MONAI

Medical Open Network for AI

14/06/2021 - Thibault Pelletier

Kitware

Contributors

In addition to Kitware, Project MONAI is supported by NVIDIA, King's College London, Chinese Academy of Sciences, DKFZ German Cancer Research Center, Stanford University, MGH & BWH Center for Clinical Data Science, and the Technical University of Munich.

Purpose

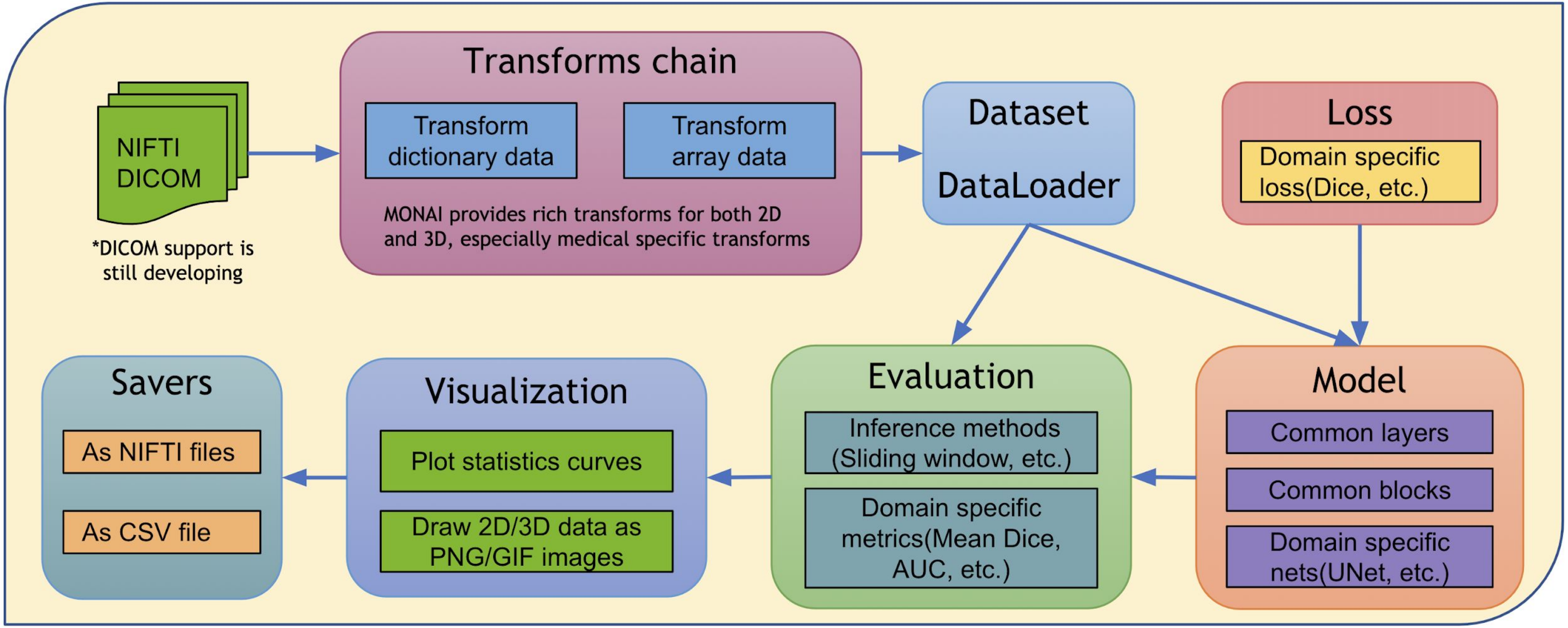
MONAI is a PyTorch-based, open-source framework for deep learning in healthcare imaging, part of PyTorch Ecosystem. Its ambitions are:

- Developing a community of academic, industrial and clinical researchers collaborating on a common foundation;
- Creating state-of-the-art, end-to-end training workflows for healthcare imaging;
- Providing researchers with the optimized and standardized way to create and evaluate deep learning models.

Key features

- Flexible pre-processing for multi-dimensional medical imaging data;
- Compositional & portable APIs for ease of integration in existing workflows;
- Domain-specific implementations for networks, losses, evaluation metrics and more;
- Customizable design for varying user expertise;
- Multi-GPU data parallelism support.

