

OpenScience (openSource + openData + openPaper)



Mahesh Shakya

Nepal Applied Mathematics and Informatics Institute for research (NAAMII)

Sagarmatha Hacktoberfest 2023

This talk

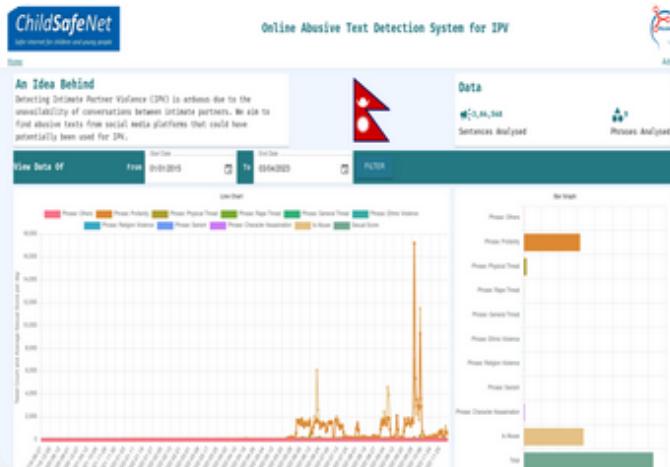
- NAAMII OpenScience and Education Outreach Initiative
- Why OpenScience?
- Medical Open Network for AI (MONAI)

NAAMII Profile



Featured Research Projects

Abusive Nepali Text Detection



This project focuses on aiding IPV research to understand the nature and prevalence of online IPV

[READ MORE](#)

AI-assisted VIA Screening of Cervical Cancer



This project aims to research and develop AI-assisted VIA screening using mobile-captured photos

[READ MORE](#)

AI Assisted Microscopy



This project aims to develop object detection models to detect diarrhea cysts from vegetables, stool, and

[READ MORE](#)

NAAMII Profile

Incubating & Supporting Start-ups

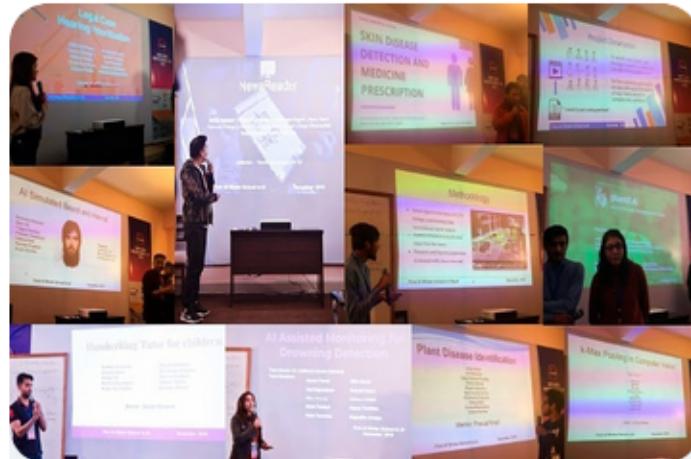


NAAMII is incubating Diyo.AI, providing scientific support for developing the next generation of

[READ MORE](#)

Industry & Innovation

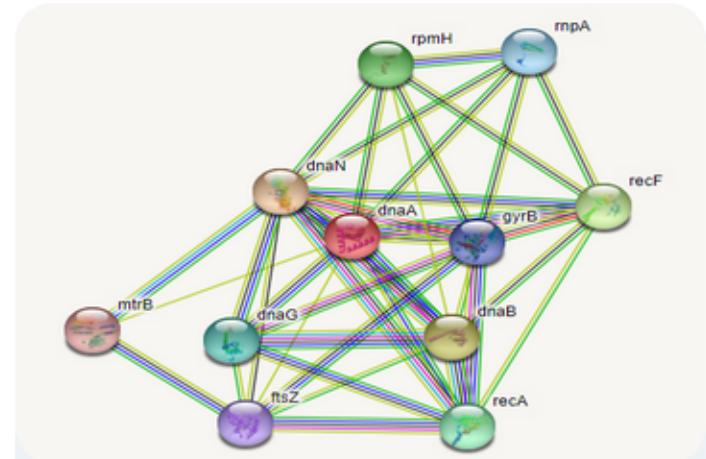
Industry Training & Capacity Building



NAAMII Research Scientist has trained, mentored, and helped build a machine learning team at different

[READ MORE](#)

Research & Development for Industry



Our team, led by a Research Scientist, does research and development work for different organization in

[READ MORE](#)

NAAMII Profile

Education & Outreach Section

Annual Nepal AI School (AN AIS)



NAAMII annually organizes Nepal AI school and has successfully conducted three winter editions. The

[READ MORE](#)

Training Workshops



NAAMII conducted a 3-month summer research Training program from June-August 2020. 40 highly

[READ MORE](#)

Public Lecture Series



NAAMII has been organizing different lecture series in collaboration with

[READ MORE](#)

Education Outreach @ NAAMII

- Annual Nepal Winter School in AI (2018, 2019, 2021, 2023)

Speakers: #ANAIS2023

KEYNOTE SPEAKERS



Prof. Angela Yao,
National University of
Singapore, Singapore
(In-person)



Prof. CV Jawahar,
IIIT Hyderabad,
India
(In-Person)



Dr. Gaël
Varoquaux,
Inria, France
(Virtual)



