MONAI

Medical Open Network for AI

Thibault PELLETIER - thibault.pelletier@kitware.com

17/04/2023

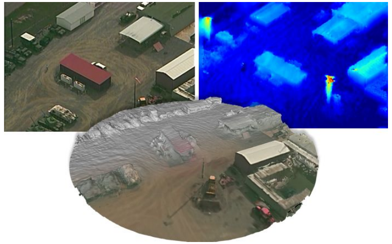
kitware

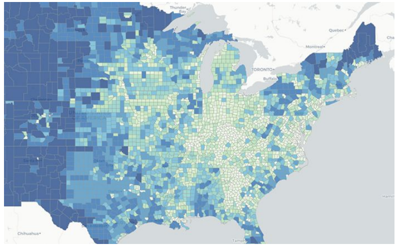
A word about me

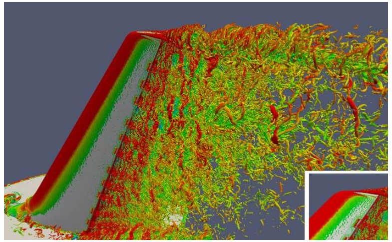
- Lead developer and MONAI instructor at Kitware EU since 2019
- 9 Years at ECA Robotics
- Double Masters from Arts et Metiers ParisTech and Lancaster University (UK)
- thibault.pelletier@kitware.com

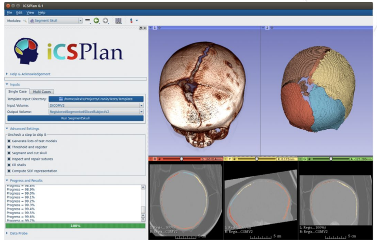
kitware

Kitware - Areas of expertise / Built on open source









Computer Vision

Data and Analytics

Scientific Computing

Medical Computing

Software Solutions

kitware

Kitware - Services

TRAINING

SUPPORT

DEVELOPMENT

GRANT COLLABORATION

Introduction

Medical Open Network for A. I. (MONAI)

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

- **Began as a collaboration between Nvidia and King's College London**
  - Prerna Dogra (Nvidia) and Jorge Cardoso (KCL)
- **Open Source: freely available and community-supported**
- **PyTorch-based**
- **Optimized for medical imaging**
- **Prioritizes reproducibility**

kitware

Why is MONAI Needed?

- Biomedical applications have specific requirements
  - Image modalities require specific processing methods: MRI, CT, etc.
  - Image formats require special support: DICOM, NIfTI, etc.
  - Image meta-data must be considered: voxel spacing, HU, etc.
- Certain network architectures are designed for, or are highly suitable for, biomedical applications
- Prioritization of capabilities is domain specific: sample size limitations, annotation uncertainties, … reproducibility

kitware

Why does MONAI emphasize reproducibility?

- **MONAI's focus on reproducibility**
  - Reduces code re-implementation (time and errors)
  - Provides baseline implementations (education and startup)
  - Demonstrates best practices for DL in medical image computing and computer-assisted interventions (quality)
  - Enables Open Science in DL for medicine (dissemination and impact)

What is MONAI?

MONAI Working Groups.

Imaging I/O

Focus: define how data is read into and written out from memory in MONAI.

Data

Focus: Defining support for bioinformatics, biomarkers, and metadata that are in scope for MONAI.

Transformations

Focus: Topics related to data preprocessing and augmentation modules in MONAI.

Federated Learning

Focus: Unify the disparate methods of Federated Learning in a common MONAI framework.

Evaluation, Reproducibility, and Benchmarking

Focus: Provide the infrastructure and tools for quality-controlled validation and benchmarking of medical image analytics methods.

Research

Focus: Establish MONAI as a catalyst for scientific progress and real-life impact.

Community Development

Focus: Establish MONAI as a common software foundation that the medical imaging research and development community can build upon.

