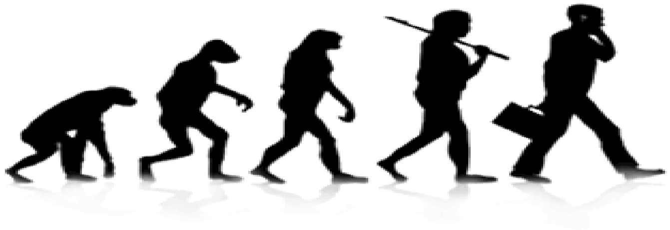
Medical Imaging with Deep Learning in MATLAB

Advanced Techniques for Classification and Segmentation

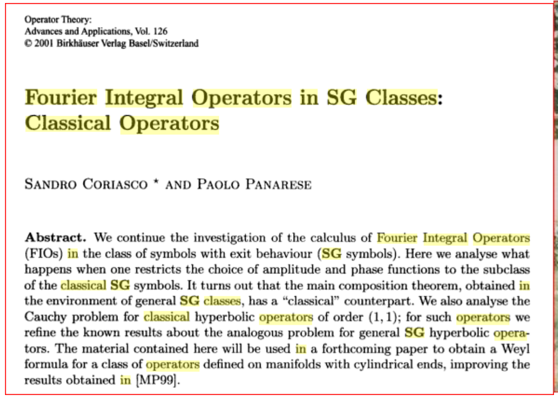
Dr Paolo Panarese, Math PhD
EDU Customer Success Engineer
Academia Team, MathWorks
8 November 2024

What about me?



Education → **Work experience**

1990-1994	1995-1999	1999-2001	2002-2021	2021-2024
Univ. of Torino M.Sc. in Maths	*Univ. of Bologna* Ph.D in Maths	*Univ. of Torino* Researcher	*MathWorks* Trainer	*MathWorks* EDU CSE

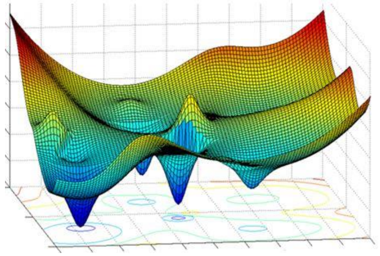


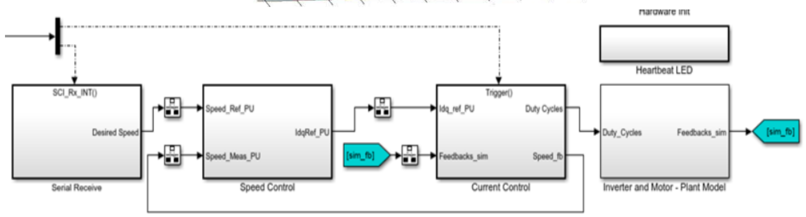
Operator Theory:
Advances and Applications, Vol. 126
© 2001 Birkhäuser Verlag Basel/Switzerland

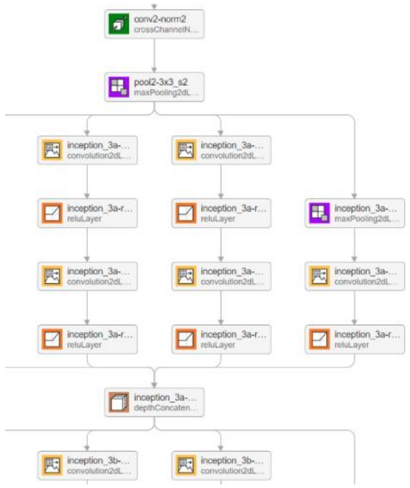
Fourier Integral Operators in SG Classes: Classical Operators

SANDRO CORIASCO * AND PAOLO PANARESE

Abstract. We continue the investigation of the calculus of Fourier Integral Operators (FIOs) in the class of symbols with exit behaviour (SG symbols). Here we analyse what happens when one restricts the choice of amplitude and phase functions to the subclass of the classical SG symbols. It turns out that the main composition theorem, obtained in the environment of general SG classes, has a "classical" counterpart. We also analyse the Cauchy problem for classical hyperbolic operators of order (1, 1); for such operators we refine the known results about the analogous problem for general SG hyperbolic operators. The material contained here will be used in a forthcoming paper to obtain a Weyl formula for a class of operators defined on manifolds with cylindrical ends, improving the results obtained in [MP99].





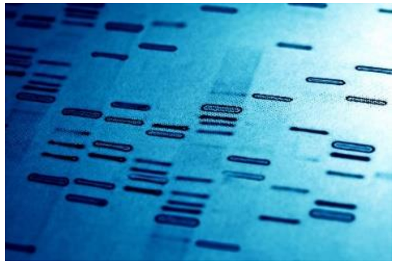


MATLAB is used in many different industries

Aerospace and Defense

Automotive

Biological Sciences



Biotech and Pharmaceutical

Communications

Electronics

Energy Production

Financial Services

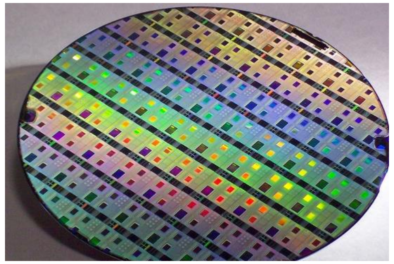
Industrial Machinery

Medical Devices

Metals, Materials, Mining

Neuroscience

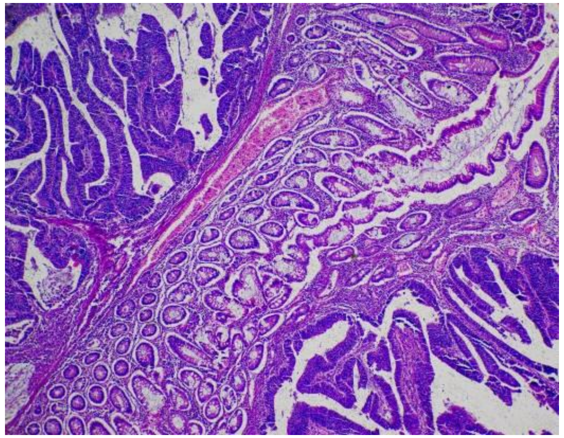
Railway Systems



Semiconductors

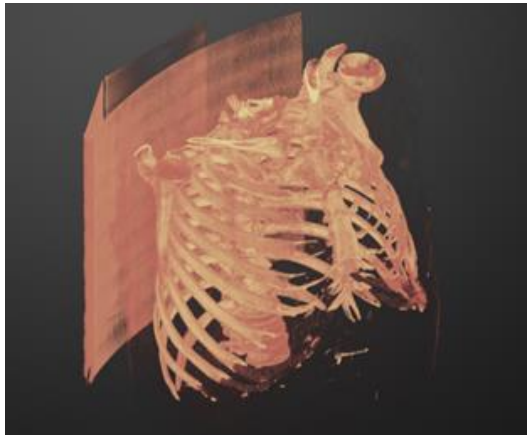
Software and Internet

Medical imaging a core part of several workflows

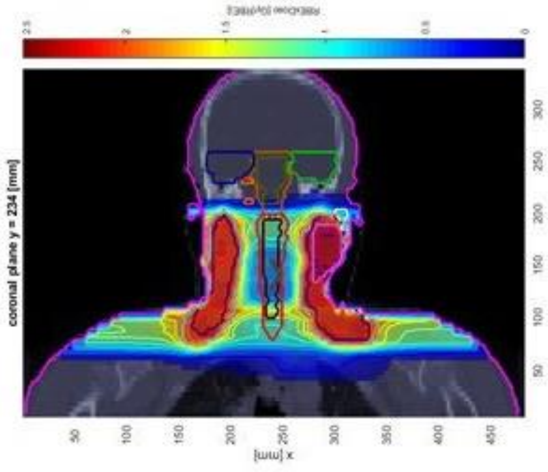


Digital Pathology

Ophthalmology/OCT

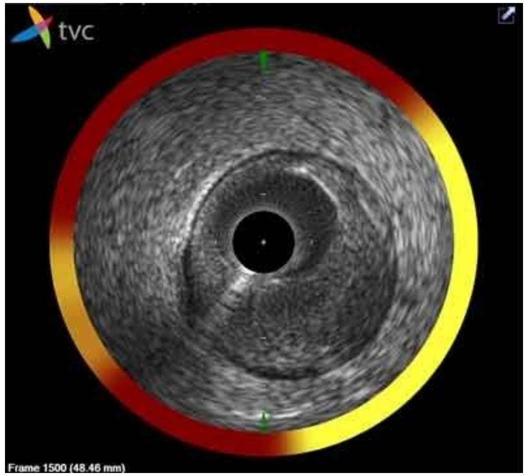


Radiology (MRI, US, X-ray, CT)



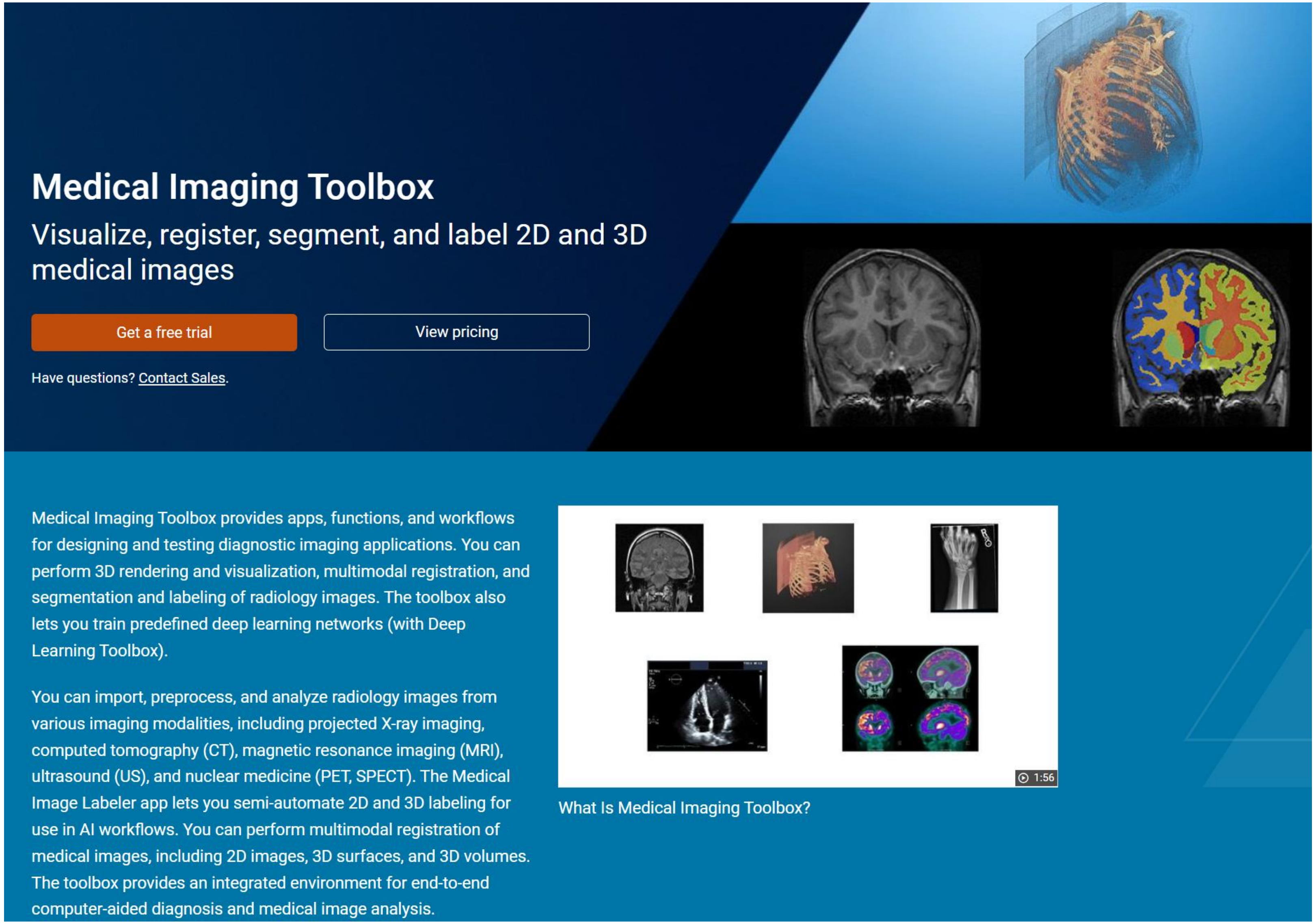
Radiotherapy Planning

Endoscopy



Intravascular Imaging

Medical Imaging Toolbox



Medical Imaging Toolbox

Visualize, register, segment, and label 2D and 3D medical images

Get a free trial | View pricing

Have questions? Contact Sales.

Medical Imaging Toolbox provides apps, functions, and workflows for designing and testing diagnostic imaging applications. You can perform 3D rendering and visualization, multimodal registration, and segmentation and labeling of radiology images. The toolbox also lets you train predefined deep learning networks (with Deep Learning Toolbox).

You can import, preprocess, and analyze radiology images from various imaging modalities, including projected X-ray imaging, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound (US), and nuclear medicine (PET, SPECT). The Medical Image Labeler app lets you semi-automate 2D and 3D labeling for use in AI workflows. You can perform multimodal registration of medical images, including 2D images, 3D surfaces, and 3D volumes. The toolbox provides an integrated environment for end-to-end computer-aided diagnosis and medical image analysis.