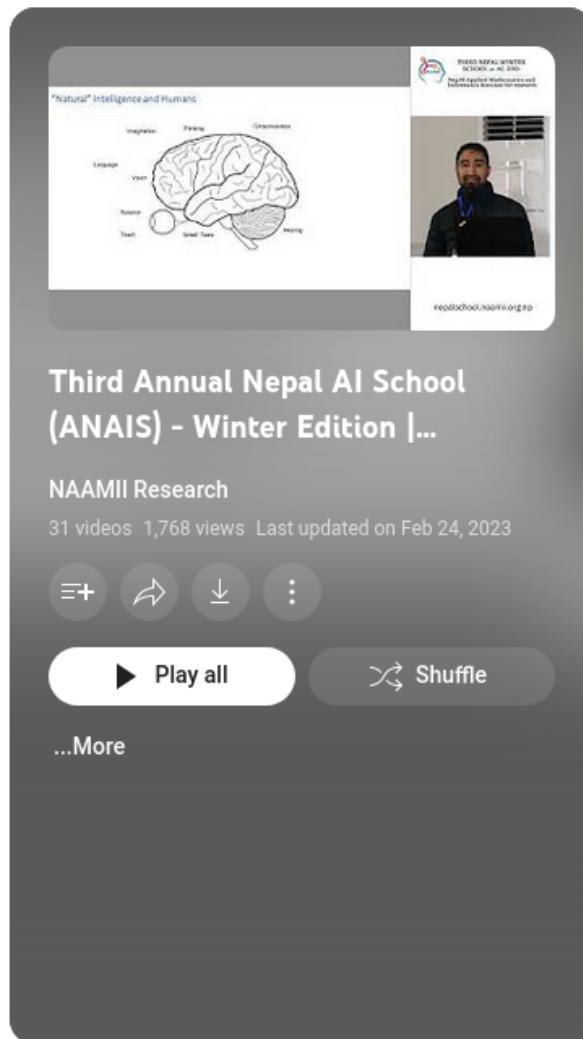
Prof. Michael
Bronstein,
University of Oxford, UK
(Virtual)



Prof. Pushpak
Bhattacharyya,
IIT Bombay, India
(Virtual)

Education Outreach @ NAAMII

- Annual Nepal Winter School in AI (2018, 2019, 2021, 2023)



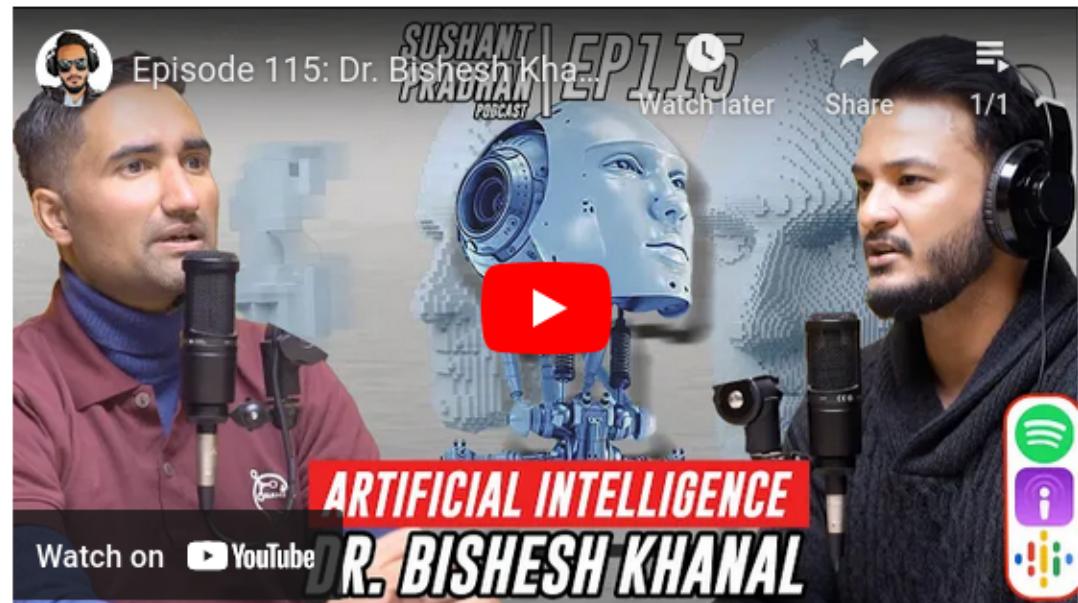
- 1 **Introduction to Machine Learning & AI School | Third Nepal Winter School in AI | Day 1 (2021)**
NAAMII Research • 1.1K views • 1 year ago
1:12:07
- 2 **Probability & Statistics for Machine Learning | Third Nepal Winter School in AI | Day 1 (2021)**
NAAMII Research • 510 views • 1 year ago
1:02:50
- 3 **Learning to Learn Linear Algebra | Third Nepal Winter School in AI | Day 1 (2021)**
NAAMII Research • 330 views • 1 year ago
1:09:22
- 4 **Linear Regression/Logistic Regression | Third Nepal Winter School in AI | Day 1 (2021)**
NAAMII Research • 177 views • 1 year ago
56:46
- 5 **Setting Up Supervised Learning Problems - Part 1 | Third Nepal Winter School in AI | Day 2 (2021)**
NAAMII Research • 192 views • 1 year ago
1:03:35
- 6 **Convolutional Neural Networks (CNNs) - Part 1 | Third Nepal Winter School in AI | Day 3 (2021)**
NAAMII Research • 255 views • 1 year ago
1:30:59

⋮

7/30

NAAMII Profile

NAAMII in Media



[View More](#)



OpenScience @ NAAMII

NepAI Applied Mathematics and Informatics Institute for research

Overview Repositories Projects Packages People

Find a repository... Type Language Sort

synthetic-boost Public

Synthetic Boost: Leveraging Synthetic Data for Enhanced Vision-Language Segmentation in Echocardiography

paper ml cardiac-segmentation echocardiography

Python MIT 0 stars 1 forks 3 updated 2 days ago

medvlsm Public

Exploring Transfer Learning in Medical Image Segmentation using Vision-Language Models

segmentation medical-images vision-and-language

Jupyter Notebook MIT 0 stars 4 forks 9 updated 2 days ago

xrayto3D-benchmark Public

Jupyter Notebook GPL-3.0 1 star 1 fork 25 updated last week

colorimetric_analysis_final Public

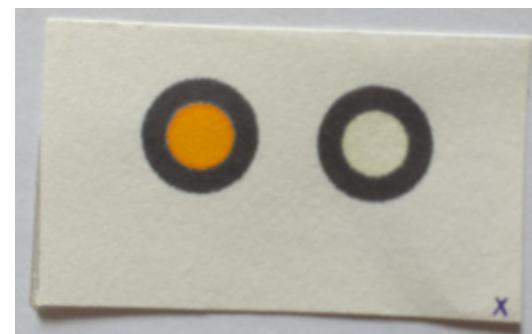
This is the final version of colorimetric analysis codes.