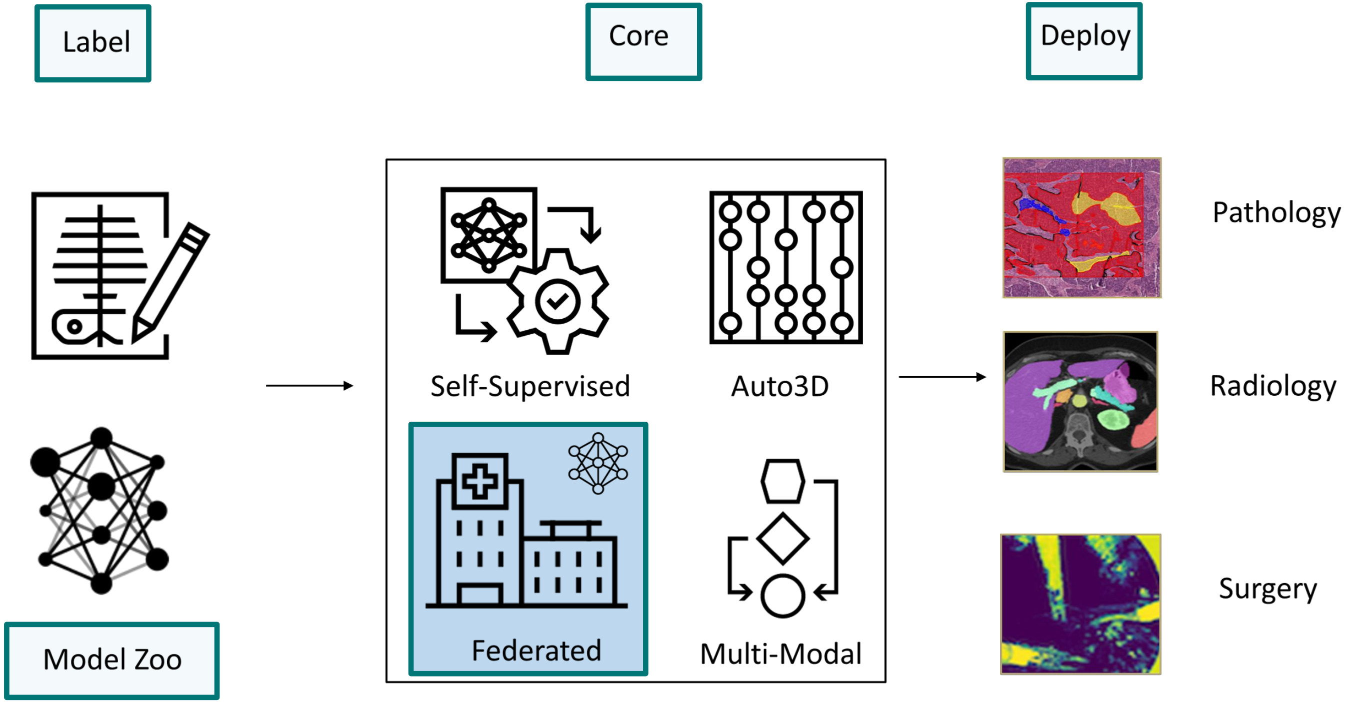
Deploy

Focus: Close the existing gap from research and development to clinical production environments by bringing AI models into the medical workflow.

Digital Pathology

Focus: Creating a standard pipeline for preprocessing, analysis, and visualization of pathology images.

What is MONAI?



What is MONAI Core?

MONAI Core - Built for customization and reproducibility

MONAI RESEARCH: *Implementations of state-of-the-art research publications*

- Multi-modality Support — *Radiogenomics*
- Unconstrained and Optimized Models — *Model Parallelism/Neural Archi. Search*
- End-to-end research lifecycle — *DICOM/HL7 FHIR/Model Exchange & Deploy*
- Comprehensive Decision-making — *COVID-19*

MONAI EXAMPLES: *Riche set of examples & demo notebooks to demonstrate the capabilities and integration with OSS*

- Segmentation
- Classification
- GANs & AutoEncoder
- Federated Learning
- Get Started Notebooks

MONAI WORKFLOWS: *Users can interface with MONAI workflows for ease of robust training & evaluation of Research Experiments*

- Engines — *SupervisedTrainer, SupervisedEvaluator*
- Event Handlers — *Checkpoint Loader; ValidationHandler; ClassificationSaver; CheckpointSaver; LrSchedulerHandler; StatsHandler; TensorBoardHandlers; SegmentationSaver; MetricLogger*
- Metrics — *MeanDice, ROCAUC*

FOUNDATIONAL COMPONENTS: *Users can integrate Independent domain specialized components into PyTorch Programs*

- Data — *CacheDataset, PersistentDataset, ZipDataset, ArrayDataset, GridDataset, EnhancedDataLoader*
- Savers & Writers — *Nifty, PNG & CSV*
- Inferers — *SimpleInferer, Slidingwindow*
- Losses — *DicesLoss & Extensions, FocalLoss, TverskyLoss*
- Visualize — *Plot 3D/2D images, Plot statistics curve*
- Networks — *UNET (2D & 3D); Layers & blocks; DenseNet(2D & 3D)*
- Metrics — *MeanDice, ROCAUC*
- Transforms — *Spatial, Intensity, IO, Utility, Post, Compose, 3rd Part adapter, BatchGenerator, Rising, TorchIO*

kitware

Data Augmentation and Pre-processing

- **Medical domain specific**
  - LoadImage
  - Spacing
  - Orientation
  - Ultrasound Linearization
- **Image transforms**
  - Blur
  - AddNoise
  - ITK Filters
  - Numpy Filters
  - …

```
[ ]  train_transforms = Compose([
         LoadPNG(image_only=True),
         AddChannel(),
         ScaleIntensity(),
         RandRotate(range_x=15, prob=0.5, keep_size=True),
         RandFlip(spatial_axis=0, prob=0.5),
         RandZoom(min_zoom=0.9, max_zoom=1.1, prob=0.5, keep_size=True),
         ToTensor()
     ])

     val_transforms = Compose([
         LoadPNG(image_only=True),
         AddChannel(),
         ScaleIntensity(),
         ToTensor()
     ])
```