1:56

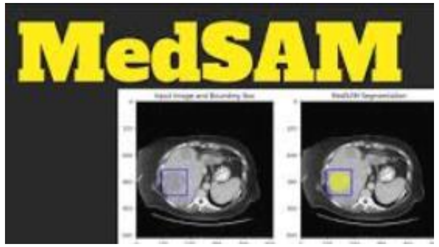
What Is Medical Imaging Toolbox?

https://www.mathworks.com/products/medical-imaging.html

We are going to learn about …

MATLAB makes Medical Imaging easier

- Interactive apps for Registration, Image Labeling
- Integration of medical models for Image segmentation



Examples:

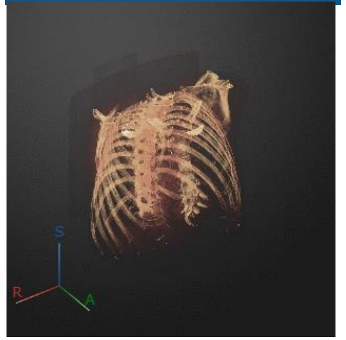
1. Classification of Tumors using Radiomics features
2. Segmentation of Brain MRI using a MONAI model
3. Segmentation of CT scans using different models of MONAI
4. Segmentation of heart ROI using MedSAM

Flexible medical imaging workflow

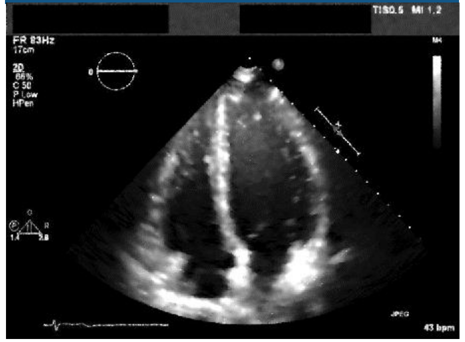
Import/Interoperate

```
medImage =
  medicalImage with properties:

          Pixels: [1540x1250 uint16]
        Colormap: []
    SpatialUnits: "mm"
       FrameTime: []
       NumFrames: 1
    PixelSpacing: [0.1390 0.1390]
        Modality: 'DX'
    WindowCenter: 2048
     WindowWidth: 4096
```

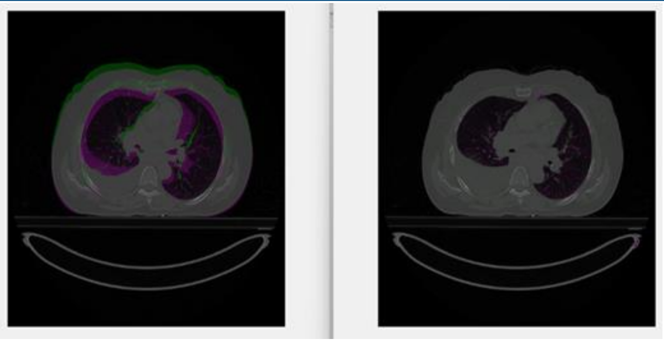
Visualize



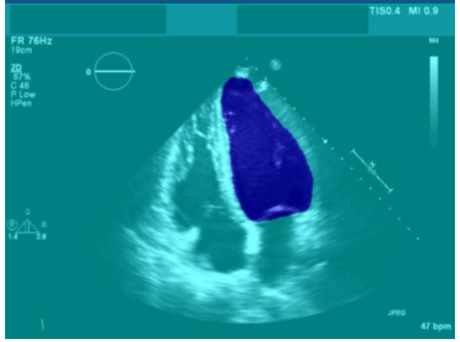
Preprocess



Register



Segment



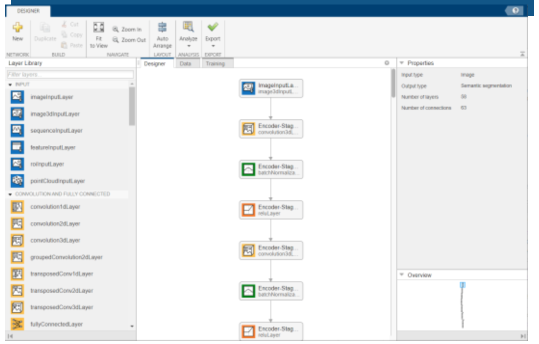
Label

Data Source

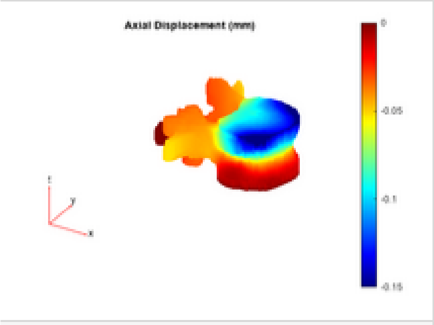
Label Image

+ Human Validation = Ground Truth

Train



Analyze



Dedicated functions to load medical images

Import

```
medImage = medicalImage("forearmXrayImage1.dcm”)

medImage =
  medicalImage with properties:

           Pixels: [1540x1250 uint16]
         Colormap: [] SpatialUnits: "mm"
        FrameTime: []
        NumFrames: 1
     PixelSpacing: [0.1390 0.1390]
         Modality: 'DX’
     WindowCenter: 2048
      WindowWidth: 4096

imshow(medImage.Pixels,[])
```

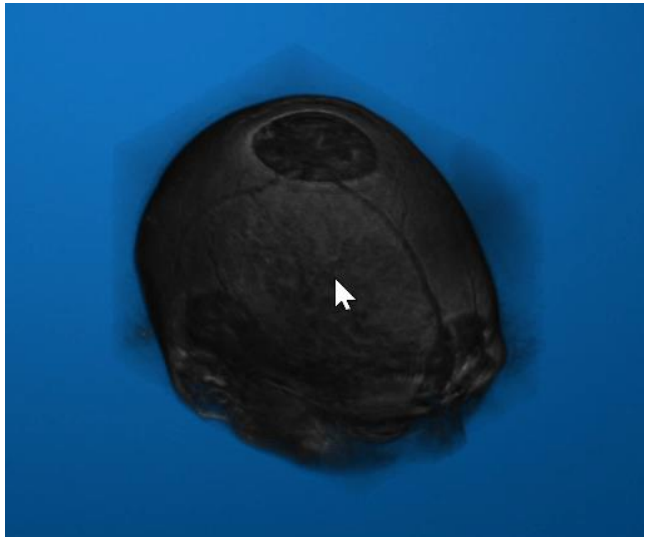
Note: you can create a medical image object from an image datastore

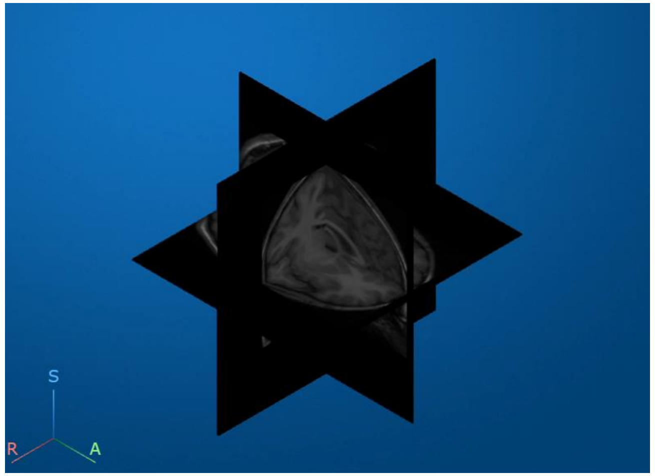
```
dicomds = imageDatastore(dataFolder, FileExtensions=".dcm",ReadFcn=@(x) dicomread(x));
medImage = medicalImage(dicomds)
```

Dedicated functions to load medical volumes

Import

```
medVolMRI = medicalVolume("brainSegData_anat.nii.gz")

medVolMRI =
  medicalVolume with properties:

                 Voxels: [256×256×128 int16]
         VolumeGeometry: [1×1 medicalref3d]
           SpatialUnits: "mm"
            Orientation: "coronal"
           VoxelSpacing: [0.9375 0.9375 1.5000]
           NormalVector: [0 -1 0]
       NumCoronalSlices: 128
      NumSagittalSlices: 256
    NumTransverseSlices: 256
           PlaneMapping: ["sagittal"    "transverse"    "coronal"]
               Modality: "unknown"
          WindowCenters: 0
           WindowWidths: 0

volshow(medVol)
volshow(medVolMRI,RenderingStyle="SlicePlanes")
```

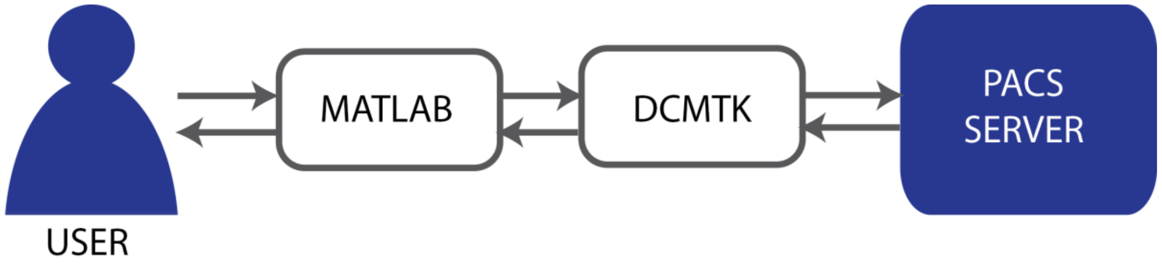


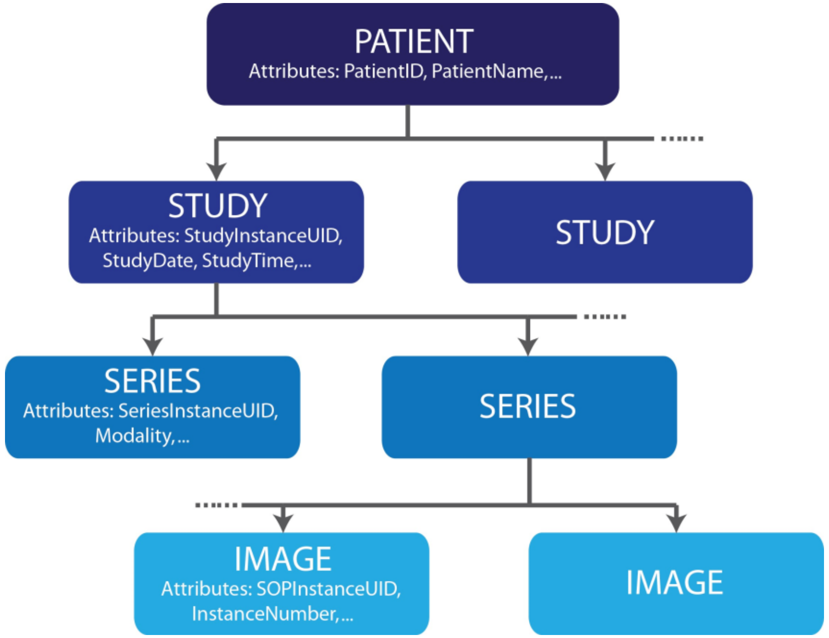


Manages spatial referencing information: intrinsic image coordinates/patient coordinate systems, anatomical planes

Dedicated functions to connect with PACS servers (Picture Archiving and Communication System)

Import





- `testConnection` - Test PACS server connection
- `dicomstore` - store DICOM images to PACS server
- `dicomquery` - Query attributes of DICOM images
- `dicomget` - Retrieve DICOM images from PACS server

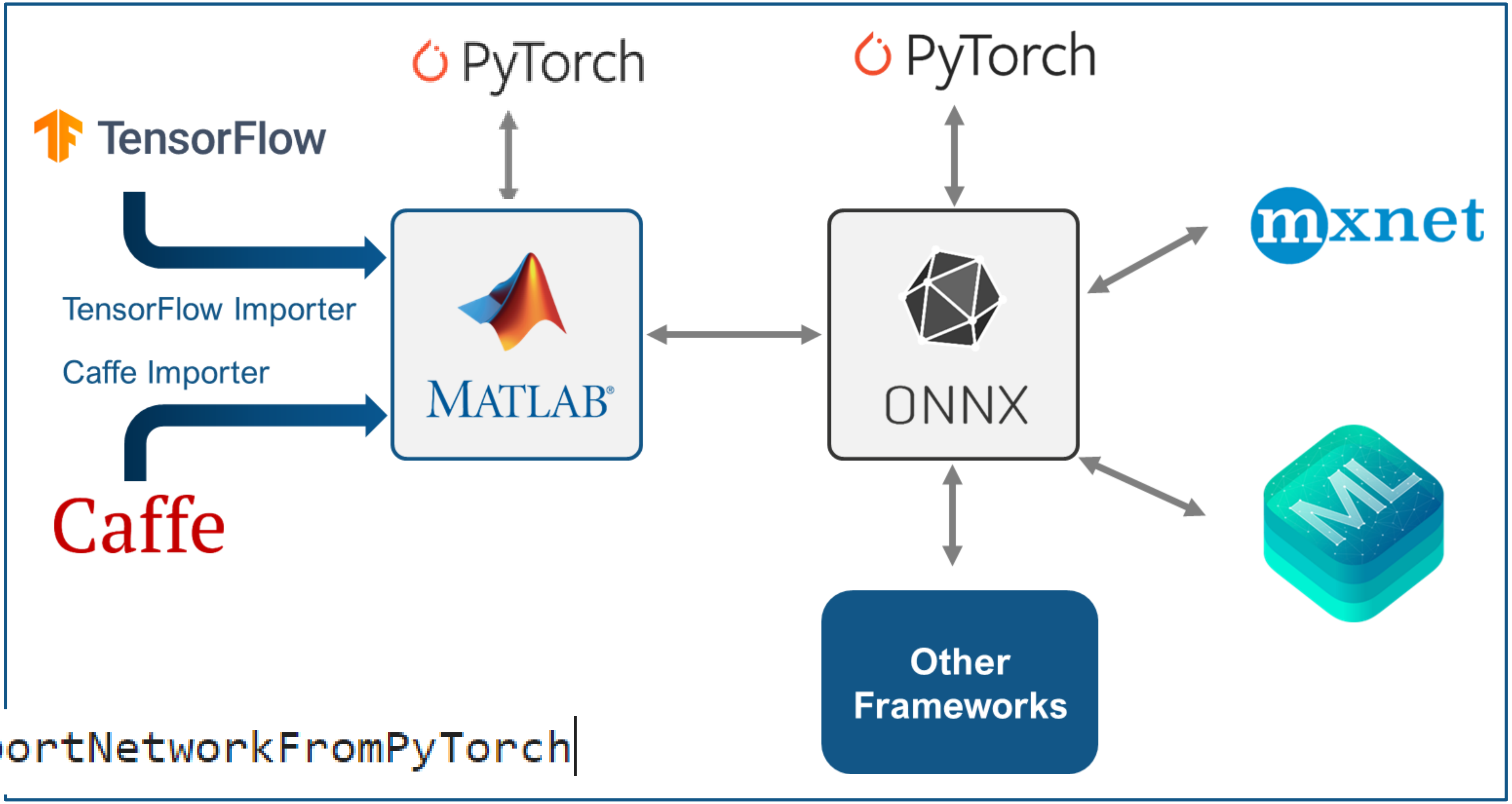
https://www.mathworks.com/help/medical-imaging/ug/working-with-pacs-server-for-dicom-image-retrieval.html

