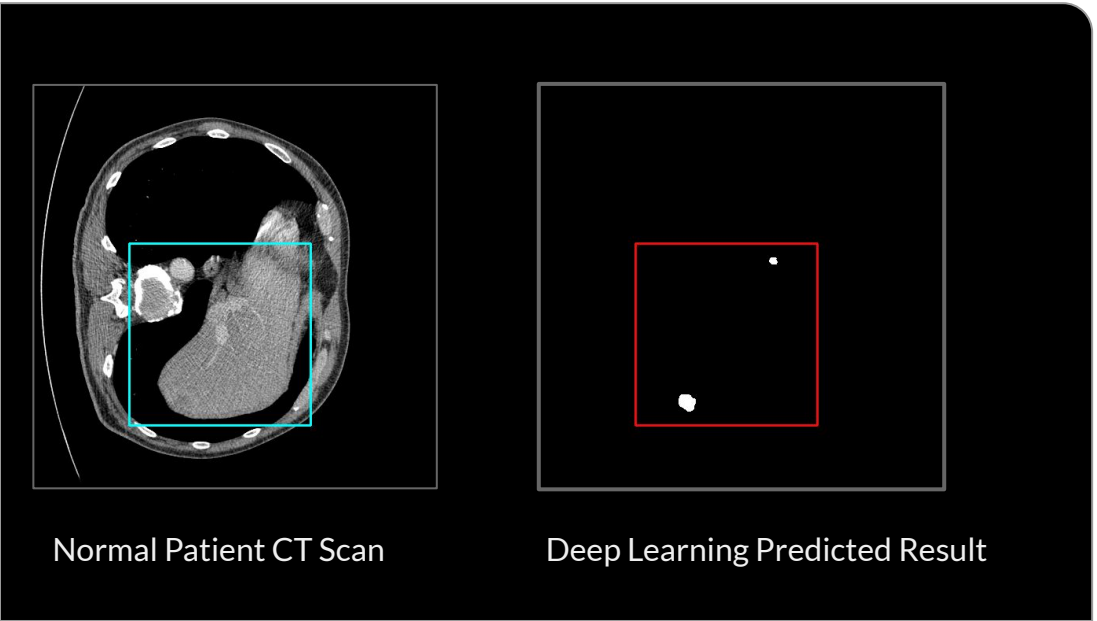
2 *Lesion Detection Network*
Metrics

DICE SCORE - 0.68 | TEST TIME - 1.90 Minutes



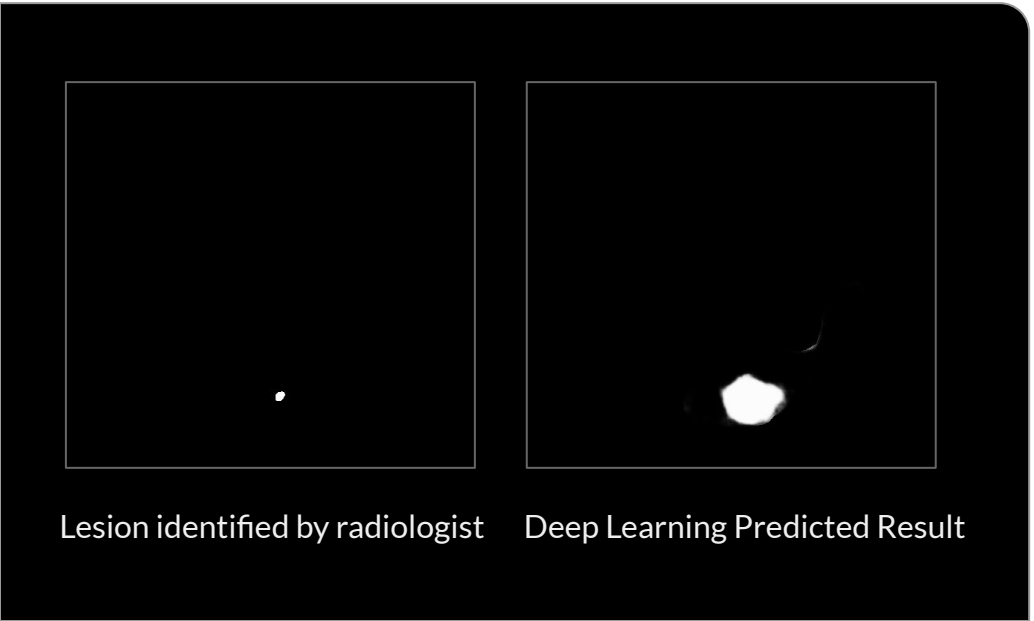


2 Lesion Detection Network




```
TERMINAL  DEBUG CONSOLE  PROBLEMS 8  OUTPUT
png')

Done step: compute_bbs_from_gt_liver

Time taken: 71 seconds or 1.0 minutes 11s to

Running step: sample_bbs_test


Done step: sample_bbs_test

Time taken: 6 seconds or 0.0 minutes 6s to ru

Running step: det_lesion_test

Started loading training files...
Started loading validation files...
Started loading testing files...
Done initializing Dataset
INFO:tensorflow:Restoring parameters from D:\
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