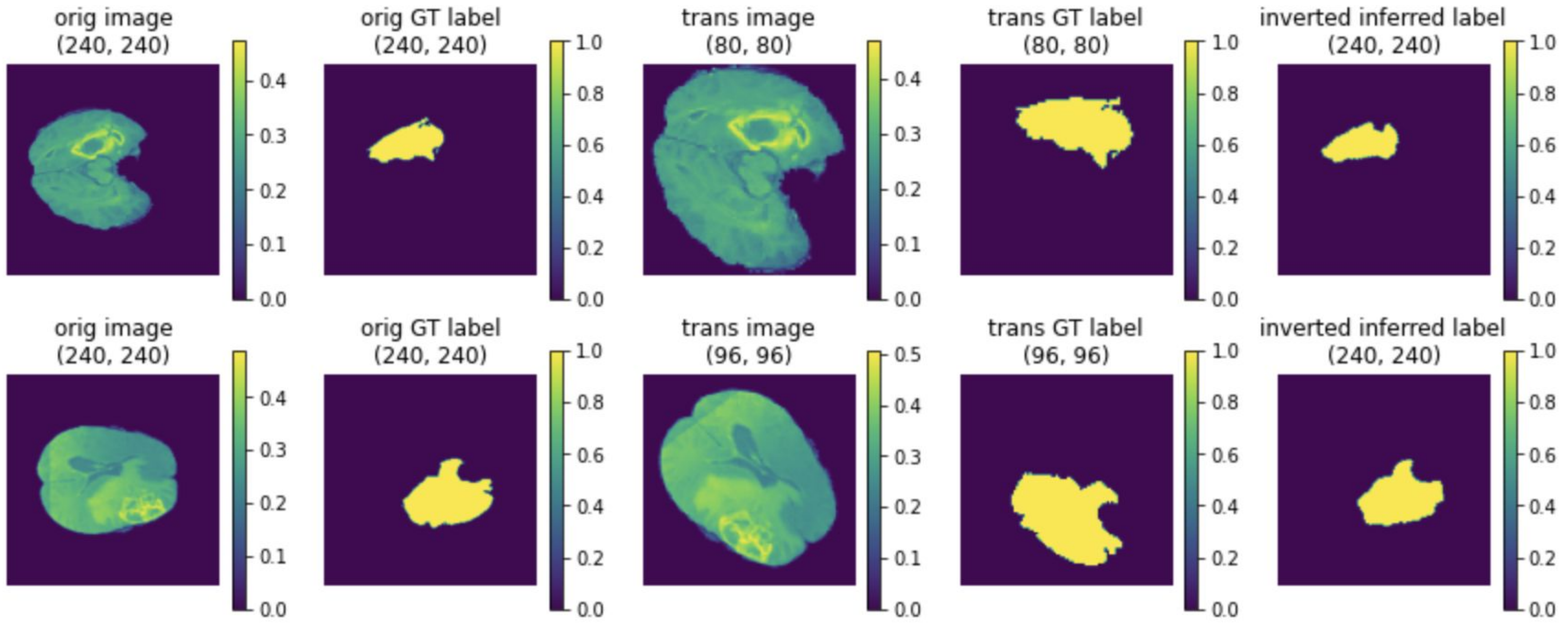
kitware

Invertible Transforms

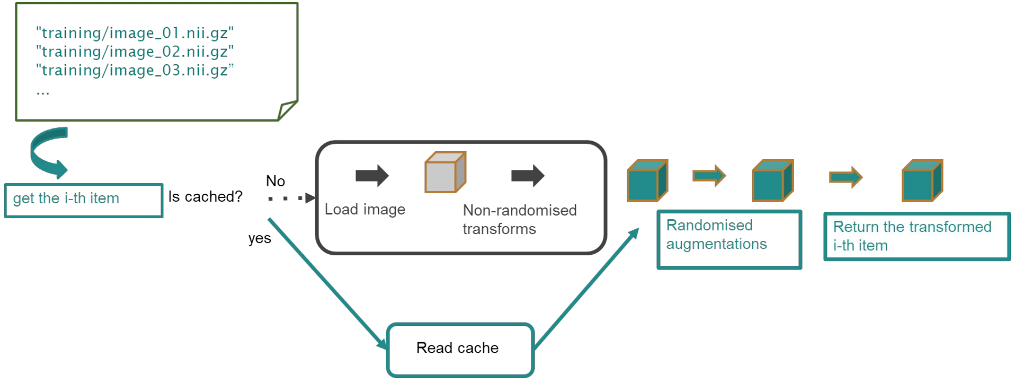
Why Invertible Transforms?