MathWorks

TensorFlow & PyTorch Interoperability

Interoperate

Import and export from deep learning frameworks

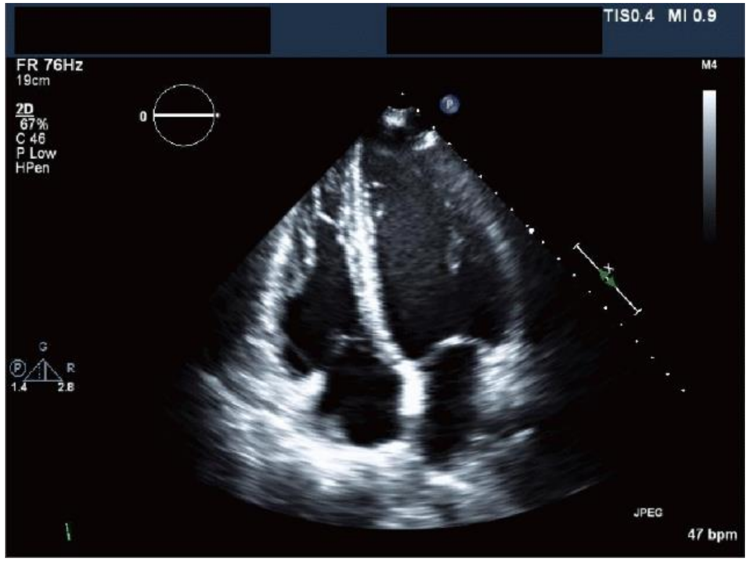


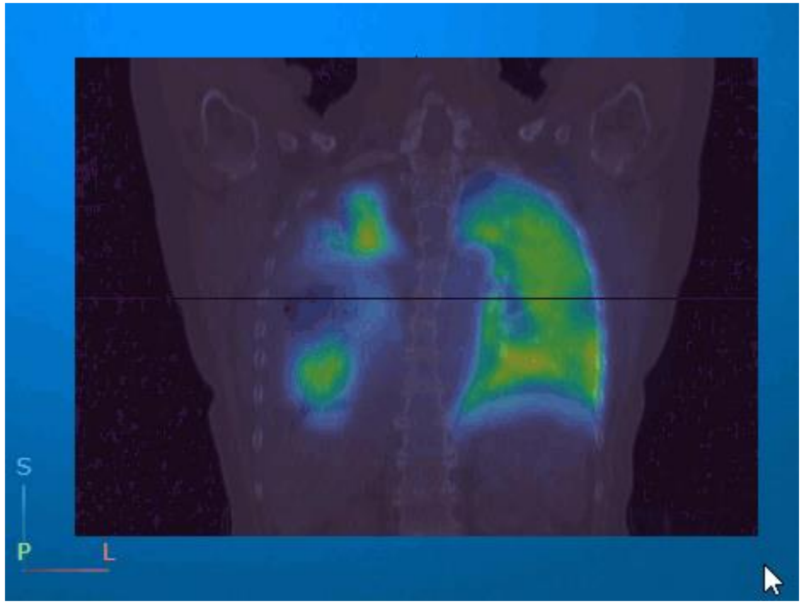
NEW

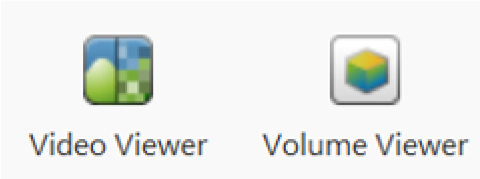
```
>> importNetworkFromPyTorch
```

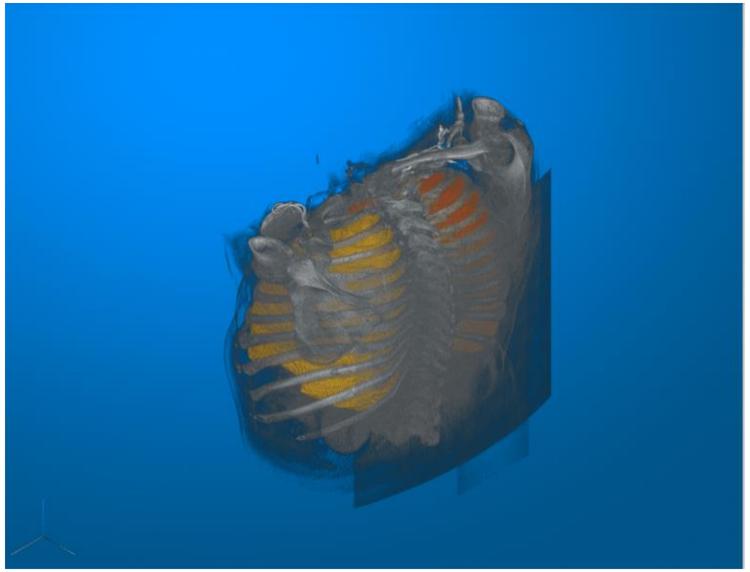
Visualize 2d, 2d+time, and 3d multimodal medical image data

Visualize









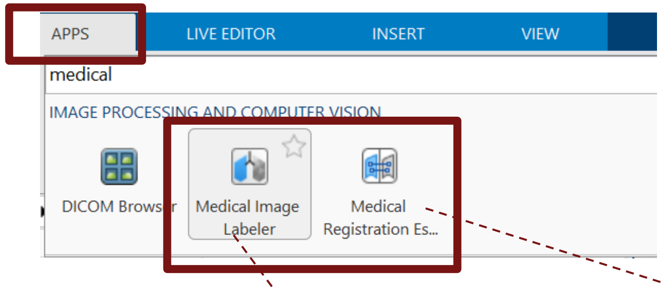
```
imshow
volshow
implay
montage
volumeViewer
medicalImageLabeler …
```

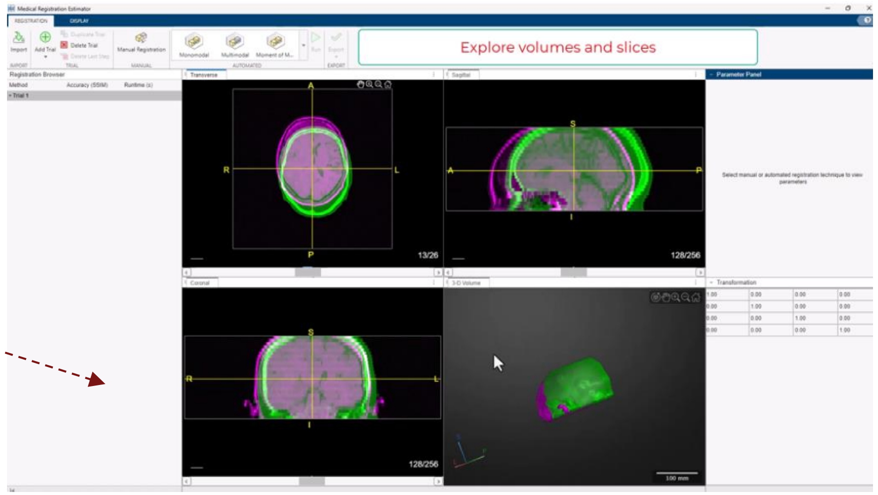
Medical Apps facilitate registration and segmentation

Visualize

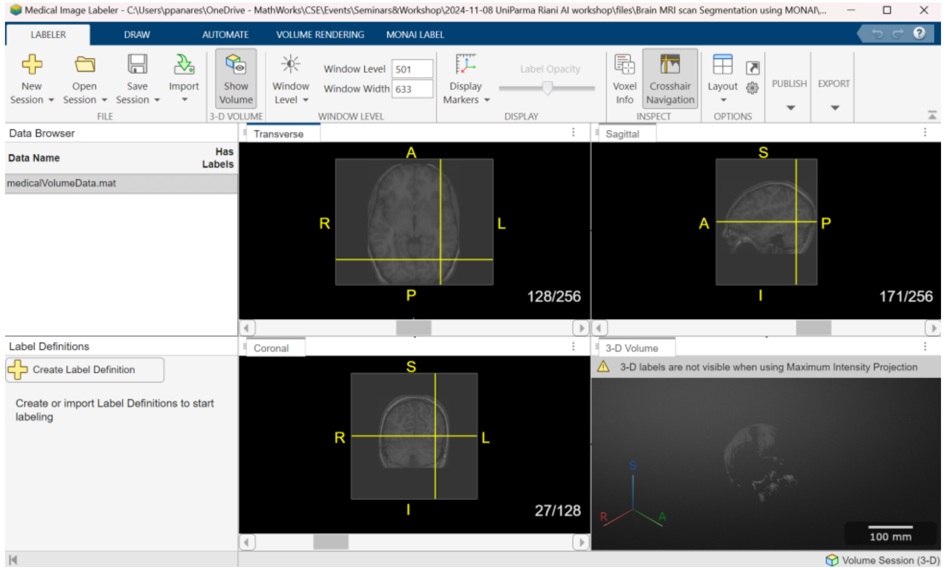
Preprocess

Segment





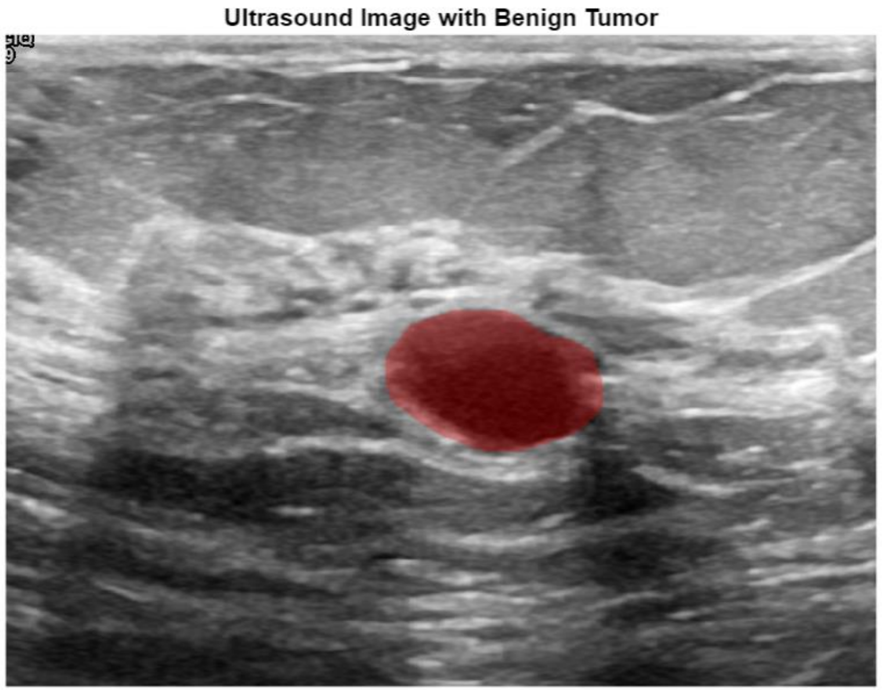
Medical Registration Estimator app



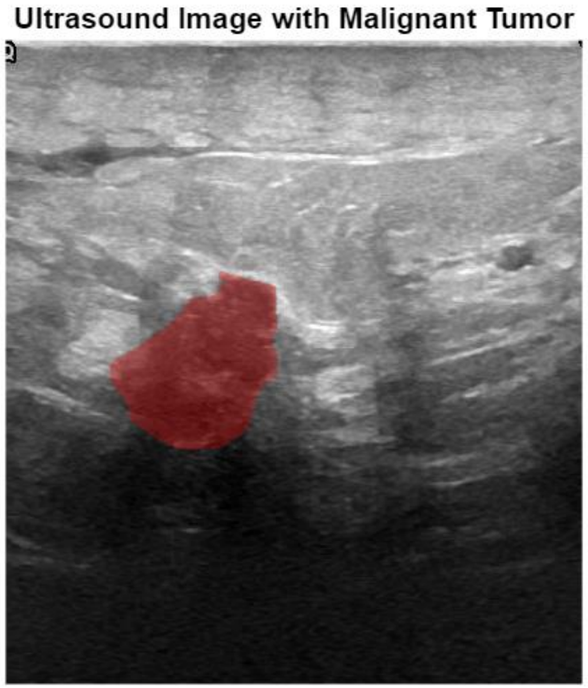
Medical Image Labeler app

Example 1:
Classification of Breast Tumors using radiomics features

Goal: classify Benign vs Malignant Tumors from Ultrasound images



437 scans with benign tumors
(with corresponding mask)