Jupyter Notebook MIT 0 stars 0 forks 0 updated 2 weeks ago

github.com/naamiinepal



OpenScience @ NAAMII

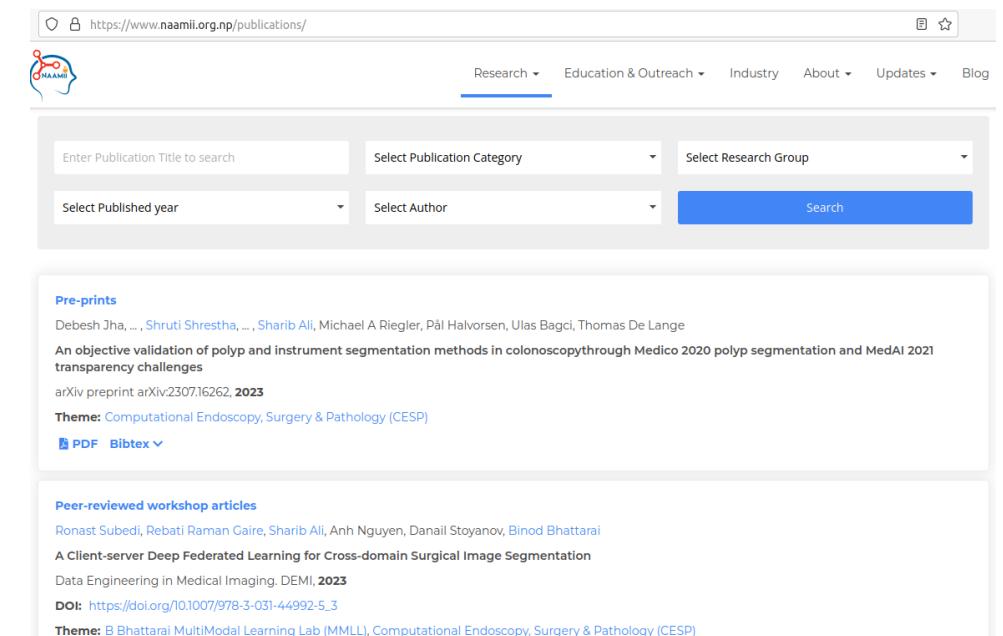


Example	English Translation	Label Prediction
कोरोनाको परीक्षण किन घटाउँदै छ नालायक सरकार?	Why is the worthless government reducing the testing of Corona?	COVID Politics, Civic views
पछिल्लो २४ घण्टामा थप ९४४ जनामा कोरोना संक्रमण पुष्टि, संकरिय संक्रमितको संख्या ६ हजार नाघ्यो	In the last 24 hours, 944 more Corona infections have been confirmed, and the number of active infected has exceeded 6 thousand	COVID Stats



open science
open code
open data
open papers

OpenScience @ NAAMII



https://www.naamii.org.np/publications/

Research Education & Outreach Industry About Updates Blog

Enter Publication Title to search Select Publication Category Select Research Group

Select Published year Select Author Search

Pre-prints
Debesh Jha, ... , Shruti Shrestha, ... , Sharib Ali, Michael A Riegler, Pål Halvorsen, Ulas Bagci, Thomas De Lange
An objective validation of polyp and instrument segmentation methods in colonoscopy through Medico 2020 polyp segmentation and MedAI 2021 transparency challenges
arXiv preprint arXiv:2307.16262, 2023
Theme: Computational Endoscopy, Surgery & Pathology (CESP)
[PDF](#) [Bibtex](#)

Peer-reviewed workshop articles
Ronast Subedi, Rebatli Raman Gaire, Sharib Ali, Anh Nguyen, Danail Stoyanov, Binod Bhattacharai
A Client-server Deep Federated Learning for Cross-domain Surgical Image Segmentation
Data Engineering in Medical Imaging. DEMI, 2023
DOI: https://doi.org/10.1007/978-3-031-44992-5_3
Theme: B Bhattacharai MultiModal Learning Lab (MMLL), Computational Endoscopy, Surgery & Pathology (CESP)

www.naamii.org.np/publications

Why OpenScience?

- Reproducibility
- accelerate pace of research
- OpenScience in AI
 - Open access software: pytorch, keras
accelerate innovation with low barrier to entry
 - Open access data
publicly available data
 - Open access publications:
arXiv

Why OpenScience?

Benchmarking Encoder-Decoder Architectures for Biplanar X-ray to 3D Shape Reconstruction

Mahesh Shakya Bishesh Khanal

NepAI Applied Mathematics and Informatics Institute for research (NAAMII)
`{mahesh.shakya, bishesh.khanal}@naamii.org.np`