- Randomly augment the test case
- Track the transform parameters
- Run model inferences (segmentation)
- Resume to the original image space
- Compute ensemble/uncertainties

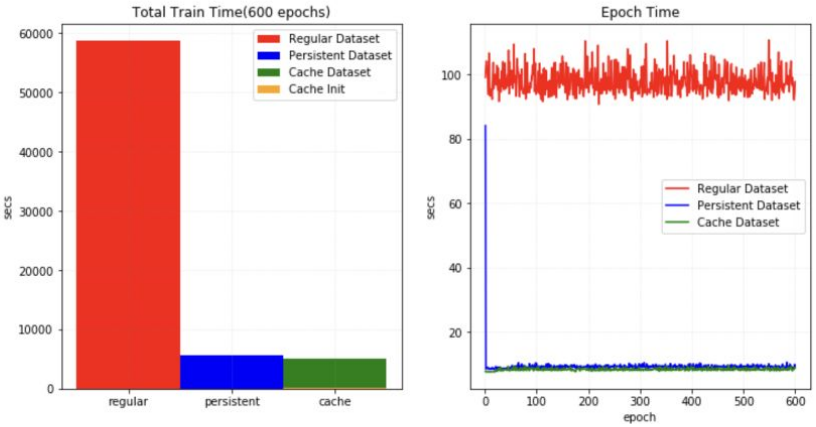
Invertible transforms



Dataset and Caching APIs.

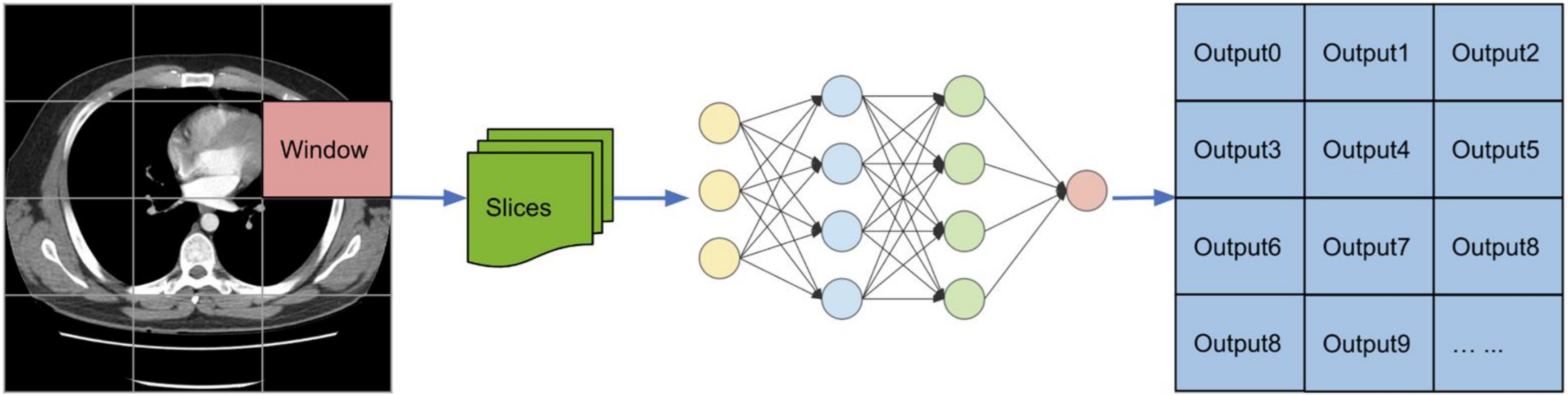


Caching Performance



kitware

Sliding Window Inference and Evaluation



(1) Generate slices from window (2) Construct batches (3) Execute on network (4) Connect all outputs

kitware

Metrics and Metrics APIs

Metrics

- Mean Dice
- Area under the ROC curve
- Confusion matrix
- Hausdorff distance
- Average surface distance
- Peak signal to noise ratio
- …