Testing D:\L_pipe\liver_open\liverseg-2017-ni
```

3 *Visual Results*
Lesion Segmentation Network



Lesion identified by radiologist

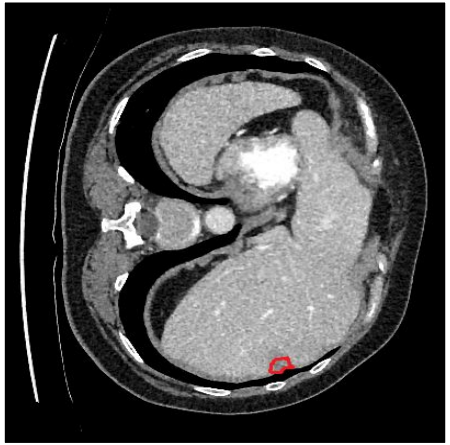
Deep Learning Predicted Result

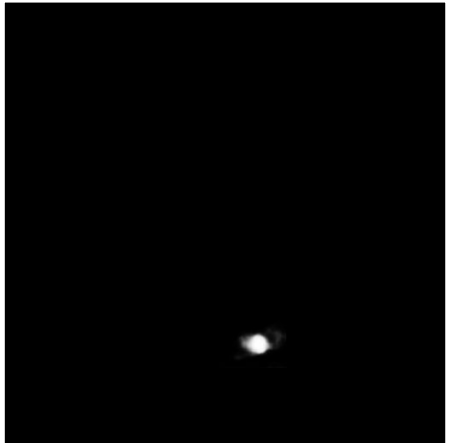
TERMINAL DEBUG CONSOLE PROBLEMS 8 OUTPUT

```
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips
Testing D:\L_pipe\liver_open\liverseg-2017-nips

Done step: det_lesion_test

Time taken: 152 seconds or 2.0 minutes 32s to r

Running step: seg_lesion_test

Started loading files...
Done initializing Dataset
segmenting lesions
INFO:tensorflow:Restoring parameters from D:\L_
Saving D:\L_pipe\greg_branch\liverseg-2017-nips
```