Abstract

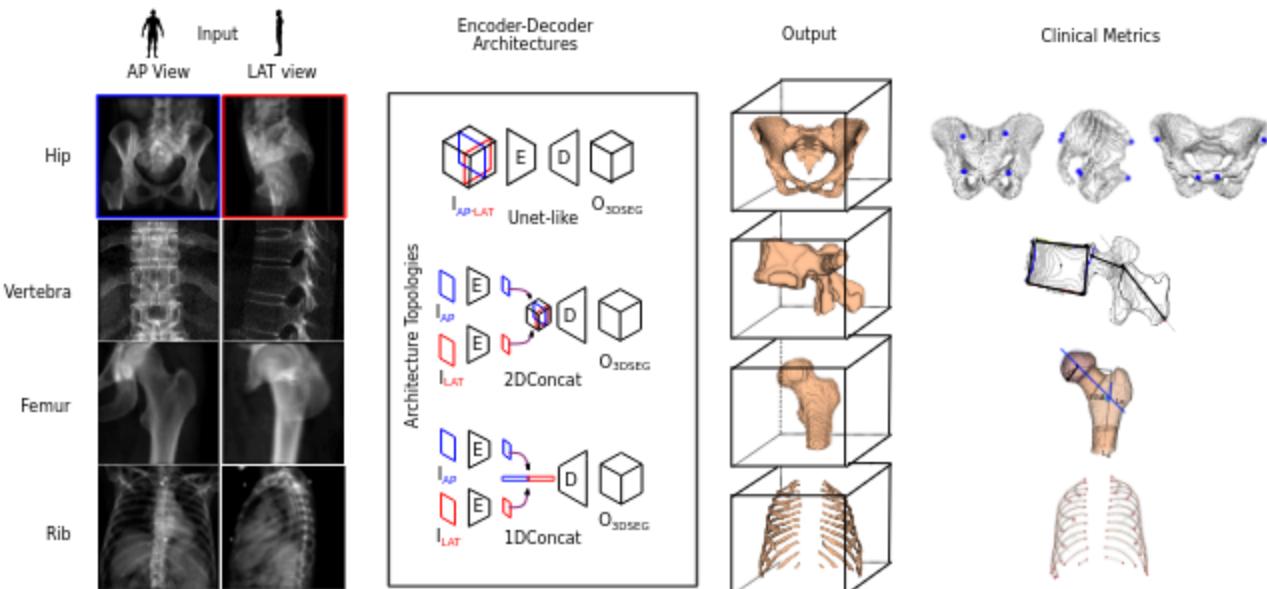
Why OpenScience?

Benchmarking Encoder-Decoder Architectures for Biplanar X-ray to 3D Shape Reconstruction

Mahesh Shakya Bishesh Khanal

NepAI Applied Mathematics and Informatics Institute for research (NAAMII)
`{mahesh.shakya, bishesh.khanal}@naamii.org.np`

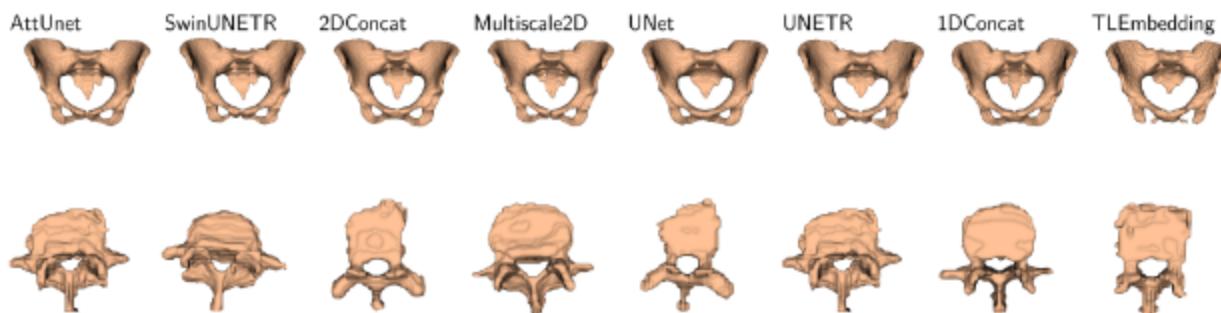
Abstract



Why OpenScience?