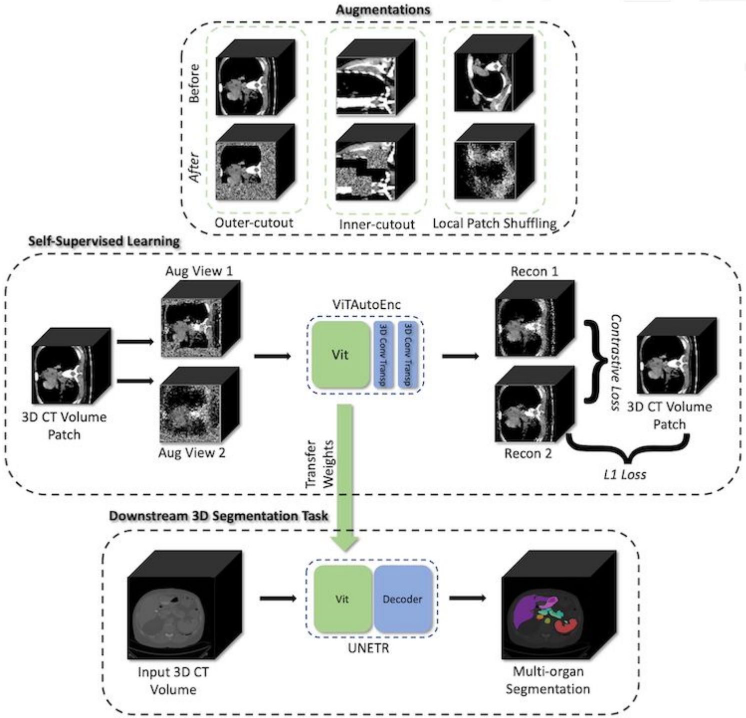
Metrics APIs

- Iterative Metric
- Cumulative
- Cumulative Average
- …

kitware

Network Architecture and Building Blocks

- Predefined Layers and Blocks
- Implementation of generic 2D and 3D networks
- Network adapter to finetune final layers
- State of the Art Architectures like: DiNTS, SSL, and Swin UNETR



MONAI Core Installation (Python)

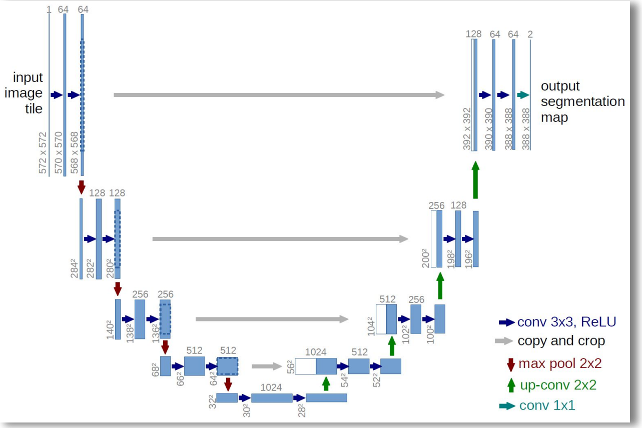
```
> pip install monai
```

```python
import monai
monai.config.print_config()
```

```
MONAI version: 0.3.0
Python version: 3.6.9 (default, Oct  8 2020, 12:12:24)  [GCC 8.4.0]
OS version: Linux (4.19.112+)
Numpy version: 1.18.5
Pytorch version: 1.7.0+cu101
MONAI flags: HAS_EXT = False, USE_COMPILED = False

Optional dependencies:
Pytorch Ignite version: 0.4.2
Nibabel version: 3.0.2
scikit-image version: 0.16.2
Pillow version: 7.0.0
Tensorboard version: 2.3.0
gdown version: 3.6.4
TorchVision version: 0.8.1+cu101
ITK version: 5.1.1
tqdm version: 4.51.0
```

Ease-of-use Example

```
net = monai.networks.nets.UNet(
    dimensions=2,              # 2 or 3 for a 2D or 3D network
    in_channels=1,             # number of input channels
    out_channels=1,            # number of output channels
    channels=[8, 16, 32],      # channel counts for layers
    strides=[2, 2]             # strides for mid layers
)
```



Access Medical Data

- **Goal: Harmonize and simplify open data and biomedical challenges**
  - Participate in / use public challenges
  - Define “challenges” (custom datasets) within your lab
- **Thin layer on top of PyTorch torch.data.utils.Dataset construct**
  - Automated (verified) download and unzip
  - Caching of data as well as intermediate results of preprocessing
  - Random splits of training, validation, and test

Access Medical Data

```python
from monai.apps import DecathlonDataset

dataset = DecathlonDataset(root_dir="./", task="Task05_Prostate", section="training", transform=None, download=True)
print(f"\nnumber of subjects: {len(dataset)}.\nThe first element in the dataset is {dataset[0]}.")
```

```
Task05_Prostate.tar: 100%|██████████| 229M/229M [03:15<00:00, 1.22MB/s]
Verified 'Task05_Prostate.tar.part', md5: 35138f08b1efaef89d7424d2bcc928db.
Verified 'Task05_Prostate.tar', md5: 35138f08b1efaef89d7424d2bcc928db.
Verified 'Task05_Prostate.tar', md5: 35138f08b1efaef89d7424d2bcc928db.
Load and cache transformed data: 100%|██████████| 26/26 [00:00<00:00, 196489.92it/s]
number of subjects: 26.
The first element in the dataset is {'image': 'Task05_Prostate/imagesTr/prostate_46.nii.gz', 'label': 'Task05_Prostate/label
```