Architecture



Research - implementation of recently published research in medical imaging, such as COVID-19 models, automatic model parallelism

- Model Parallel
- COVID-19 CT Image Segmentation
- Other Topics

Examples & Application - show highlight features and integration with other packages

- Segmentation
- Classification
- Perf Acceleration
- Modules Demo
- Public Datasets

Workflows - engines, metrics and event-handlers that are compatible with PyTorch ignite APIs, support AMP and distributed data parallel

Engines
SupervisedTrainer
GANTrainer
SupervisedEvaluator
EnsembleEvaluator

Event-Handlers
CheckpointLoader
CheckpointSaver
TensorBoardHandlers
ValidationHandler
MetricLogger
SegmentationSaver
ClassificationSaver
StatsHandler
LrScheduleHandler

Metrics
MeanDice
ROCAUC
ConfusionMatrix

Components - independent modules that can be integrated into PyTorch programs directly

Data
CacheDataset
SmartCacheDataset
PersistentDataset
ImdbDataset
ArrayDataset
GridDataset
PatchDataset
Image readers
Cross validation API

Savers
NiftiSaver
PNGSaver
CSVSaver

Losses
DiceLoss + extensions
FocalLoss TverskyLoss

Networks
UNet / DynUNet / VNet
DenseNet / SENets
SegResNet / AHNet
Common Blocks
Common Layers

Transforms
Spatial Transforms
Intensity Transforms
IO Transforms
Utility Transforms
Post Transforms

Inferers
SimpleInferer
SlidingWindowInferer

Visualize
Plot 3D / 2D images
GradCam feature map

Optimizers
Novograd, Layer wise LR

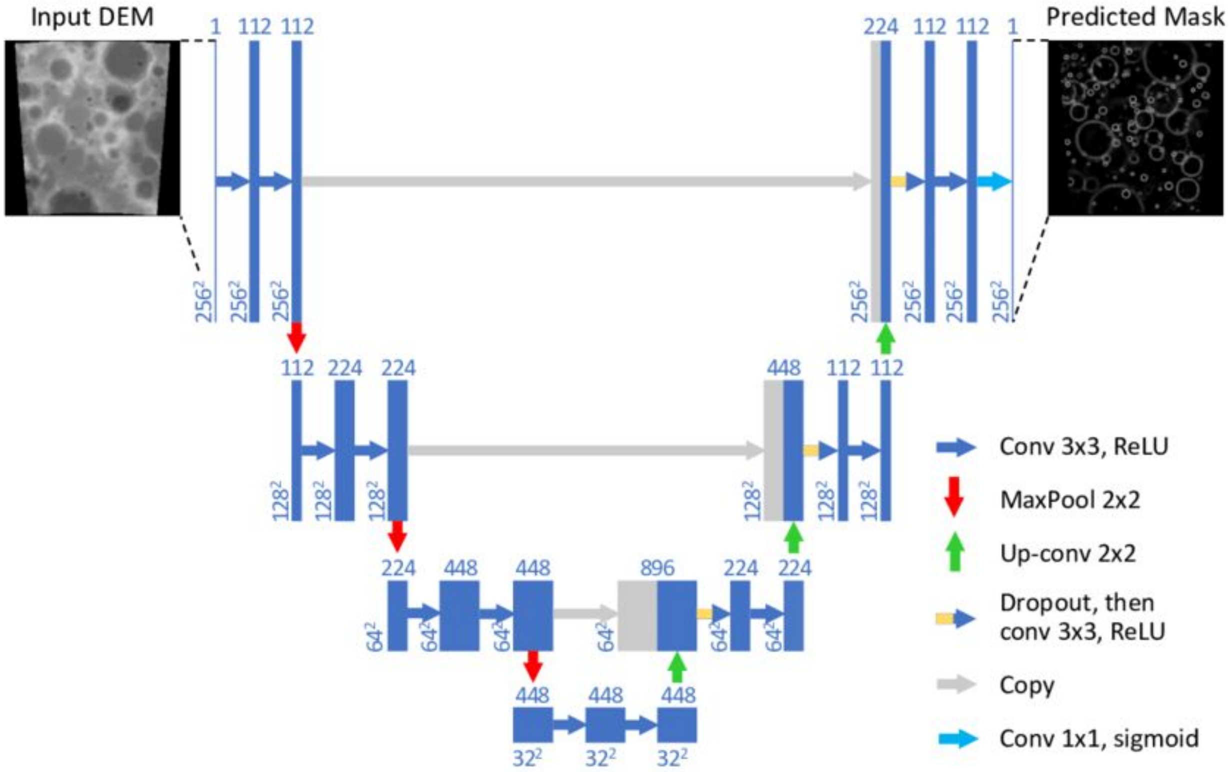
Metrics
MeanDice
Confusion matrix
Occlusion sensitivity
Surface distance
Hausdorff distance
ROCAUC