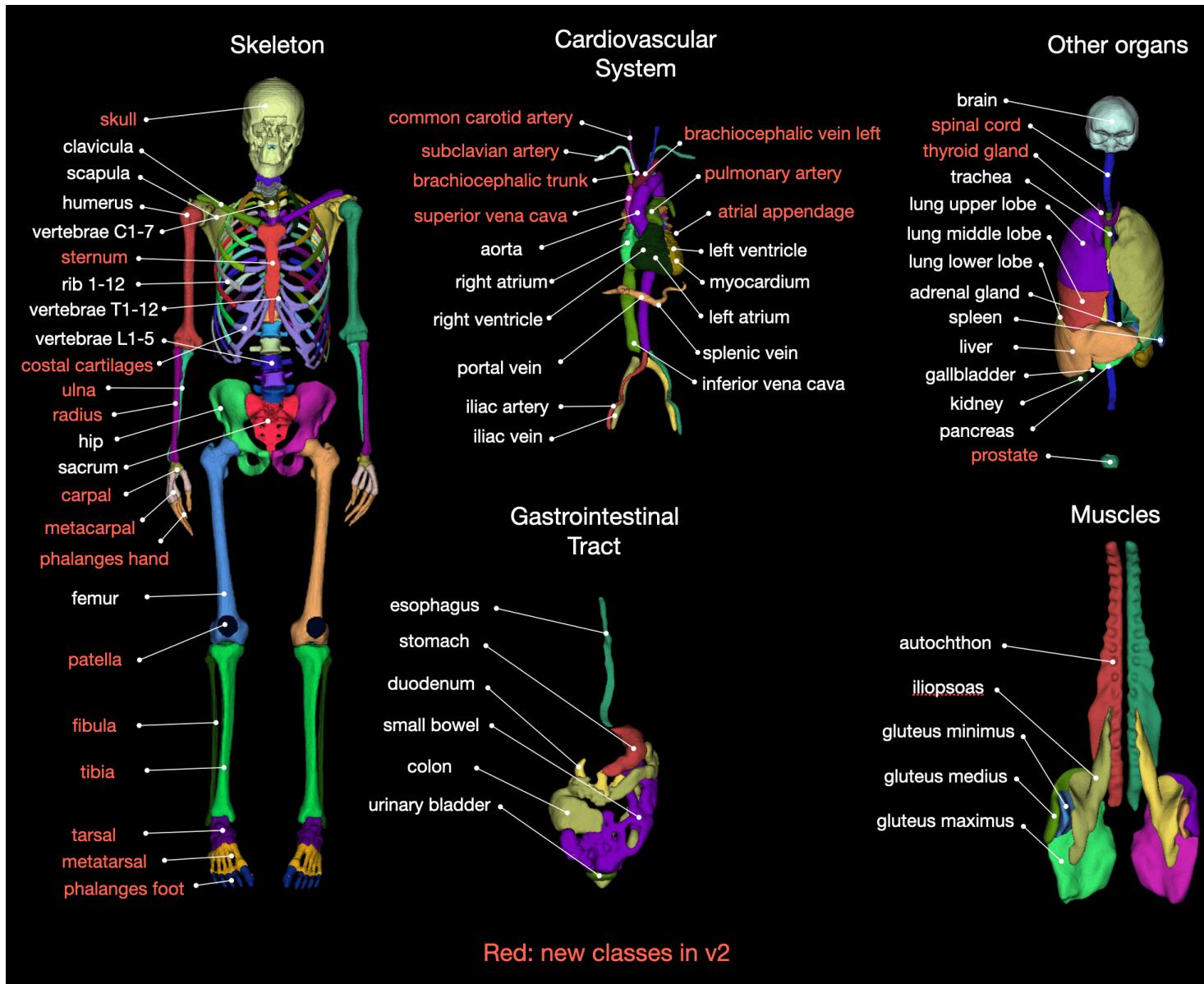
Table 1: Benchmark Evaluation

Dataset (#train/#test) (vol size (voxel resolution)	Method Reference	#Param	Dice(%)↑	HD95(mm)↓	ASD(mm)↓	NSD@1.5mm↑
TotalSeg-Femur-DRR 786/138 128x128x128 1.0	SwinUNETR [20]	62.2M	93.64	3.32	0.94	0.85
	AttentionUnet [38]	1.5M	93.66	3.12	0.93	0.85
	2DConcat [4]	1.2M	93.05	3.73	1.02	0.83
	UNet [28]	1.2M	93.36	3.32	0.95	0.84
	MultiScale2DConcat [57]	3.5M	92.79	3.64	1.11	0.81
	UNETR [21]	96.2M	92.39	3.96	1.16	0.80
	TL-Embedding [17]	6.6M	90.43	4.28	1.41	0.73
	1DConcat [11]	40.6M	89.66	4.86	1.57	0.67



Need for Public Dataset

TotalSegmentor: CT scan segmentation of major bones and organs



Public Dataset Curation

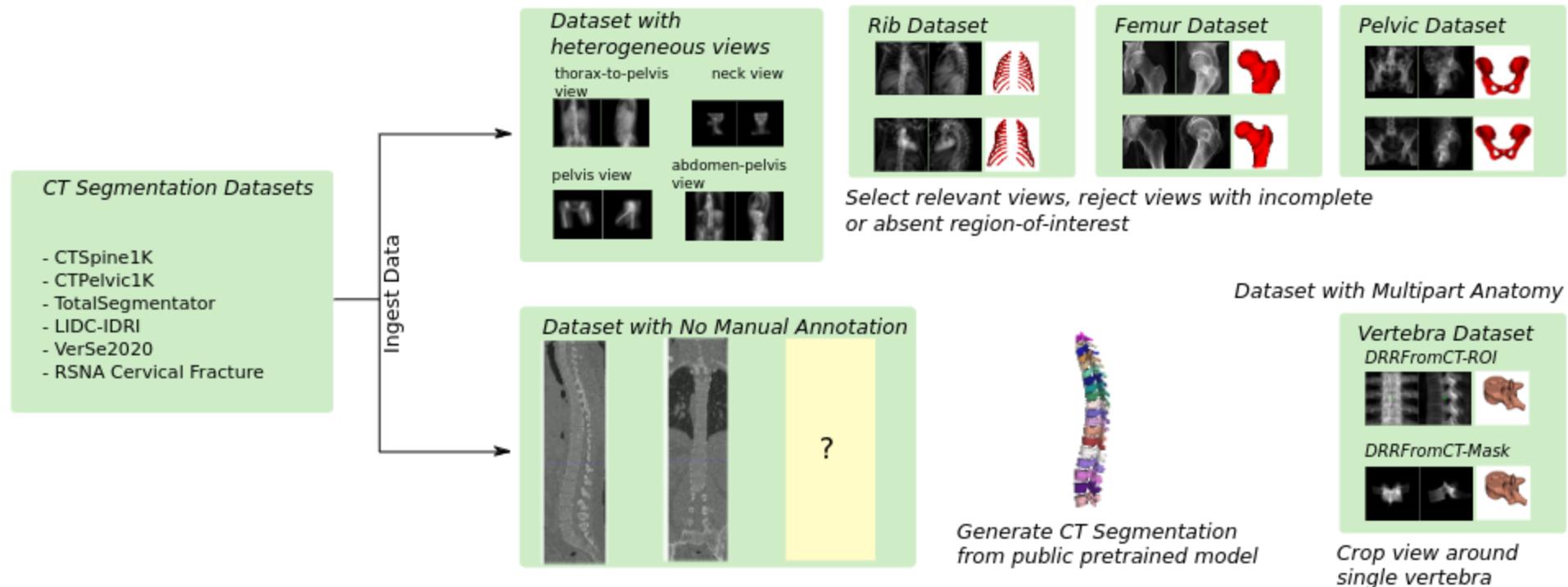


Figure 2: Data Ingestion: Various data preprocessing scenarios for Ingesting CT Segmentation Datasets for Biplanar X-ray to 3D Bone Shape Dataset

Medical Open Network for AI (MONAI)