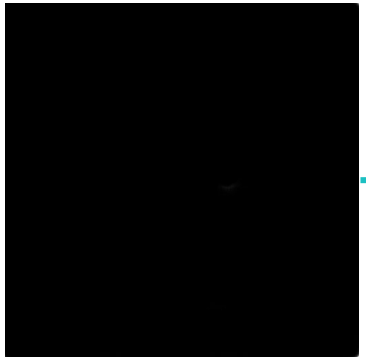
3 *Lesion Detection Network*
Metrics

DICE SCORE - **0.71** | TEST TIME - 4.36 Minutes

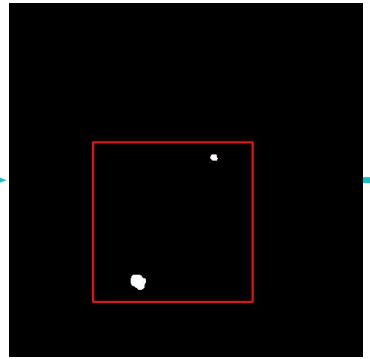




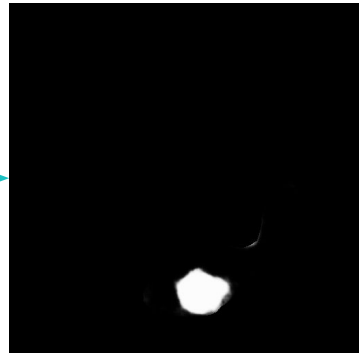
VISUAL OUTPUTS



Predicted Liver Mask



Detected Lesions



Predicted Lesion Mask

	Technical Description	DICE SCORE	Time
Liver Segmentation	Liver mask predicted by the model & segmented out of the region of interest (ROI) in the patient.	0.8233	4.11 m
Lesion Detection	Network classifies areas of interest in liver mask as positive (containing lesions) or negative.	0.68	1.90 m
Lesion Segmentation	The positive lesions are segmented out as results.	0.71	4.36 m

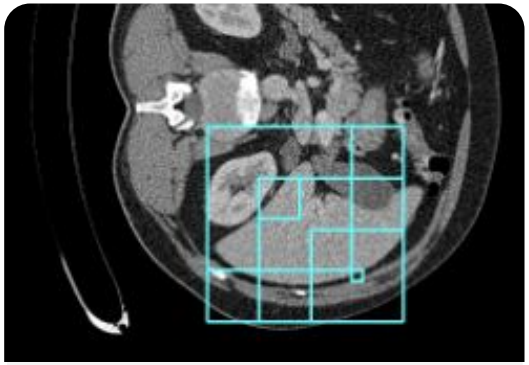
NEXT STEPS

March - June 2021

OpenVessel

Timelines & Deliverables

Specific Objectives



Improving DICE Lesion Detection

Adaptive Thresholding

Google Collab/ Jupyter notebooks

ROAR Supercomputer

User Interface

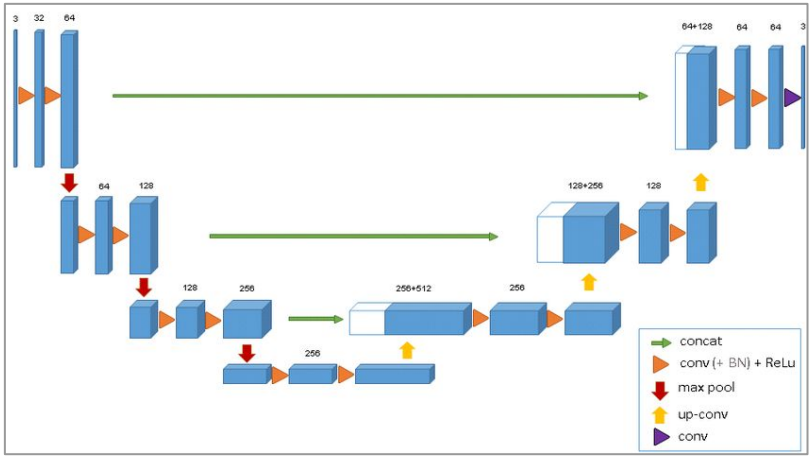
LI-RADS®

Dr. Jeph's Li-RADs Classification Data

Release Healthy Livers Dataset

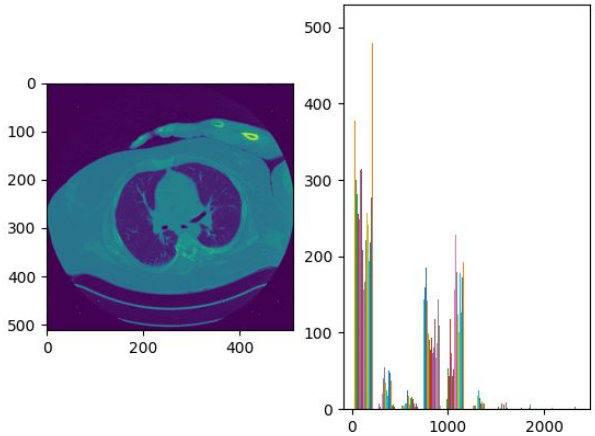
How to improve DICE?