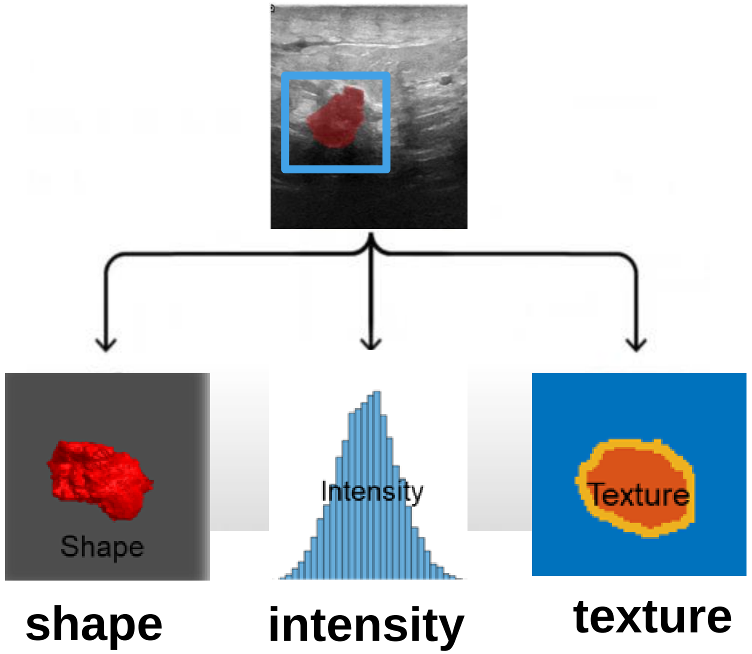
210 scans with malignant tumors
(with corresponding mask)

Split the data into 70% training data and 30% test data

Compute Radiomics features for the training set

- **Radiomics** is a technique that extracts **shape**, **intensity** and **texture** features from a specified region of interest (ROI)
- Reduces subjectivity in data analysis because it uses the same radiomics features for any medical imaging modality: MRI, CT or ultrasound

```
radiomicsFeaturesTrain = table;
for each image i-th  in the training set
    image → DATA (medicalVolume)
    mask → roiDATA (medicalVolume)

    R = radiomics(DATA, roiDATA);
    S = shapeFeatures(R, SubType="2D");
    radiomicsFeaturesTrain(i-th,:) = S;
end
```

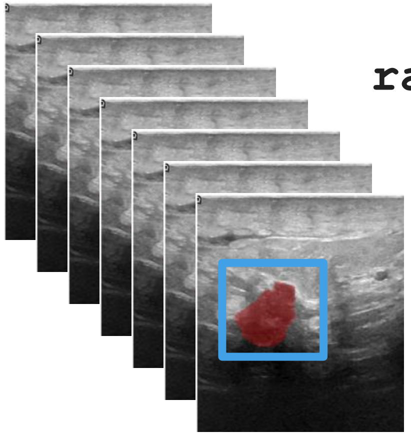


Methods of radiomics object:
shapeFeatures
intensityFeatures
textureFeatures

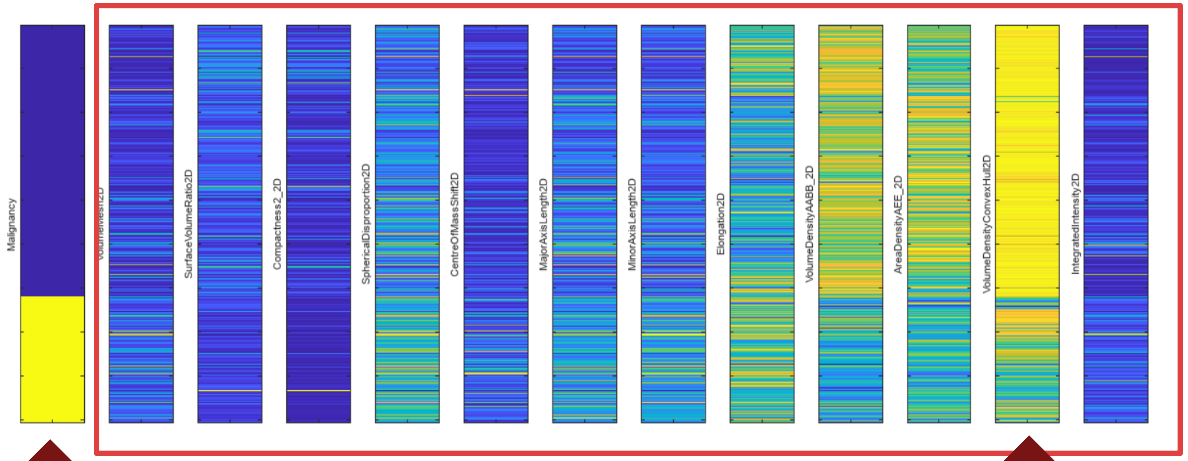
The image biomarker standardisation initiative (IBSI) provides standardized nomenclature and definitions for radiomics features and reporting guidelines

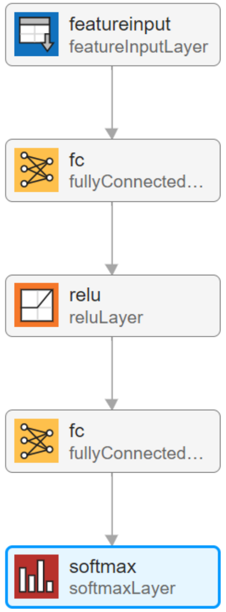
Train a classification neural network using radiomics features

- Remove redundant radiomics features (i.e. with correlation >=0.95)



`radiomics`





- Train a neural network (automatically select classification model with optimized hyperparameters)

```
model = fitcauto(radiomicsFeaturesTrain,labelsTrain, Learners="net");
```

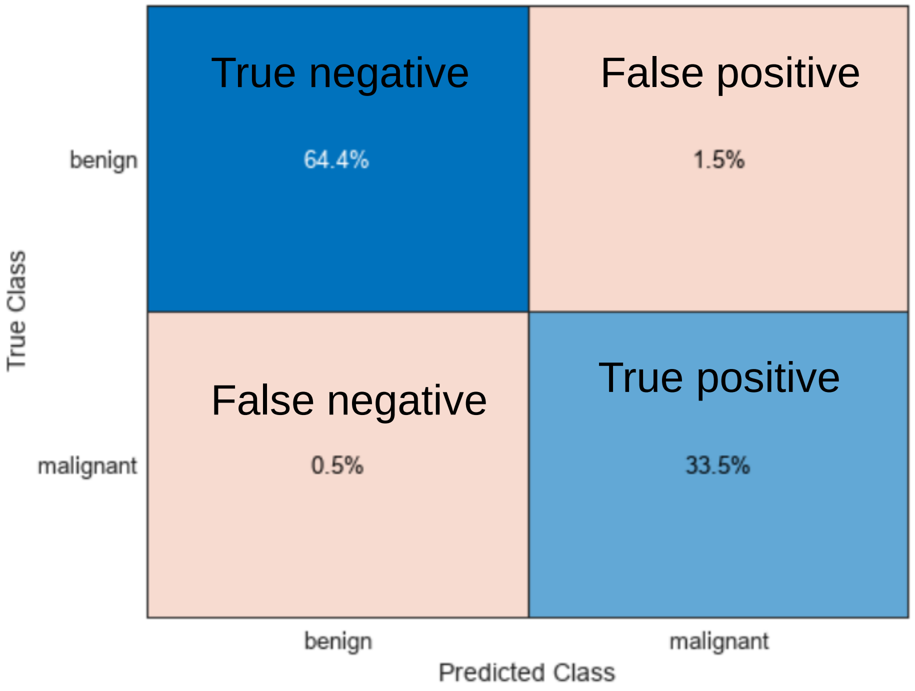
- Use the trained model to make predictions on the test dataset

```
predictedLabelsTest = predict(model,radiomicsFeaturesTest);
```

Evaluate performance and accuracy

- False negatives can be more undesirable than false positives in automated medical diagnosis.

```
falseNegativeCost = 8;
model.Cost = [0 1;falseNegativeCost 0];
predictedLabelsTest = predict(model,radiomicsFeaturesTest);
confusionchart(labelsTest,predictedLabelsTest,Normalization="total-normalized")
```



Accuracy = 97%

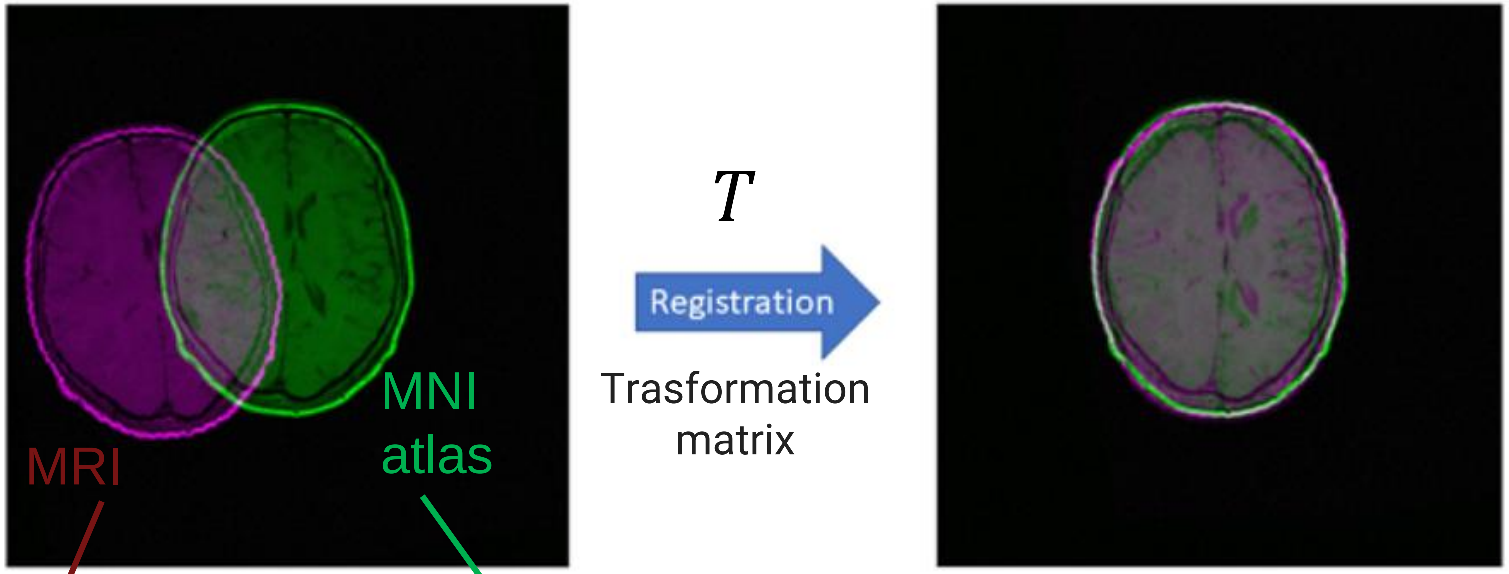
Example 2
Segmentation of Brain MRI scans
using a MONAI model

MathWorks

Step 1: Medical Registration

Preprocess

- [***Medical image registration***]() is the process of aligning multiple medical images, volumes, to a common coordinate system. You may need to compare scans of multiple patients or scans of the same patient taken in different sessions under different conditions.
- Types of registration: translation, rigid, similarity, affine, non-rigid/deformable
- In this example, before segmentation, we need to **register the brain MRI data to the MNI305 atlas**, a standardized brain atlas commonly used in neuroimaging analysis.