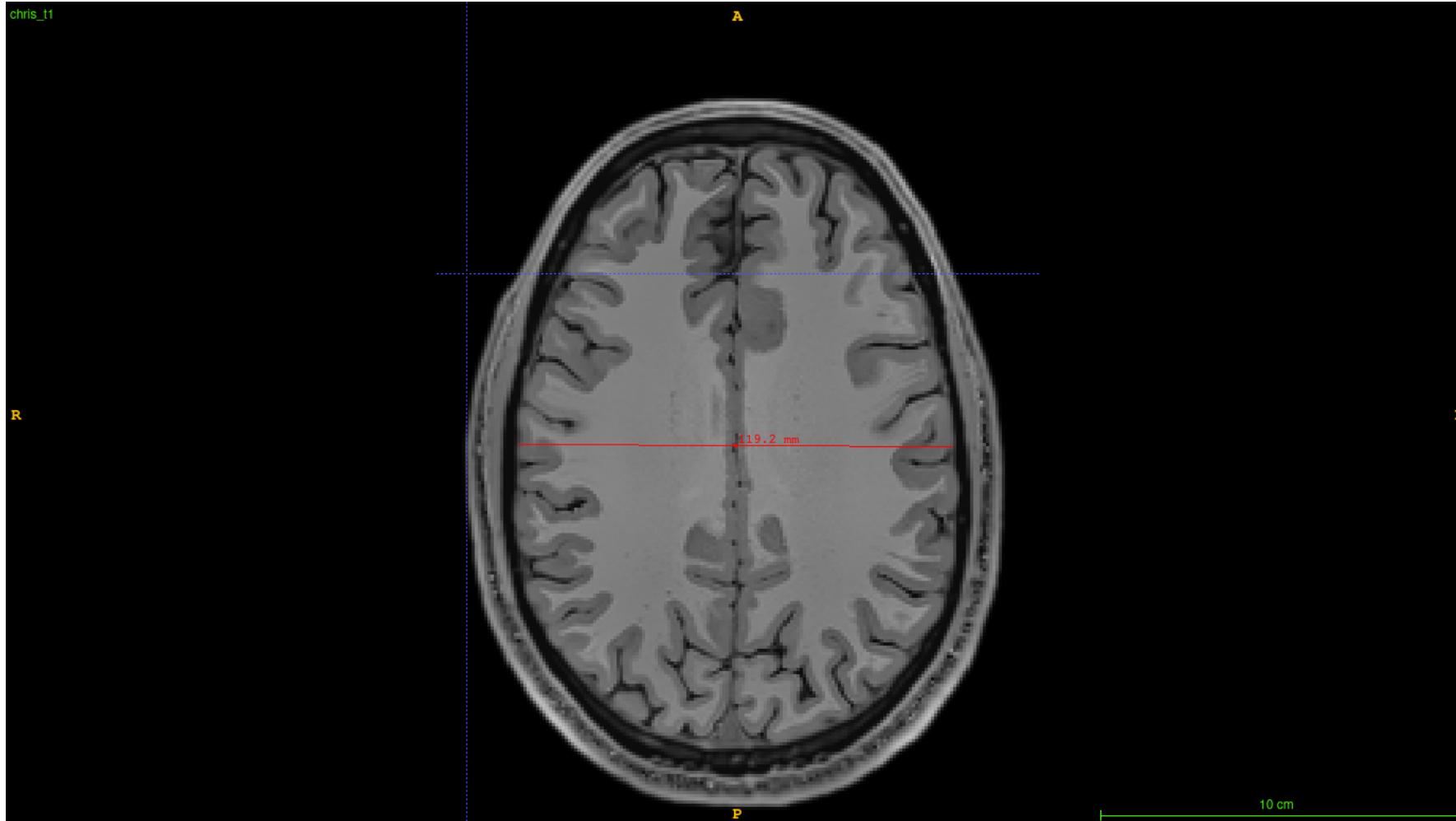
Goal: accelerate pace of research and development by providing a common software foundation and a vibrant community for medical imaging deep learning

Slides adapted from Stephen Aylward's MONAI and Open Science presentation at MONAI MIDL Meetup 2023

Why MONAI ?

- Medical Imaging applications have specific requirements
- Image formats require special support: DICOM, NIfTI etc.
- Image meta-data must be considered: voxel spacing, HU etc.
- Image modalities have specific processing methods: MRI, CT, etc.
- domain-specific Problems: sample size limitations, image sizes

Medical Imaging has specific requirements



MONAI and Open Science

- reduce re-implementation
- provide baseline implementations
- demonstrate best practices