Kitware

Medical specific transforms

- LoadImage: Load medical specific formats file from provided path
- Spacing: Resample input image into the specified pixdim
- Orientation: Change the image's orientation into the specified axcodes
- RandGaussianNoise: Perturb image intensities by adding statistical noises
- NormalizeIntensity: Intensity Normalization based on mean and standard deviation
- Affine: Transform image based on the affine parameters
- Rand2DElastic: Random elastic deformation and affine in 2D
- Rand3DElastic: Random elastic deformation and affine in 3D

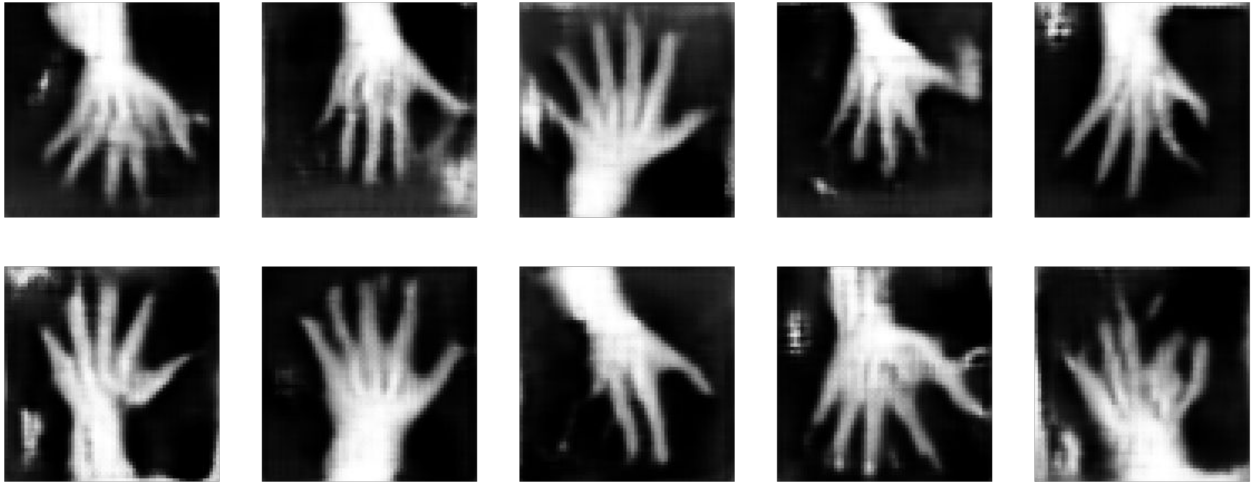
Models : UNET (1D, 2D, 3D)



Models : UNET (Code usage)