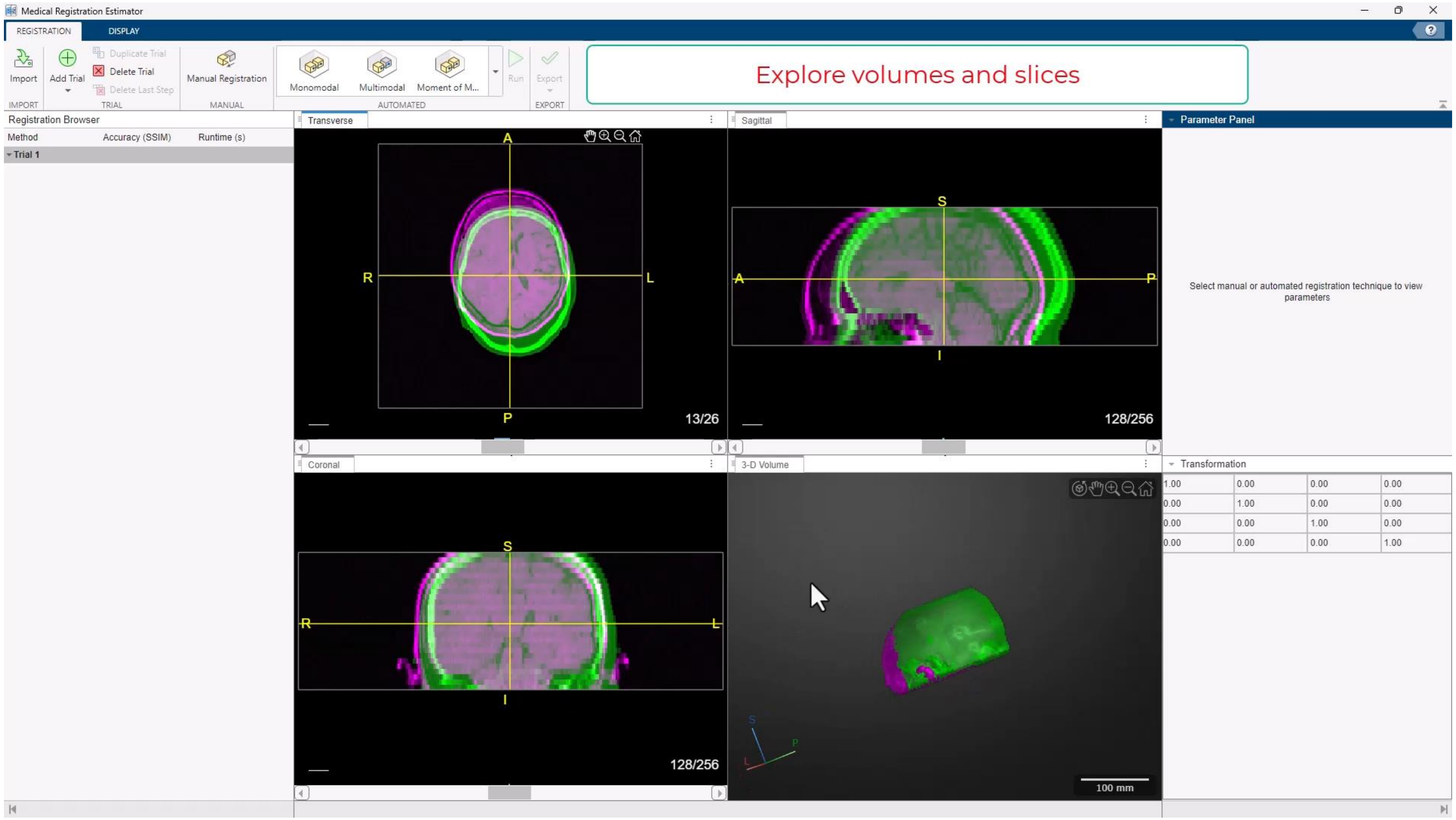


256x256x128

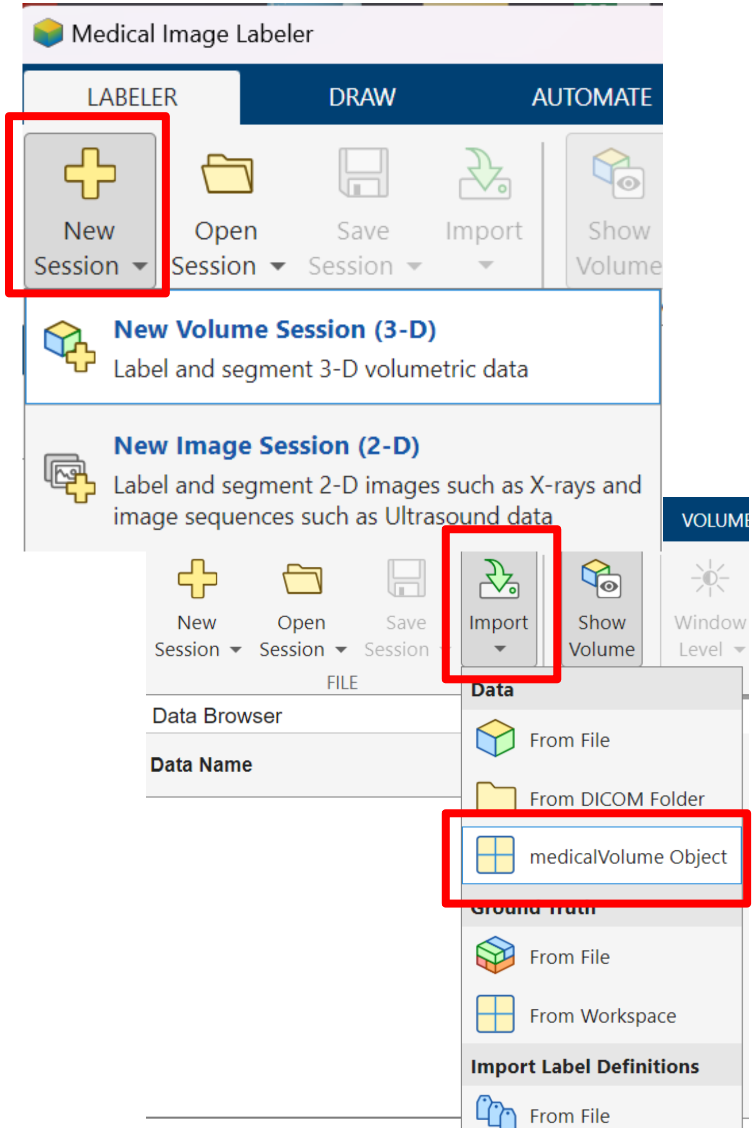
172x220x156

`resample`
`imregtform` (returns T)
`imwarp` (to apply T)
`imregicp`
`imregmoment`
`imregdemons`
`imregdeform`
…

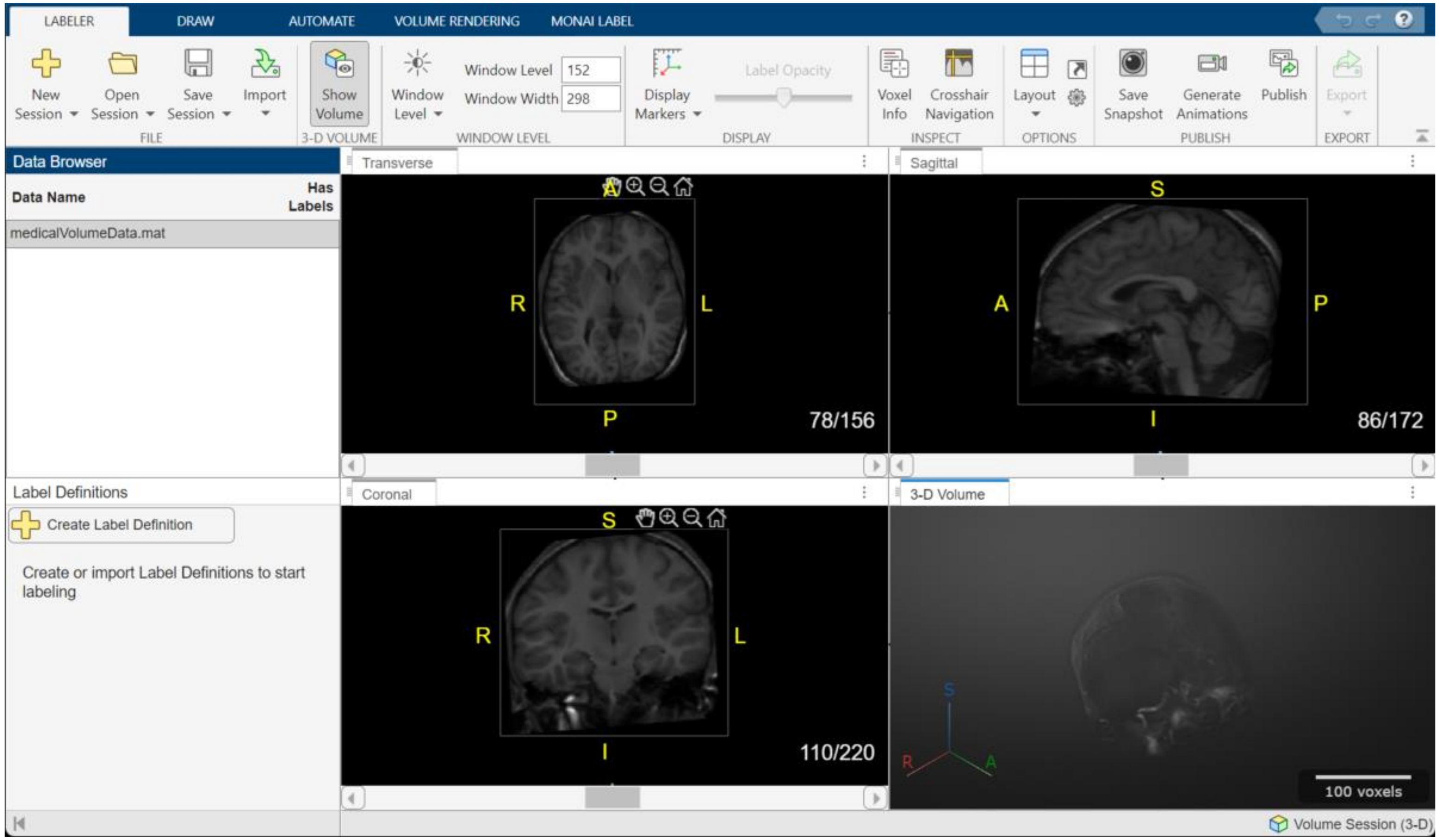
Step 1 bis: use Medical Registration Estimator App



Step 2: import registered image into Medical Image Labeler App



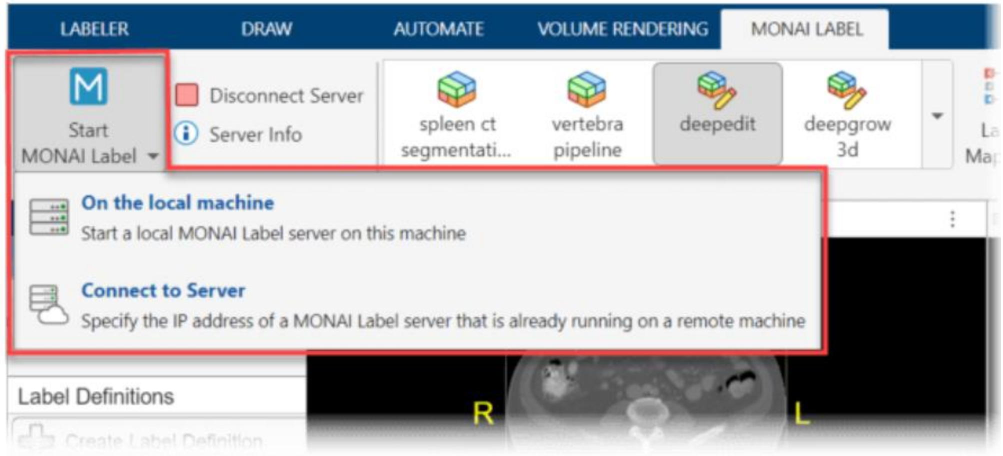

```
>> medicalImageLabeler
```

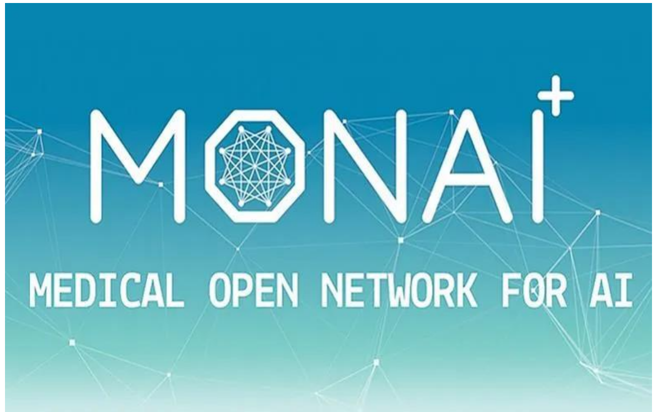


Step 3a: Use MONAI from Medical Image Labeler app

- First, install Medical Imaging Toolbox™ Interface for MONAI Label Library support package

- Use MONAI Label within the **Medical Image Labeler** app
- Start a MONAI Label server containing the deep learning models

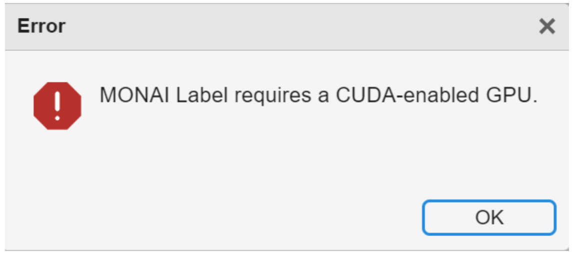




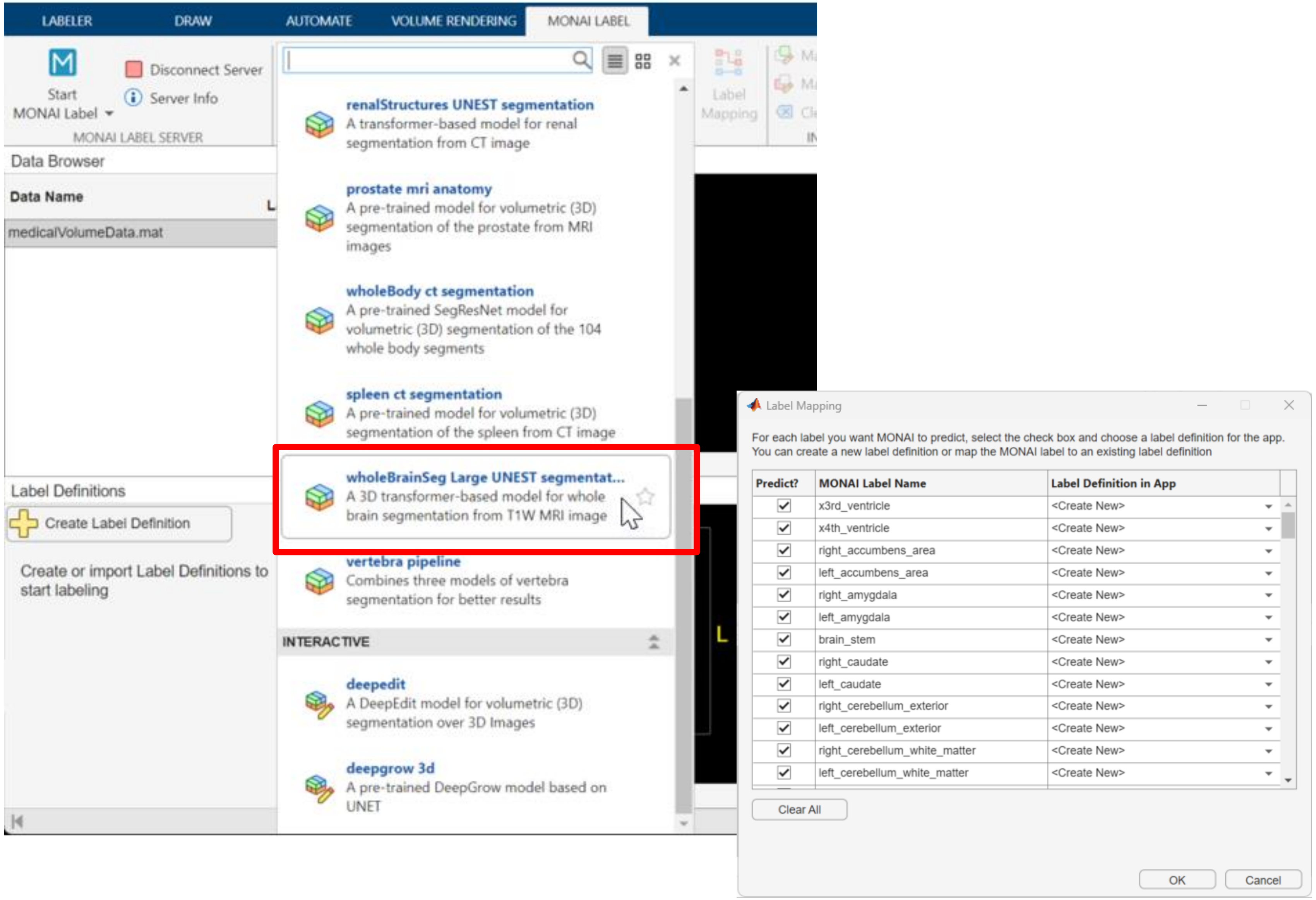
The MONAI Label platform provides fully automated and interactive deep learning models for segmenting radiology images

https://monai.io/index.html

Note: for local machine, a CUDA-enabled GPU is required, otherwise you get an error