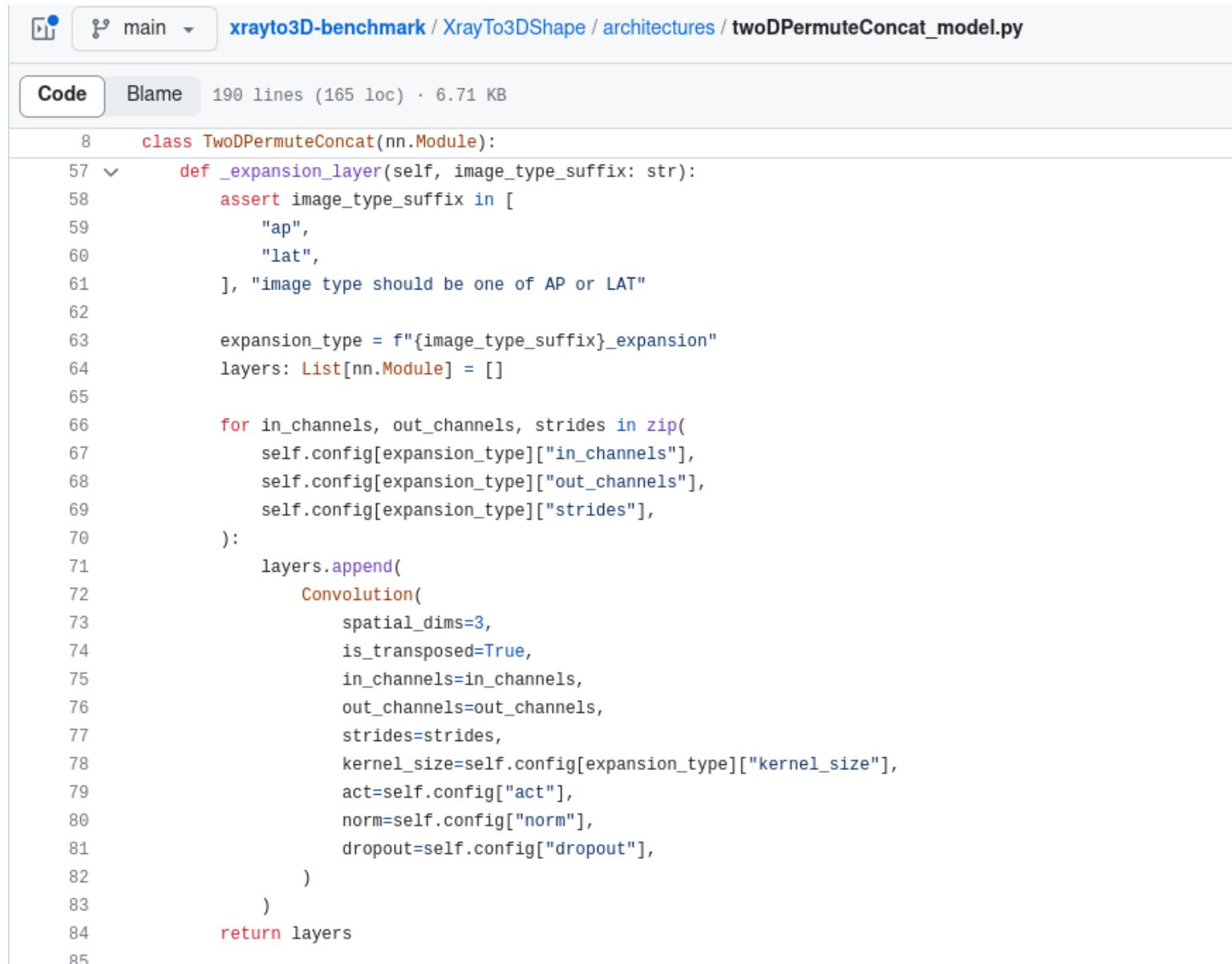
MONAI: demonstrate best practices

dev ▾ MONAI / monai / networks / nets / autoencoder.py

Code Blame 306 lines (268 loc) · 12.3 KB Raw ⌂ ⌂

```
25
26 class AutoEncoder(nn.Module):
27     """
28         Simple definition of an autoencoder and base class for the architecture implementing
29         :py:class:`monai.networks.nets.VarAutoEncoder`. The network is composed of an encode sequence of blocks, followed
30         by an intermediary sequence of blocks, and finally a decode sequence of blocks. The encode and decode blocks are
31         default :py:class:`monai.networks.blocks.Convolution` instances with the encode blocks having the given stride
32         and the decode blocks having transpose convolutions with the same stride. If `num_res_units` is given residual
33         blocks are used instead.
34
35         By default the intermediary sequence is empty but if `inter_channels` is given to specify the output channels of
36         blocks then this will become a sequence of Convolution blocks or of residual blocks if `num_inter_units` is
37         given. The optional parameter `inter_dilations` can be used to specify the dilation values of the convolutions in
38         these blocks, this allows a network to use dilated kernels in this middle section. Since the intermediary section
39         isn't meant to change the size of the output the strides for all these kernels is 1.
40
41     Args:
42         spatial_dims: number of spatial dimensions.
43         in_channels: number of input channels.
44         out_channels: number of output channels.
45         channels: sequence of channels. Top block first. The length of `channels` should be no less than 2.
46         strides: sequence of convolution strides. The length of `stride` should equal to `len(channels) - 1`.
47         kernel_size: convolution kernel size, the value(s) should be odd. If sequence,
48             its length should equal to dimensions. Defaults to 3.
49         up_kernel_size: upsampling convolution kernel size, the value(s) should be odd. If sequence,
50             its length should equal to dimensions. Defaults to 3.
51         num_res_units: number of residual units. Defaults to 0.
52         inter_channels: sequence of channels defining the blocks in the intermediate layer between encode and decode.
53         inter_dilations: defines the dilation value for each block of the intermediate layer. Defaults to 1.
54         num_inter_units: number of residual units for each block of the intermediate layer. Defaults to 0.
55         act: activation type and arguments. Defaults to PReLU.
```

learn from MONAI code repo



The screenshot shows a GitHub code editor interface. At the top, there's a navigation bar with a user icon, a 'main' dropdown, and the repository path 'xrayto3D-benchmark / XrayTo3DShape / architectures / twoDPermuteConcat_model.py'. Below the navigation is a toolbar with 'Code' (which is selected) and 'Blame' buttons, along with statistics: '190 lines (165 loc) · 6.71 KB'. The main area displays the Python code for the 'TwoDPermuteConcat' module. The code defines a class 'TwoDPermuteConcat' that inherits from 'nn.Module'. It includes methods for expanding layers based on image type suffixes ('ap' or 'lat') and a constructor that initializes layers for each channel configuration.

```
8  class TwoDPermuteConcat(nn.Module):
57     def __init__(self, image_type_suffix: str):
58         assert image_type_suffix in [
59             "ap",
60             "lat",
61         ], "image type should be one of AP or LAT"
62
63         expansion_type = f"{image_type_suffix}_expansion"
64         layers: List[nn.Module] = []
65
66         for in_channels, out_channels, strides in zip(
67             self.config[expansion_type]["in_channels"],
68             self.config[expansion_type]["out_channels"],
69             self.config[expansion_type]["strides"],
70         ):
71             layers.append(
72                 Convolution(
73                     spatial_dims=3,
74                     is_transposed=True,
75                     in_channels=in_channels,
76                     out_channels=out_channels,
77                     strides=strides,
78                     kernel_size=self.config[expansion_type]["kernel_size"],
79                     act=self.config["act"],
80                     norm=self.config["norm"],
81                     dropout=self.config["dropout"],
82                 )
83             )
84     return layers
85
```

MONAI Model Zoo

Brats mri axial slices generative diffusion

MONAI team

A generative model for creating 2D brain MRI axial slices from Gaussian noise based on BraTS dataset

[Model Details](#)