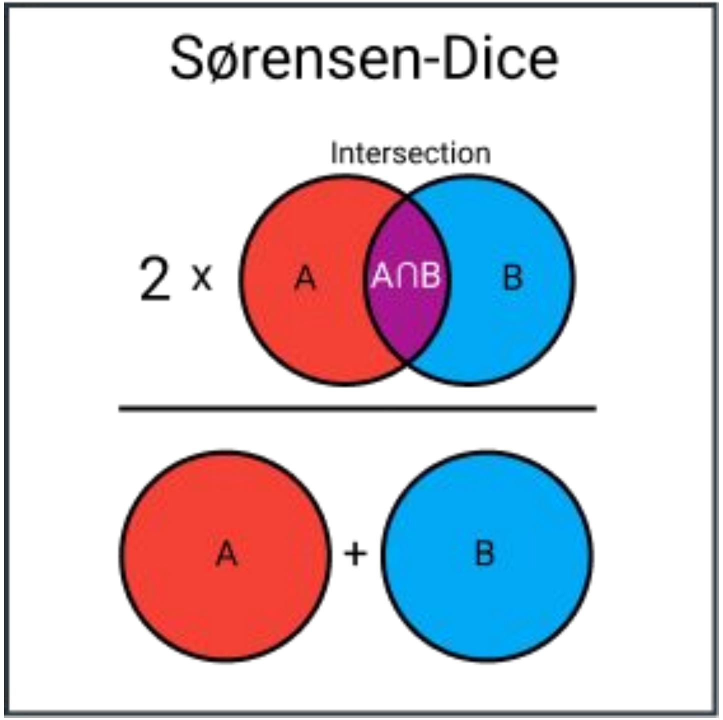
```
device = torch.device("cuda:0")
model = UNet(
    dimensions=3,
    in_channels=1,
    out_channels=2,
    channels=(16, 32, 64, 128, 256),
    strides=(2, 2, 2, 2),
    num_res_units=2,
    norm=Norm.BATCH,
).to(device)
```