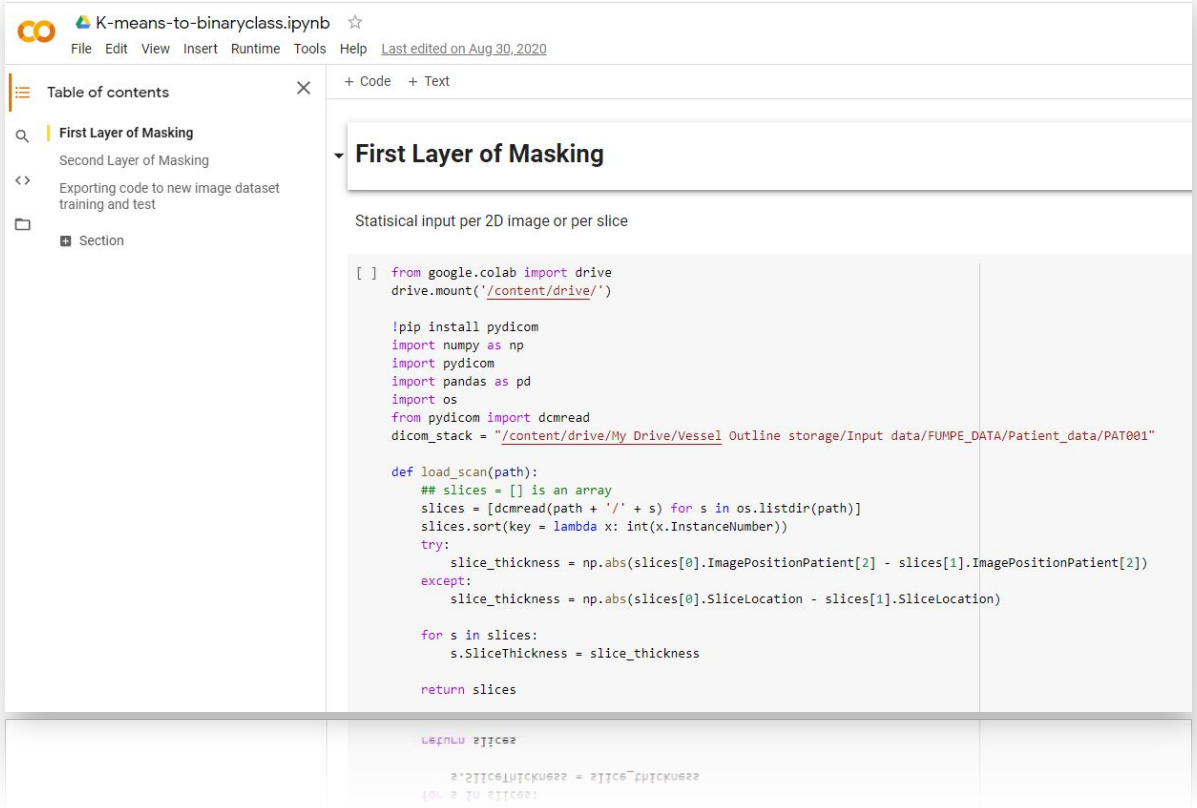
- Engineering preprocessing
- Better bounding boxes
- Implement 3D U-net

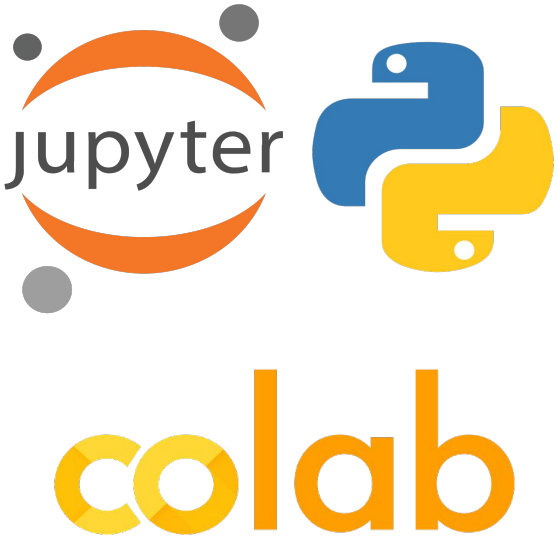


Adaptive Thresholding

Dataset analysis & Adaptive Thresholding Research (dataset description report)