Step 3b: choose MONAI models for segmentation



Models for one organ:

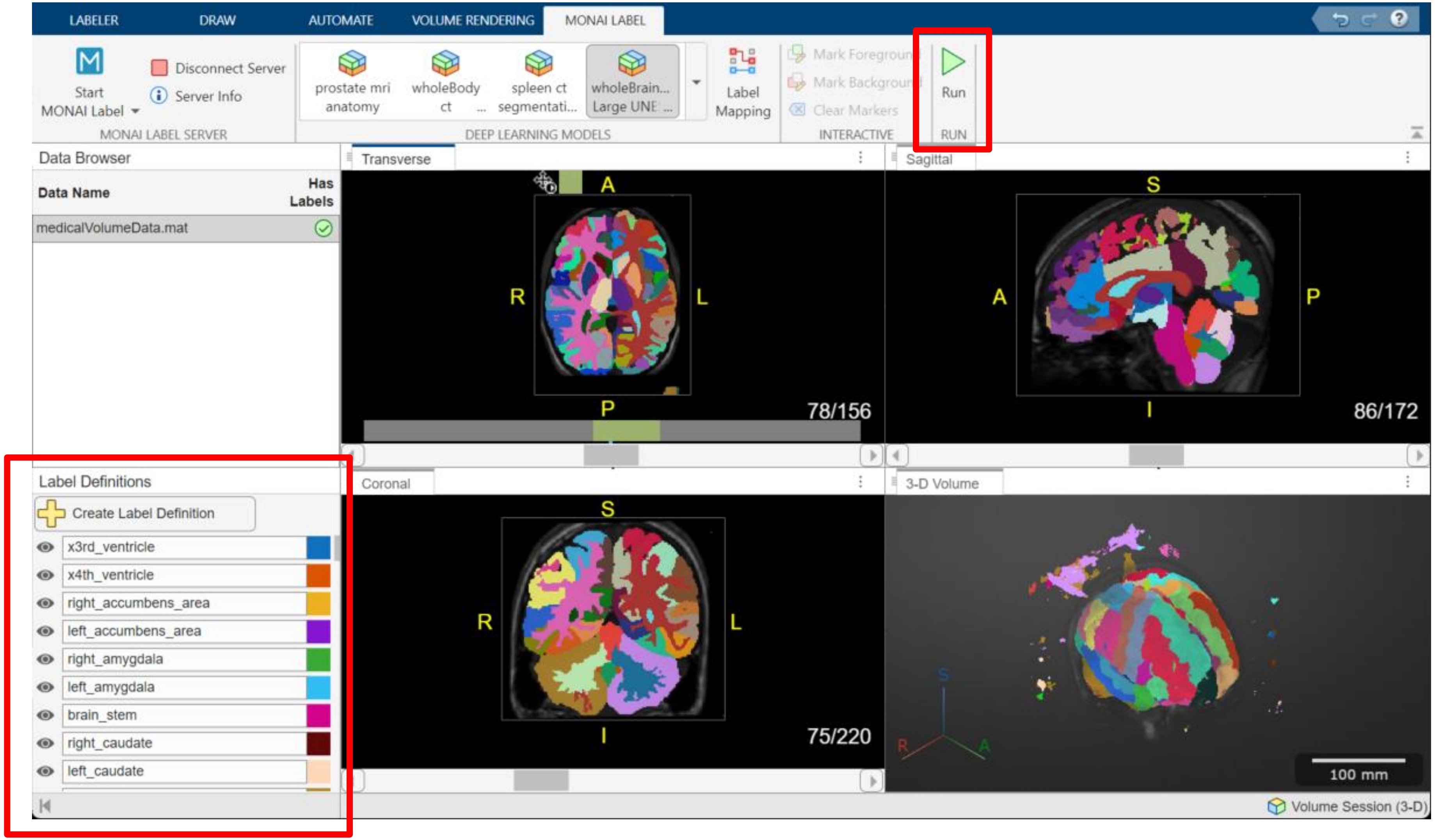
- segmentation spleen
- pancreas ct dints segmentation
- localization spine
- localization vertebra
- **wholeBrainSeg Large UNEST segmentation**

Models for multiple organs:

- wholeBody ct segmentation
- segmentation
- deepedit (interactive)
- deepgrow 3d (interactive)

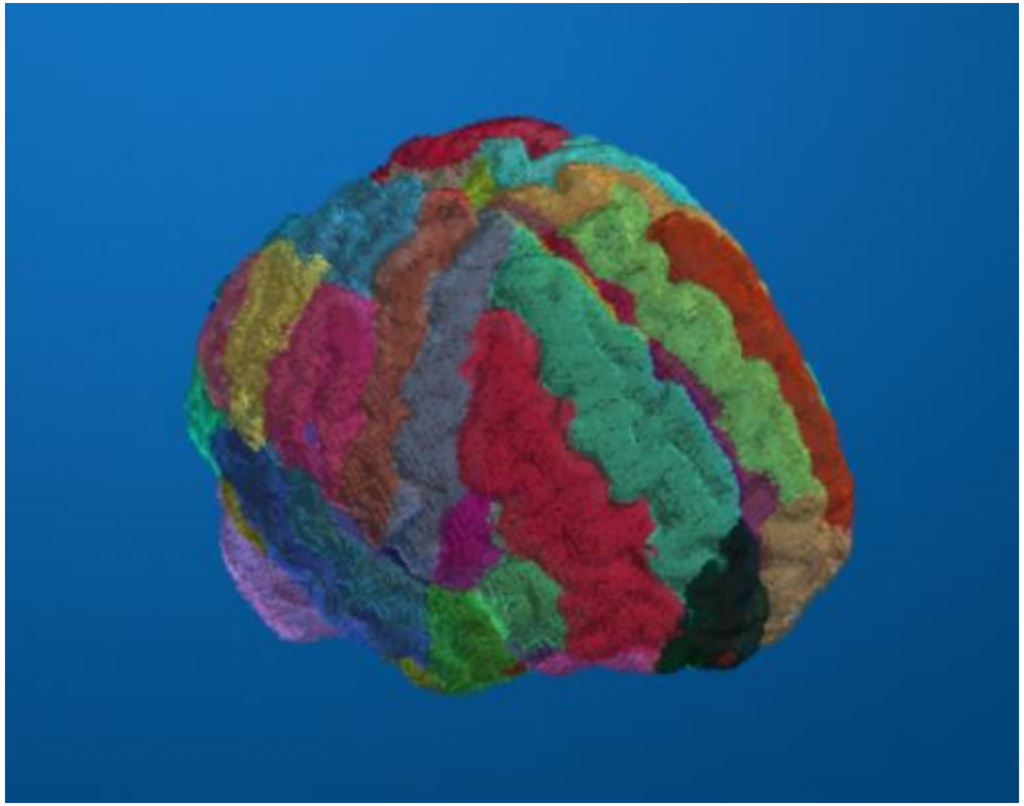
https://monai.io/model-zoo.html

Step 3c: Run the MONAI model



Step 4: analyze the segmentation results

- Overlay of the MONAI labels on the registered volume



- Calculate the volume of different brain structures and the overall brain volume

Name	Volume (voxels)	Volume (ccm)
"x3rd_ventricle"	1280	1.6875
"x4th_ventricle"	1935	2.551
"right_accumbens_area"	340	0.44824
"left_accumbens_area"	522	0.68818
"right_amygdala"	925	1.2195
"left_amygdala"	959	1.2643
"brain_stem"	14316	18.874

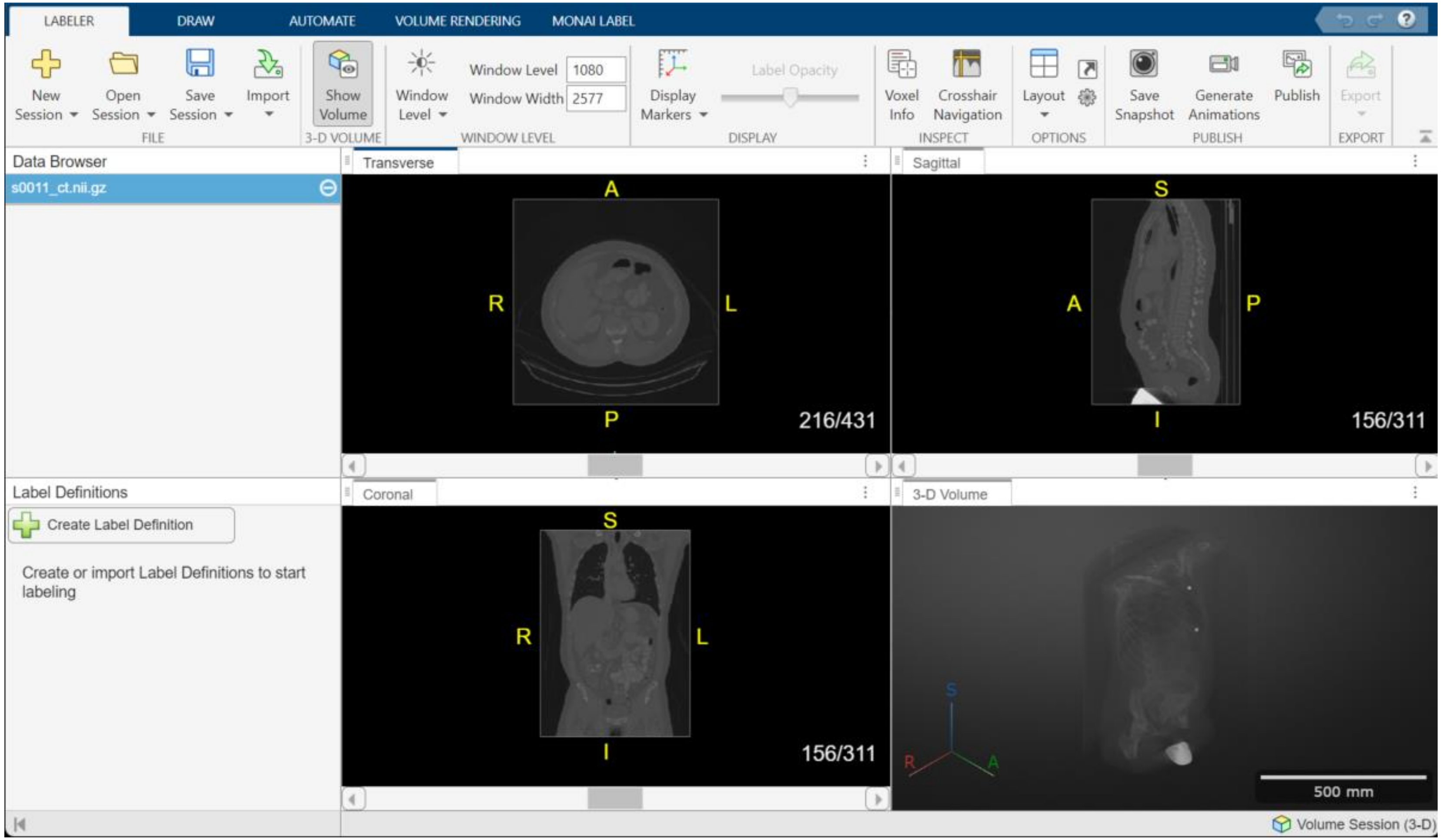
The total brain volume is about 1446 cubic cm

```
totalBrainVol = sum(volume)
totalBrainVol =
        1.4462e+03
```

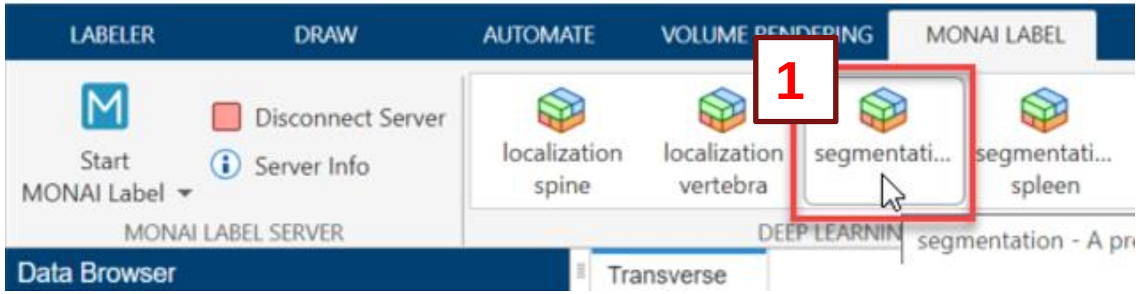
Example 3:
Segment CT scans using different MONAI models and refining labels

Step 1: import CT scans in the Medical Image Labeler app

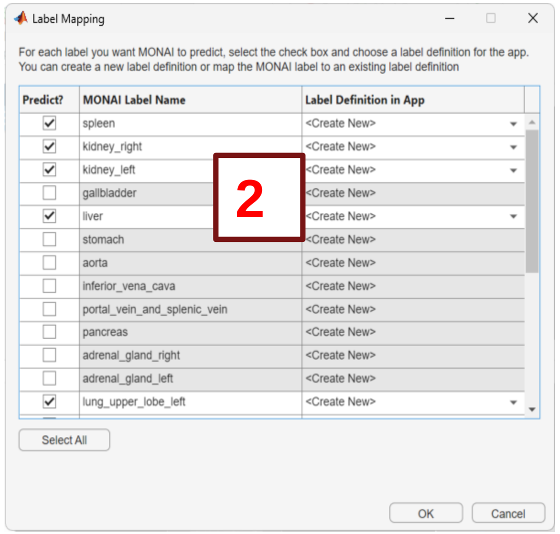
- Dataset: 1228 CT images, including abdominal and whole-body scans, with segmentation labels for up to 117 anatomical structures per scan.