Transforms for training and validation

```
[ ] train_transforms = Compose([
        LoadPNG(image_only=True),
        AddChannel(),
        ScaleIntensity(),
        RandRotate(range_x=15, prob=0.5, keep_size=True),
        RandFlip(spatial_axis=0, prob=0.5),
        RandZoom(min_zoom=0.9, max_zoom=1.1, prob=0.5, keep_size=True),
        ToTensor()
    ])

    val_transforms = Compose([
        LoadPNG(image_only=True),
        AddChannel(),
        ScaleIntensity(),
        ToTensor()
    ])
```

```
from monai.apps import DecathlonDataset

dataset = DecathlonDataset(root_dir="./", task="Task05_Prostate", section="training", transform=None, download=True)
print(f"\nnumber of subjects: {len(dataset)}.\nThe first element in the dataset is {dataset[0]}.")

Task05_Prostate.tar: 100%|██████████| 229M/229M [03:15<00:00, 1.22MB/s]
Verified 'Task05_Prostate.tar.part', md5: 35138f08b1efaef89d7424d2bcc928db.
Verified 'Task05_Prostate.tar', md5: 35138f08b1efaef89d7424d2bcc928db.
Verified 'Task05_Prostate.tar', md5: 35138f08b1efaef89d7424d2bcc928db.
Load and cache transformed data: 100%|██████████| 26/26 [00:00<00:00, 196489.92it/s]
number of subjects: 26.
The first element in the dataset is {'image': 'Task05_Prostate/imagesTr/prostate_46.nii.gz', 'label': 'Task05_Prostate/label
```

Random yet reproducible:

set_determinism(seed=XXXXXX)

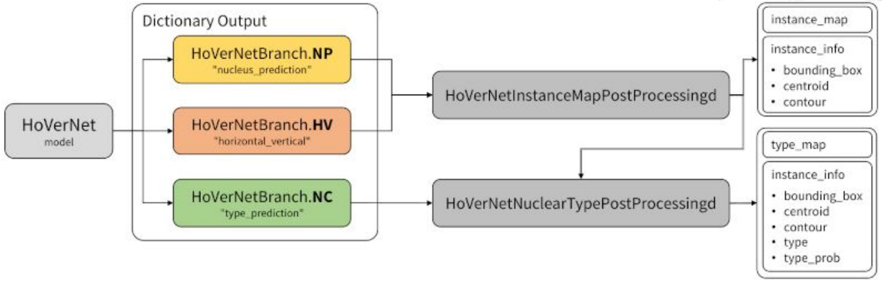
MONAI:End-End Training Workflow in ~10 Lines of Code

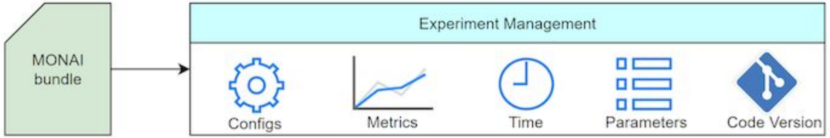
```
from monai.application import MedNISTDataset
from monai.data import DataLoader
from monai.transforms import LoadPNGd, AddChanneld, ScaleIntensityd, ToTensord, Compose
from monai.networks.nets import densenet121
from monai.inferers import SimpleInferer
from monai.engines import SupervisedTrainer

transform = Compose(
       [
              LoadPNGd(keys="image"),
              AddChanneld(keys="image"),
              ScaleIntensityd(keys="image"),
              ToTensord(keys=["image", "label"])
       ]
)

dataset = MedNISTDataset(root_dir="./", transform=transform, section="training", download=True)

trainer = SupervisedTrainer(
       max_epochs=5,
       train_data_loader=DataLoader(dataset, batch_size=2, shuffle=True, num_workers=4),
       network=densenet121(spatial_dims=2, in_channels=1, out_channels=6),
       optimizer=torch.optim.Adam(model.parameters(),lr=1e-5),
       loss_function=torch.nn.CrossEntropyLoss(),
       inferer=SimpleInferer()
)

trainer.run()
```

MONAI Core v1.1

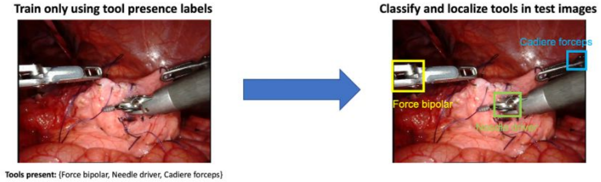
Latest Release

- Digital pathology workflows
- Experiment management for MONAI bundle
- Auto3dSeg enhancements
- New models in MONAI Model Zoo
- State-of-the-art SurgToolLoc solution