K-means-to-binaryclass.ipynb
File Edit View Insert Runtime Tools Help Last edited on Aug 30, 2020
Table of contents
First Layer of Masking
Second Layer of Masking
Exporting code to new image dataset training and test
Section
+ Code + Text
First Layer of Masking
Statisical input per 2D image or per slice
[ ] from google.colab import drive
drive.mount('/content/drive/')
!pip install pydicom
import numpy as np
import pydicom
import pandas as pd
import os
from pydicom import dcmread
dicom_stack = "/content/drive/My Drive/Vessel Outline storage/Input data/FUMPE_DATA/Patient_data/PAT001"
def load_scan(path):
slices = [] is an array
slices = [dcmread(path + '/' + s) for s in os.listdir(path)]
slices.sort(key = lambda x: int(x.InstanceNumber))
try:
slice_thickness = np.abs(slices[0].ImagePositionPatient[2] - slices[1].ImagePositionPatient[2])
except:
slice_thickness = np.abs(slices[0].SliceLocation - slices[1].SliceLocation)
for s in slices:
s.SliceThickness = slice_thickness
return slices


jupyter
colab

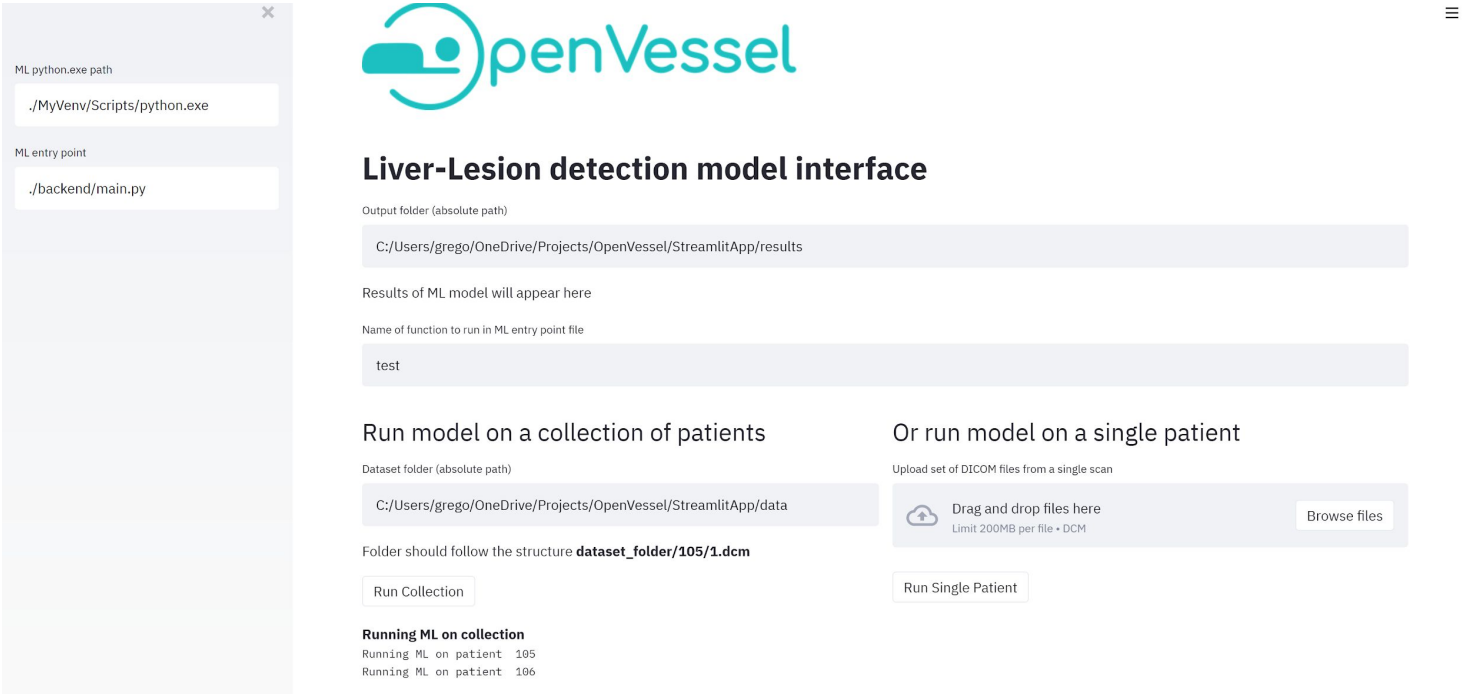
PSU Supercomputer

Industrial level Research

- Run 5> models all at once
- Unlimited storage
- CPU
- GPUs

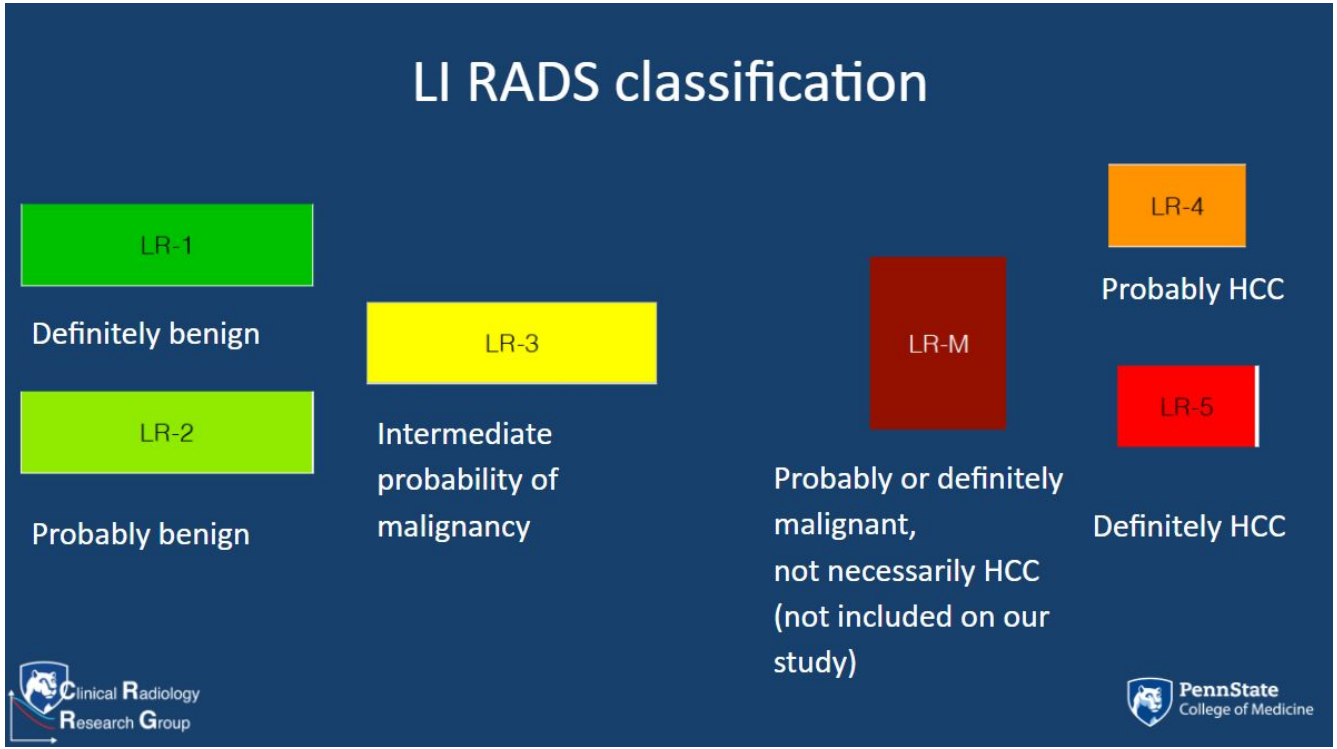
Dr. Justin Matthew Petucci

Test User Interface for Researchers



LiRADs Classification

LiRADs classification research (dataset description report)



Work Cited

[1] Patrick Bilic, et al. (2019). “The Liver Tumor Segmentation Benchmark (LiTS).” arXiv: 1901.04056v1.

[2] Kevis-Kokitsi Maninis and Jordi Pont-Tuset and Pablo Andrés Arbeláez and Luc Van Gool (2016). Deep Retinal Image Understanding*CoRR, abs/1609.01103*.

[3] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, & Olaf Ronneberger. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation.

Artwork Credits

Icons (Slides 3 - 5) sourced from flaticon.com and used in this document with attribution to creators:

“Neural” By Becris (Slide 3)
“Microscope” By Freepik (Slide 3)
“Contract” By Freepik (Slide 3)
“Presentation” By Pause08 (Slide 5)