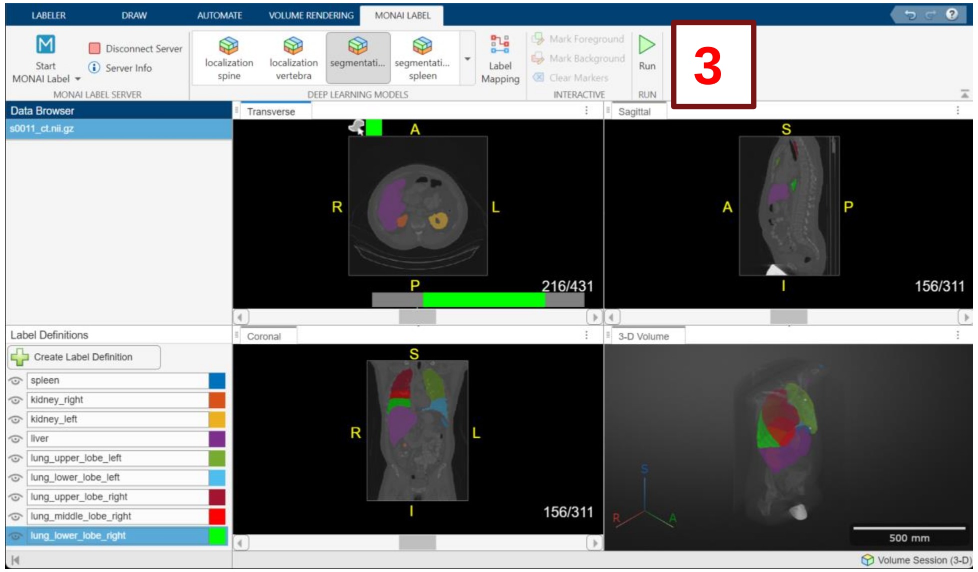


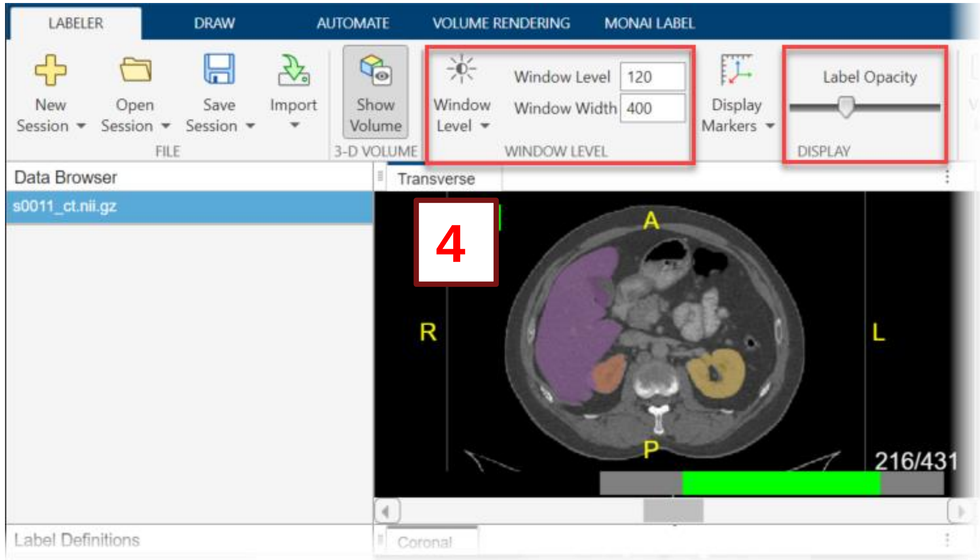
Step 2: choose multi-organ model and select the labels you want



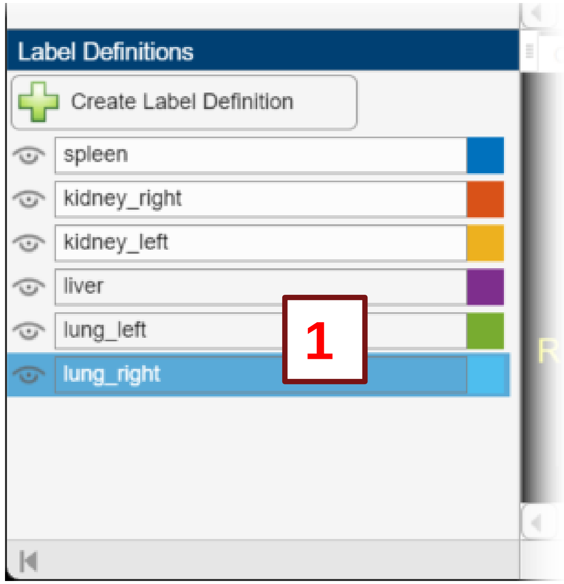
1. **Choose the model based on the labels you want to predict**
2. Select the labels you want to predict: **spleen, kidneys, all lobes of lungs,…**
3. Run the model
4. Adjust contrast and brightness changing window level/width parameters

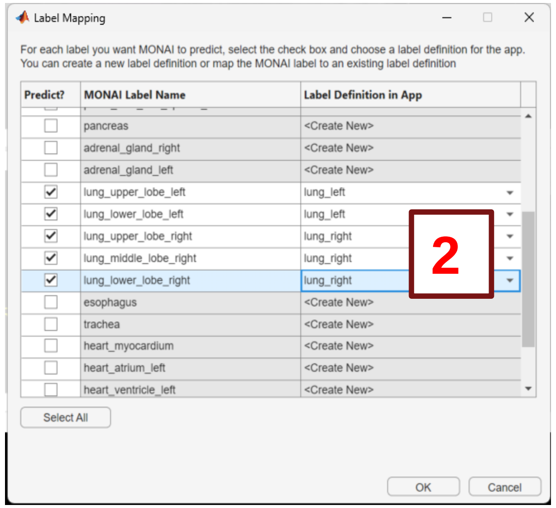




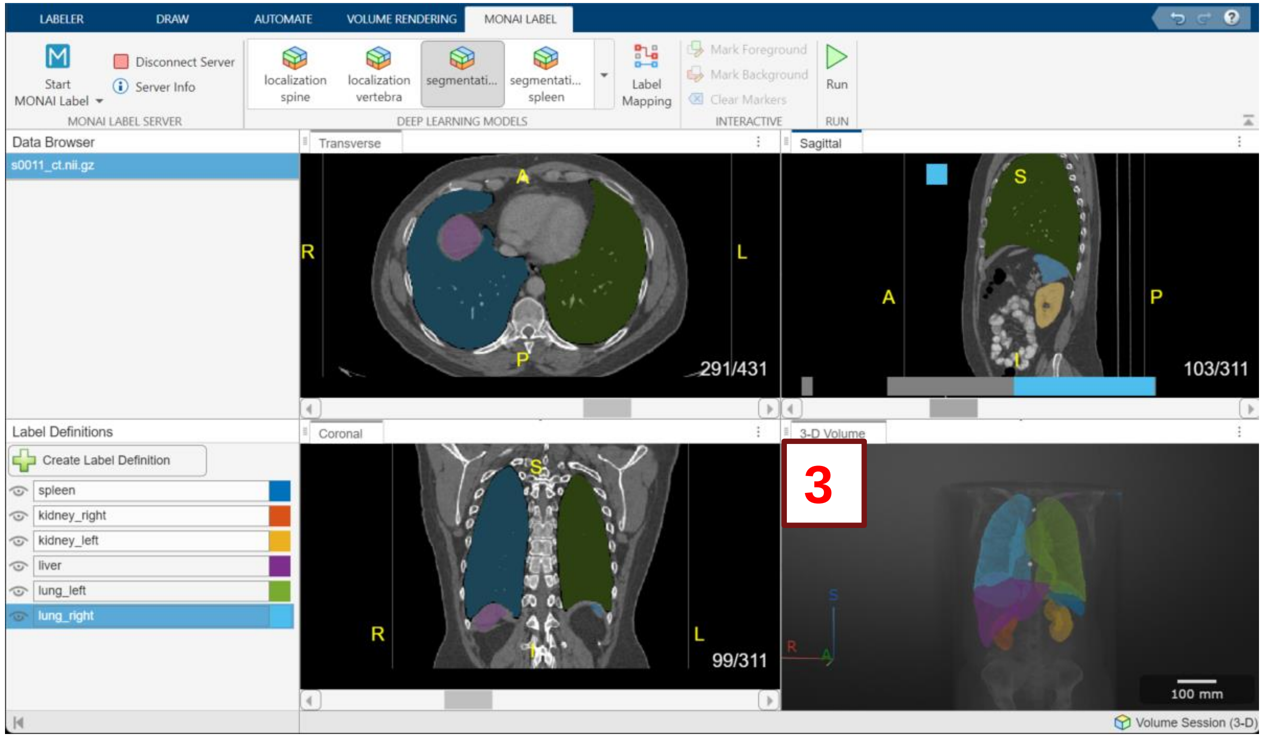


Step 3: combine labels using Label Mapping

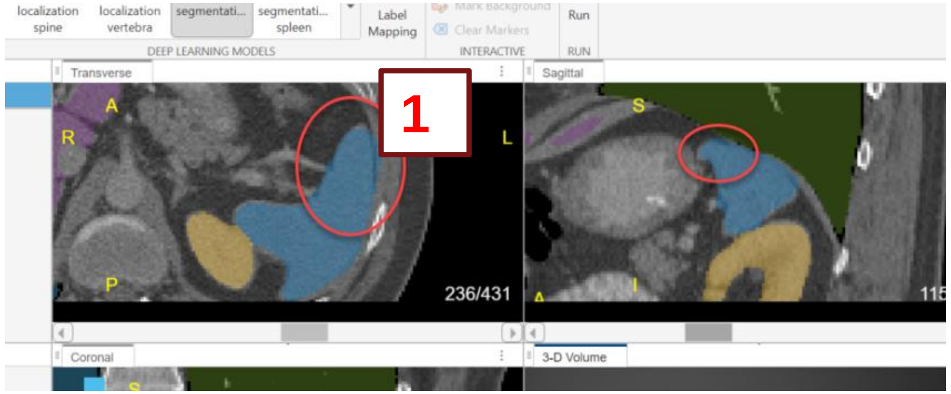




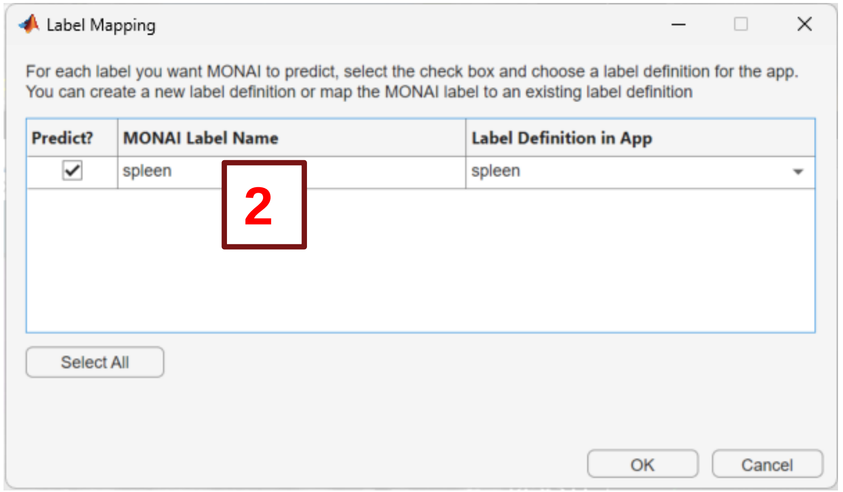
1. Delete labels for lung lobes and create 2 new labels: `lung_left` and `lung_right`
2. Remap labels
3. Rerun the same model: each lung label contains all of the corresponding lobes.