Models : Generator / Discriminator

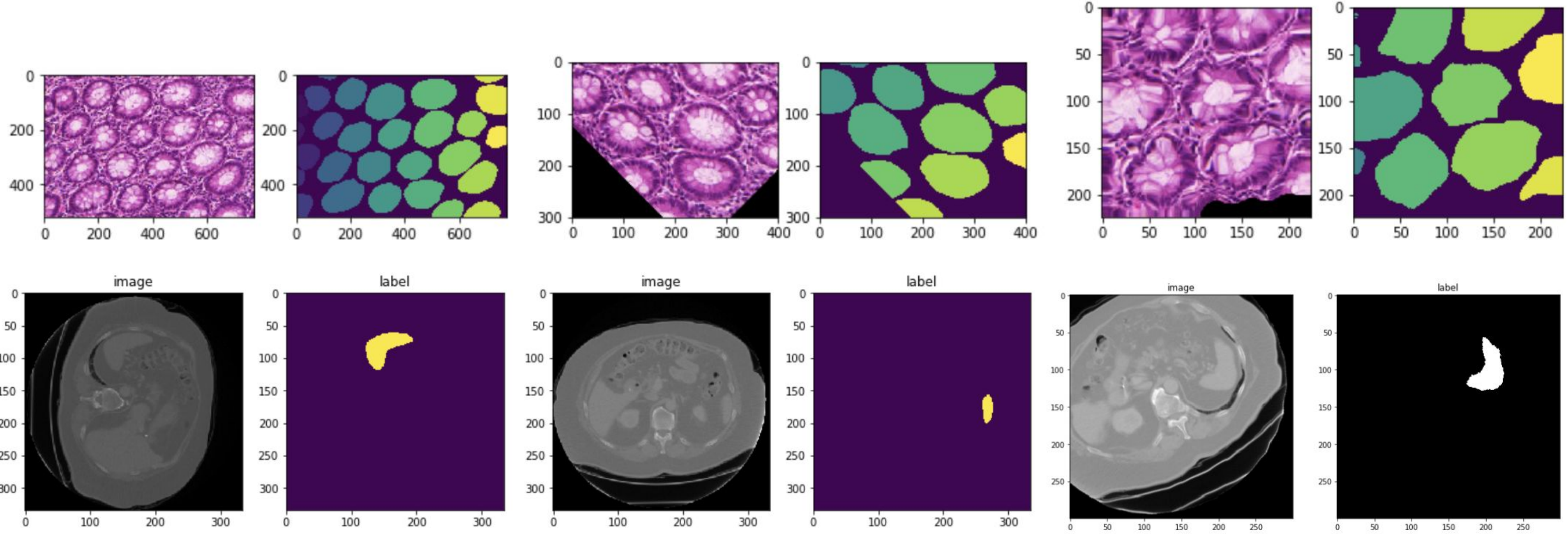


Loss functions : DiceLoss, FocalLoss, TverskyLoss

-



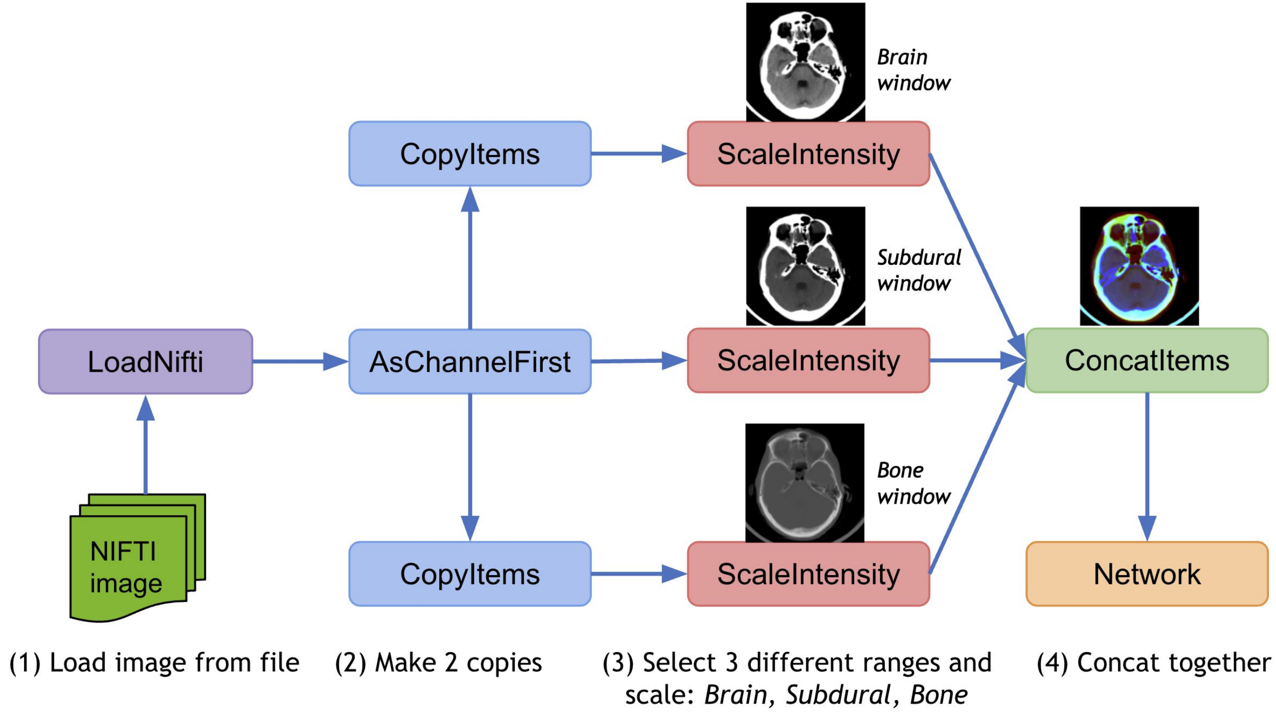
Data augmentation - 2D and 3D transforms