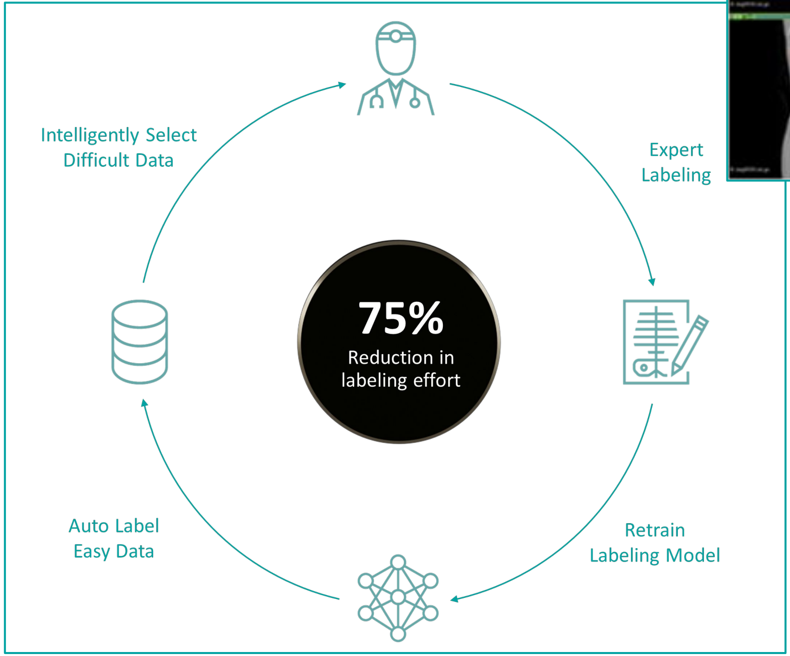


Surgical Tool Localization in endoscopic videos



What is MONAI Label?

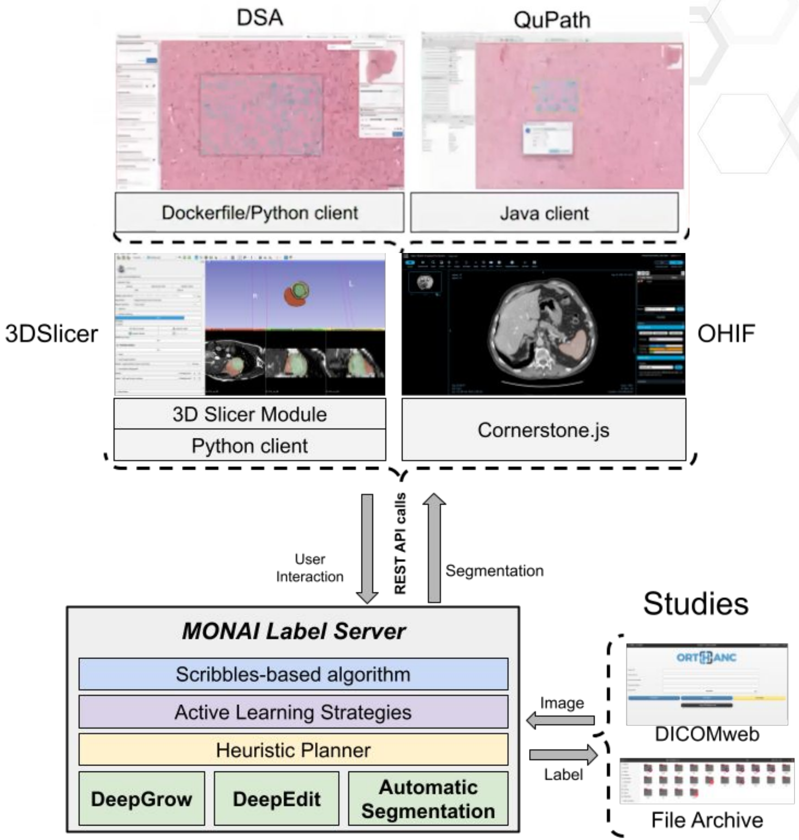
MONAI Label - AI-Assisted Annotation (AIAA)



MONAI Label Infrastructure.

Three Main Parts:
server-client system

- **MONAI Label Server**
- **Client / GUIs**
- **Datastore**



kitware

Why MONAI Label?

For Clinician

Radiology: X-Ray, CT, and MRI
Pathology: Whole Slide Images

Viewer Integration

Existing viewer integration with common applications in both radiology and pathology workflow including 3D Slicer and DSA.

Multiple Annotation Methods

Start by using traditional annotation methods like Scribbles or use an interactive algorithm like DeepEdit.

Sample Apps and Pretrained Models

MONAI Label includes sample applications for both radiology and pathology. You can also use the our pretrained models or start from scratch.

For Researcher and Data Scientists

Quickly get started with a common framework

Rapid App Prototyping

Use a sample app to jumpstart the development of your own custom labeling app.