Brats mri generative diffusion

MONAI team

A generative model for creating 3D brain MRI from Gaussian noise based on BraTS dataset

[Model Details](#)

Brats mri segmentation

MONAI team

A pre-trained model for volumetric (3D) segmentation of brain tumor subregions from multimodal MRIs based on BraTS 2018 data

[Model Details](#)

Breast density classification

Center for Augmented Intelligence in Imaging, Mayo Clinic Florida

A pre-trained model for classifying breast images (mammograms)

[Model Details](#)

Endoscopic inbody classification

NVIDIA DLMED team

A pre-trained binary classification model for endoscopic inbody classification task

[Model Details](#)

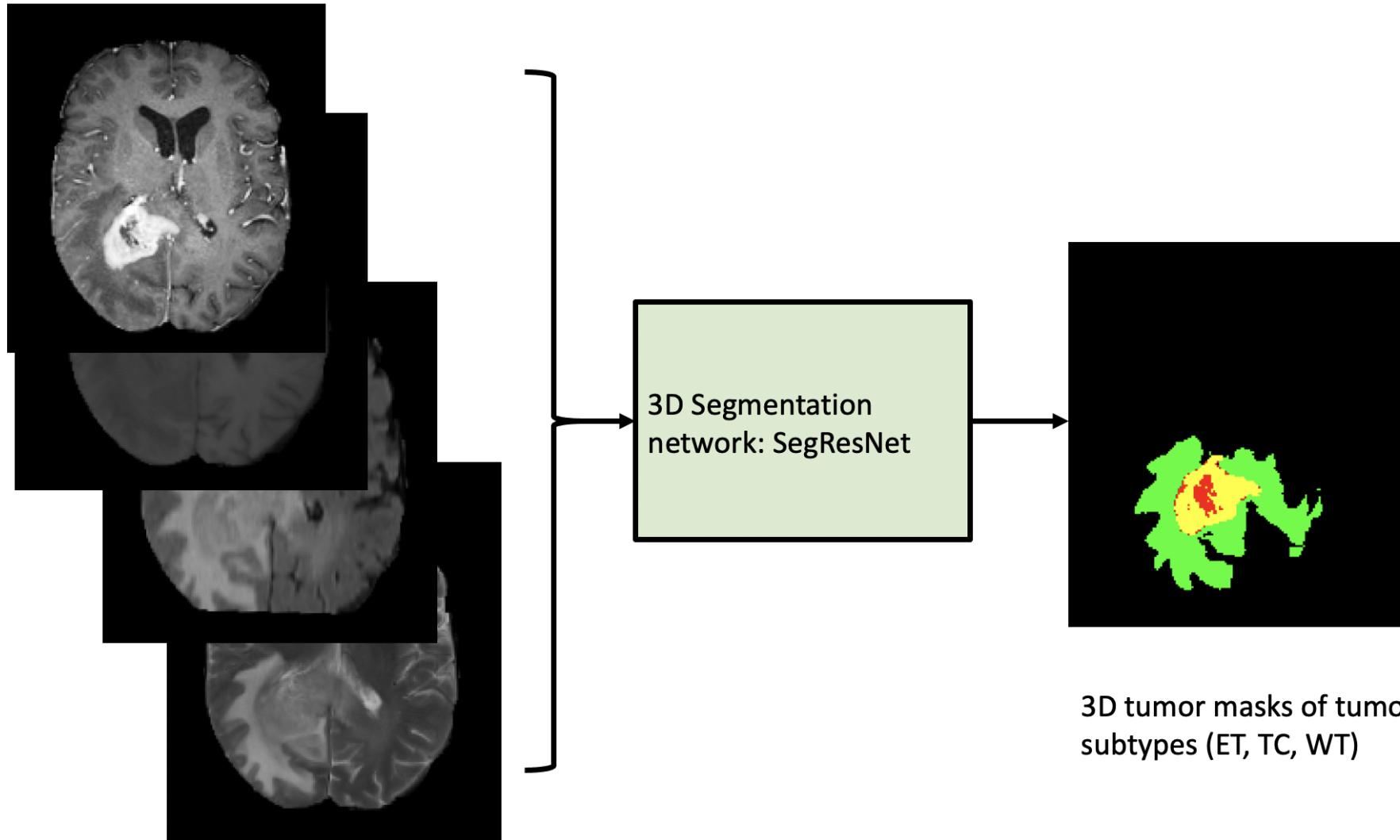
Endoscopic tool segmentation

NVIDIA DLMED team

A pre-trained binary segmentation model for endoscopic tool segmentation

[Model Details](#)

MONAI Model Zoo: MRI Tumor Segmentation



4x 3D brain MRIs modalities
(T1c, T1, T2, Flair)

MONAI Model Zoo: MRI Tumor Segmentation

Brats mri segmentation

Download

X

Execute inference:

```
python -m monai.bundle run --config_file configs/inference.json
```

Export checkpoint to TensorRT based models with fp32 or fp16 precision:

```
python -m monai.bundle trt_export --net_id network_def \
--filepath models/model_trt.ts --ckpt_file models/model.pt \
--meta_file configs/metadata.json --config_file configs/inference.json \
--precision <fp32/fp16> --input_shape "[1, 4, 240, 240, 160]" --use_onnx "True" \
--use_trace "True"
```

Execute inference with the TensorRT model:

```
python -m monai.bundle run --config_file "[configs/inference.json', 'configs/inference_trt.json']"
```

References

[1] Myronenko, Andriy. "3D MRI brain tumor segmentation using autoencoder regularization." International MICCAI Brainlesion Workshop. Springer, Cham, 2018.

MONAI Core

- patch inference
- state-of-the-art architecture implementations
- data preprocessing

Gap between AI Model and Deployable App

- Load DICOM datasets
- Preprocess Input Images
- Perform Inference
- Post Process results
- Export AI results to DICOM
- Visualize inference results

Find more here

- Documentation
<https://docs.monai.io>
- MONAI Issue Tracker / Discussion Board
<https://github.com/Project-MONAI/MONAI>