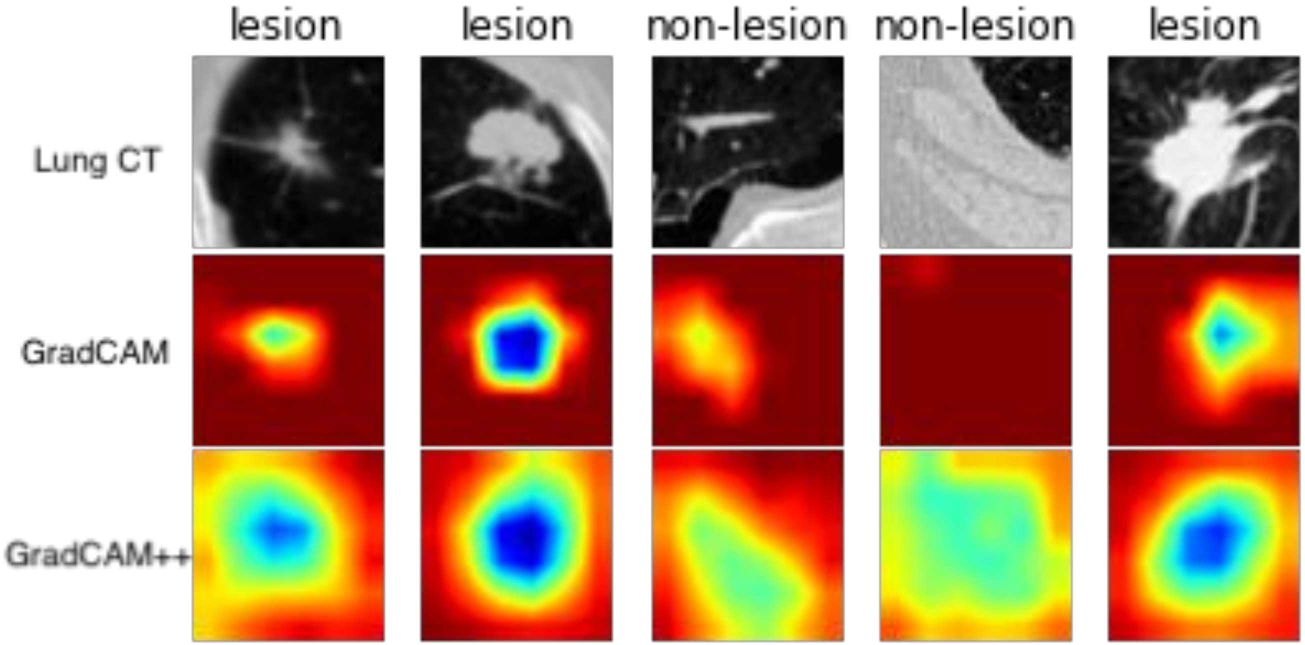
Post processing

(a) Original image (b) Segmentation result (c) Keep largest connected (d) Contour of regions (e) Map contour to image

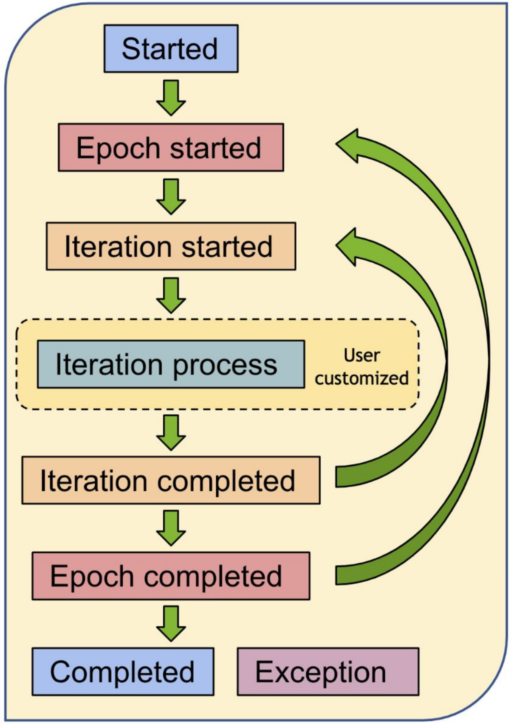
Multiple transform chains



Visualisation and interpretability



Workflows



Handlers	Started	Iteration completed	Epoch completed	Completed
Post Transforms		•		
MeanDice(metric)		•	•	
ROCAUC(metric)		•	•	
CheckpointLoader	•			
CheckpointSaver		•	•	•
ClassificationSaver		•		•
LrScheduleHandler		•	•	
SegmentationSaver		•		
StatsHandler		•	•	
TensorBoardHandler		•	•	
ValidationHandler		•	•	

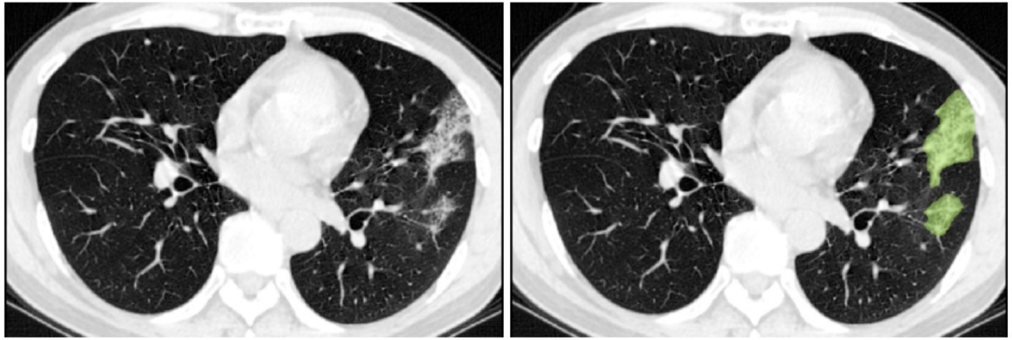
Research contributions

COPLE-Net for COVID-19 Pneumonia Lesion Segmentation

A reimplementation of the COPLE-Net originally proposed by:

G. Wang, X. Liu, C. Li, Z. Xu, J. Ruan, H. Zhu, T. Meng, K. Li, N. Huang, S. Zhang. (2020) “A Noise-robust Framework for Automatic Segmentation of COVID-19 Pneumonia Lesions from CT Images.” IEEE Transactions on Medical Imaging. 2020.

DOI: 10.1109/TMI.2020.3000314

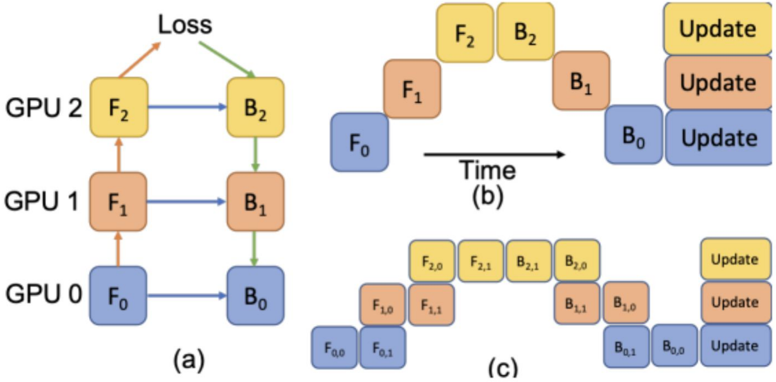


Research contributions

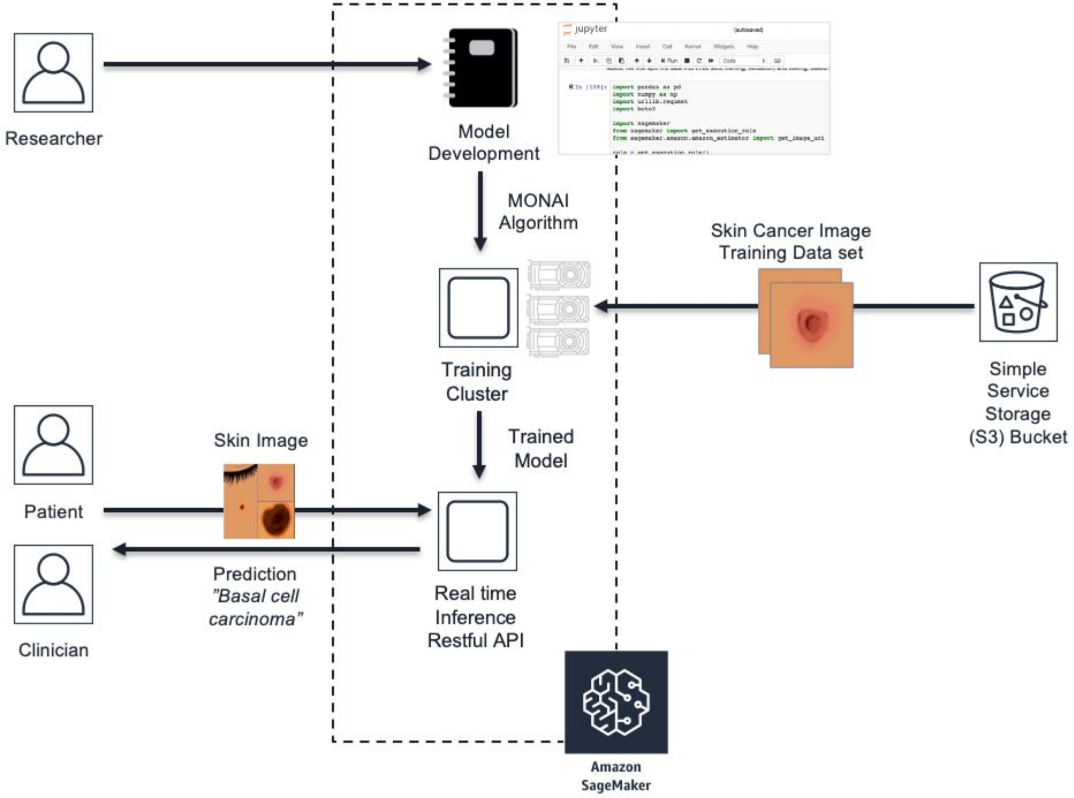
LAMP: Large Deep Nets with Automated Model Parallelism for Image Segmentation