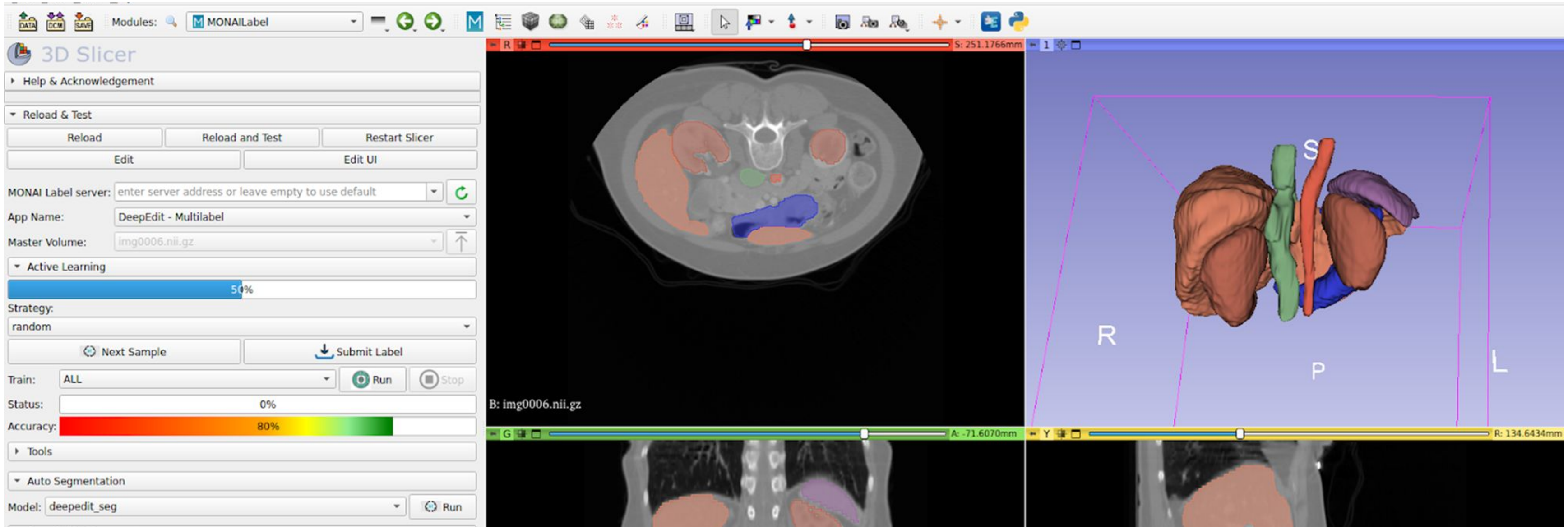
Active Learning Techniques

Use existing Active Learning strategies or implement your own.

Easy Integration

MONAI Label exposes a REST API that you can use to integrate in to your own viewer or workflow.

MultiLabel DeepEdit



kitware

MONAI Label Live Demo

kitware

MONAI Label v0.6

Latest Release

- Pathology Improvements
- QuPath Improvements
- Experiment Management
- 3D Slicer: Detection model support in MONAI Bundle App for Radiology use-case
- Multi-GPU/Multi-Threaded support for Batch Inference

What is MONAI Deploy?

MONAI Deploy - Packaging and deployment

- Aims to become the standard for packaging, testing, deploying and running medical AI applications in clinical production
- Creates a set of intermediate steps where researchers and physicians can build confidence in the techniques and approaches used with AI

Key features

- **MONAI Application Package (MAP)**
  - Defines how applications can be packaged. and distributed amongst MONAI Working Group member organizations.
- **MONAI Deploy App SDK**
  - Set of development tools to create MAPs out of MONAI / Pytorch models.
- **MONAI Deploy Informatics Gateway**
  - I/0 for DICOM and Fast Healthcare Interoperability Resources (FHIR) .
- **MONAI Deploy Workflow Manager**
  - Orchestrates what has to be executed based on the clinical workflow specification and incoming requests.
- **MONAI Deploy Express**
  - End-to-end pipeline for testing and validation of MONAI Applications (MAPs).

MONAI Deploy v0.5

Latest Release

- **App SDK compatible with MONAI v0.9.1 and later**
- **Additional DICOM support**
  - DICOM Encapsulated PDF Writer.
  - DICOM Segmentation Writer
  - …
- **Updated tutorials and notebooks**

Walkthrough

https://github.com/Project-MONAI/monai-bootcamp

MONAI Resources

- MONAI Website: https://monai.io/
- MONAI Slack: https://forms.gle/QTxJq3hFictp31UM9
- MONAI Docs:
  - MONAI Core: https://docs.monai.io/en/stable/
  - MONAI Label: https://docs.monai.io/projects/label/en/latest/index.html
  - MONAI Deploy App SDK: https://docs.monai.io/projects/monai-deploy-app-sdk/en/latest/
- MONAI Github: https://github.com/Project-MONAI
  - MONAI Core: https://github.com/Project-MONAI/MONAI
  - MONAI Label: https://github.com/Project-MONAI/MONAILabel
  - MONAI Deploy: https://github.com/Project-MONAI/monai-deploy
- MONAI YouTube: https://www.youtube.com/c/Project-MONAI
  - Overview Videos, Deep Dive Series, Bootcamp and other event recordings
- MONAI Twitter: https://twitter.com/ProjectMONAI
  - Follow for the latest announcements
- MONAI Medium: https://monai.medium.com/
  - Read about our latest releases and our upcoming research interview series

Questions