A reimplementation of the LAMP system originally proposed by:

Wentao Zhu, Can Zhao, Wenqi Li, Holger Roth, Ziyue Xu, and Daguang Xu (2020) “LAMP: Large Deep Nets with Automated Model Parallelism for Image Segmentation.” MICCAI 2020 (Early Accept, paper link: https://arxiv.org/abs/2006.12575)



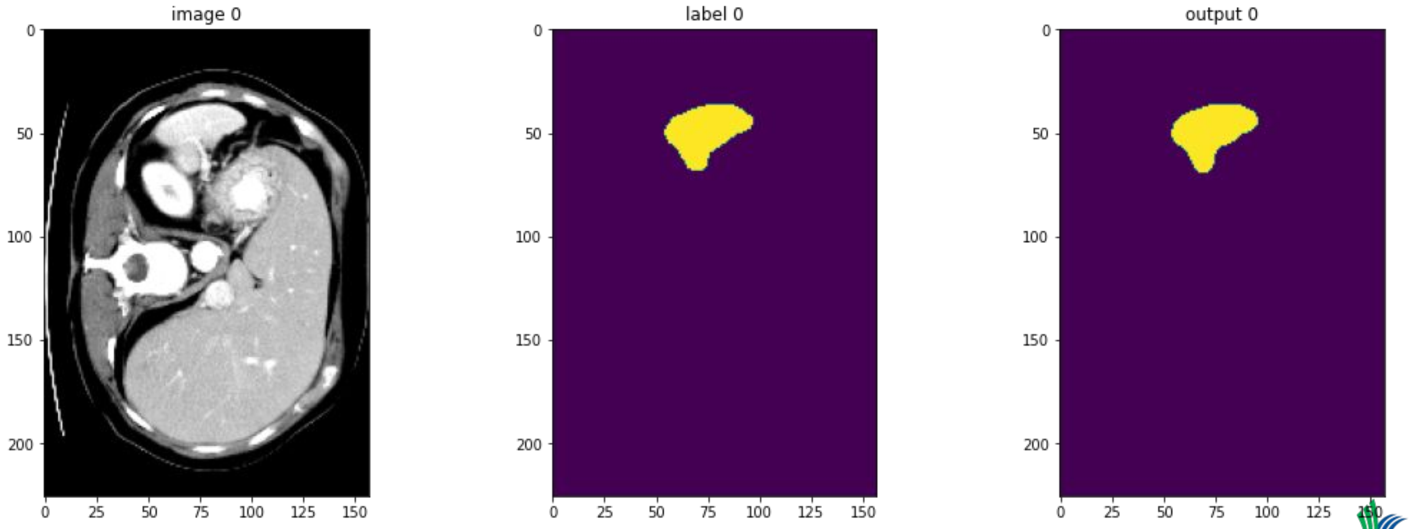
AWS SageMaker integration



Kitware

Tutorials

https://github.com/Project-MONAI/tutorials/blob/master/3d_segmentation/spleen_segmentation_3d.ipynb



Links

- Website: https://monai.io/
- API documentation: https://docs.monai.io
- Code: https://github.com/Project-MONAI/MONAI
- Project tracker: https://github.com/Project-MONAI/MONAI/projects
- Issue tracker: https://github.com/Project-MONAI/MONAI/issues
- Wiki: https://github.com/Project-MONAI/MONAI/wiki
- Test status: https://github.com/Project-MONAI/MONAI/actions
- Semantic segmentation data: http://medicaldecathlon.com/
- AWS Tutorial:
  https://aws.amazon.com/fr/blogs/industries/build-a-medical-image-analysis-pipeline-on-amazon-sagemaker-using-the-monai-framework/

Kitware

Questions ?