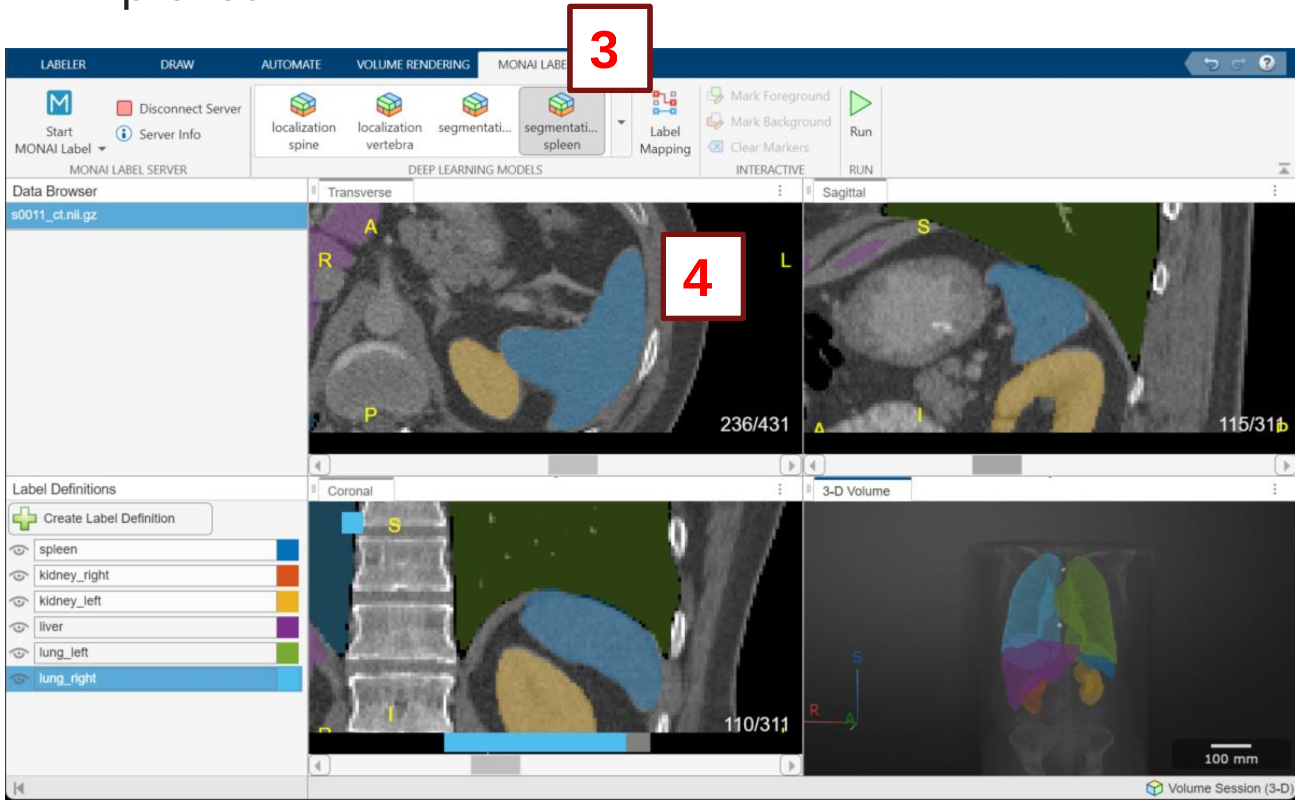


Step 4: refine labels by choosing a specific model

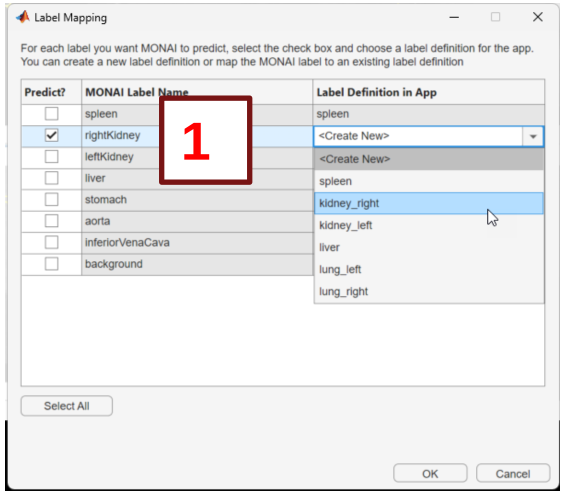


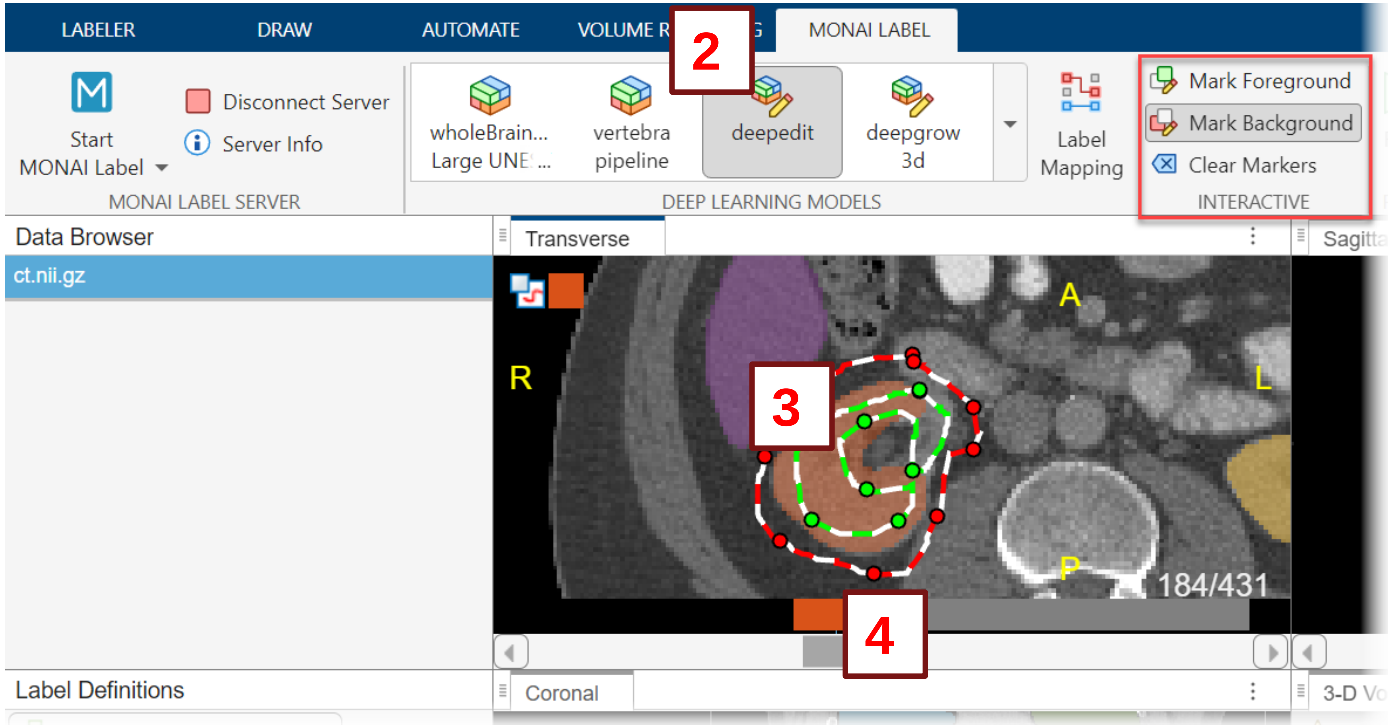
1. a few areas of the spleen label may need refinement
2. In the Label Mapping, select the spleen label
3. Select the `segmentation spleen` model
4. Run the model and visually assess whether the segmentation improved





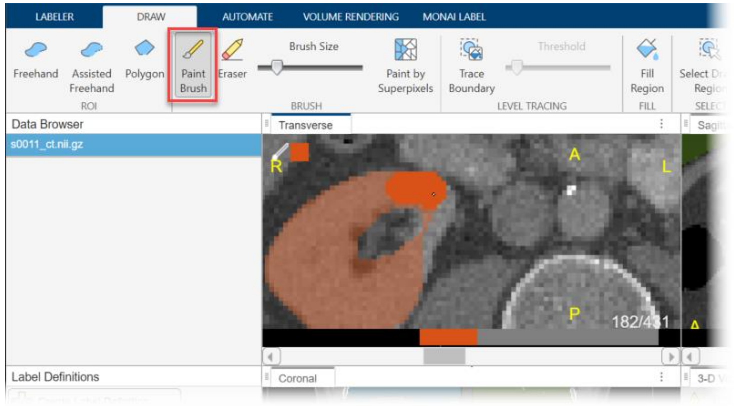
Step 5: refine labels using interactive models



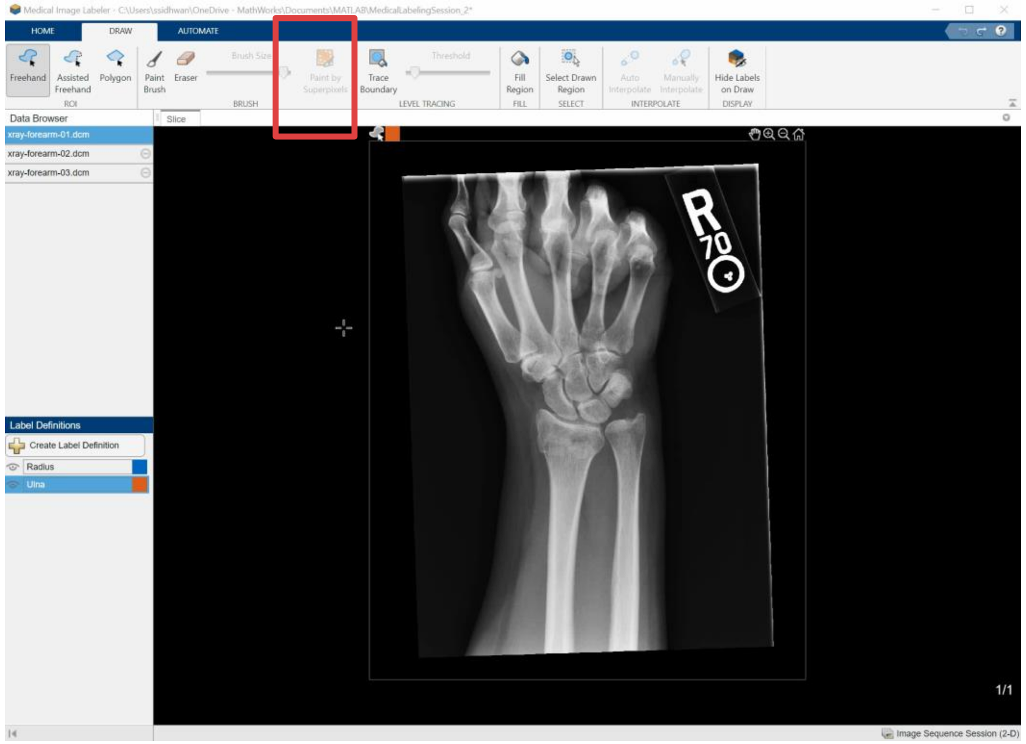


1. Select `deepedit` interactive model (requires foreground/background markers)
2. select the rightKidney label and map it to the o the existing `kidney_right` label definition.
3. To draw foreground markers (GREEN): select Mark Foreground and drag INSIDE the kidney.
4. To draw background markers (RED): select Mark Background and drag OUTSIDE the kidney
5. Run to run the model. Visually inspect the updated kidney_right label to assess whether the segmentation has improved.

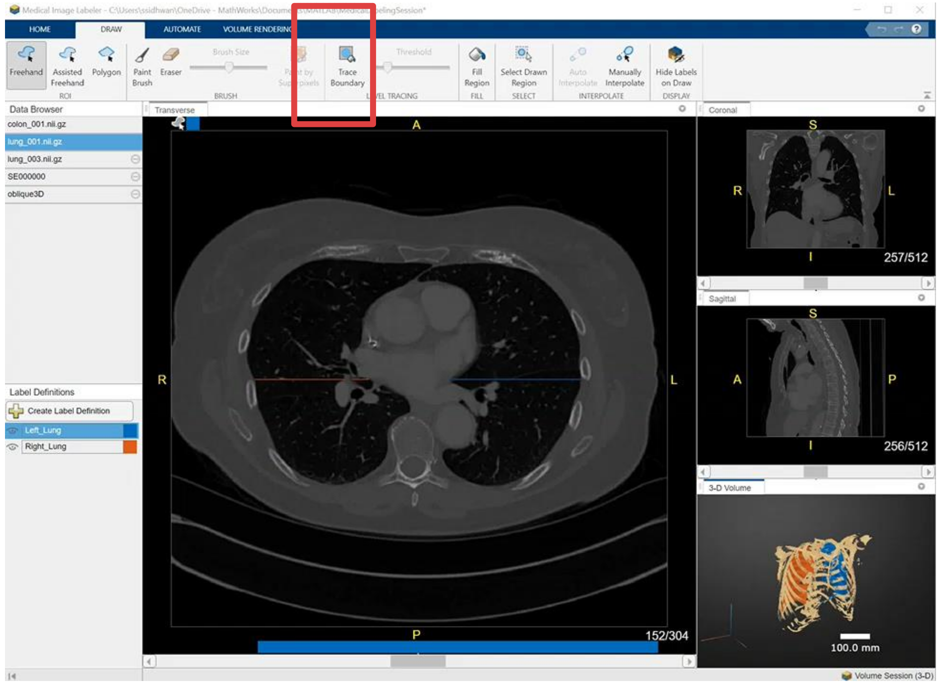
Step 6: refine labels manually (without MONAI)



Paint Brush



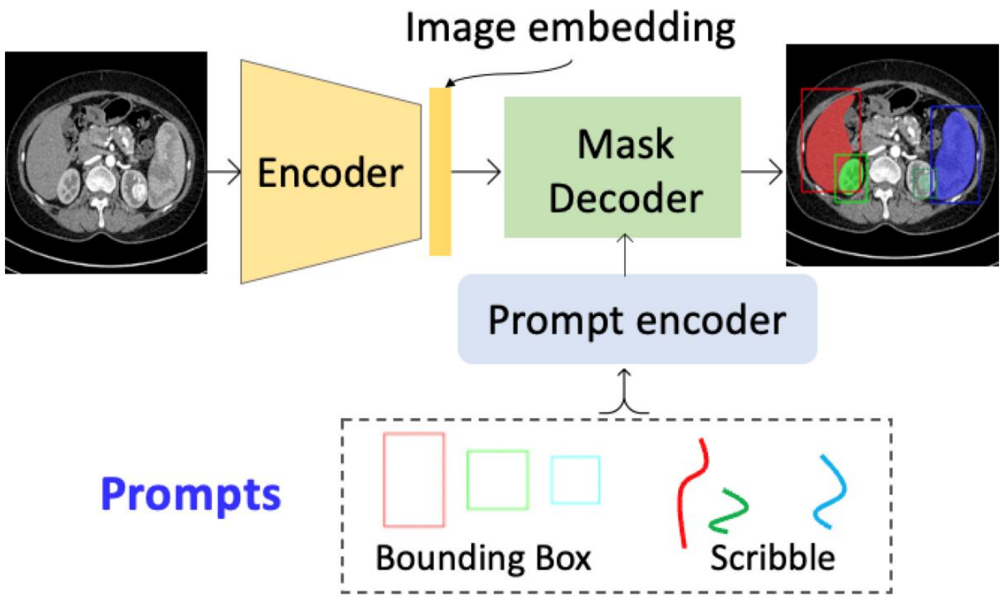
Paint by Superpixels



Level Tracing

Example 4:
Segment Heart ROI using MedSAM

MedSAM (Medical Segment Anything Model)

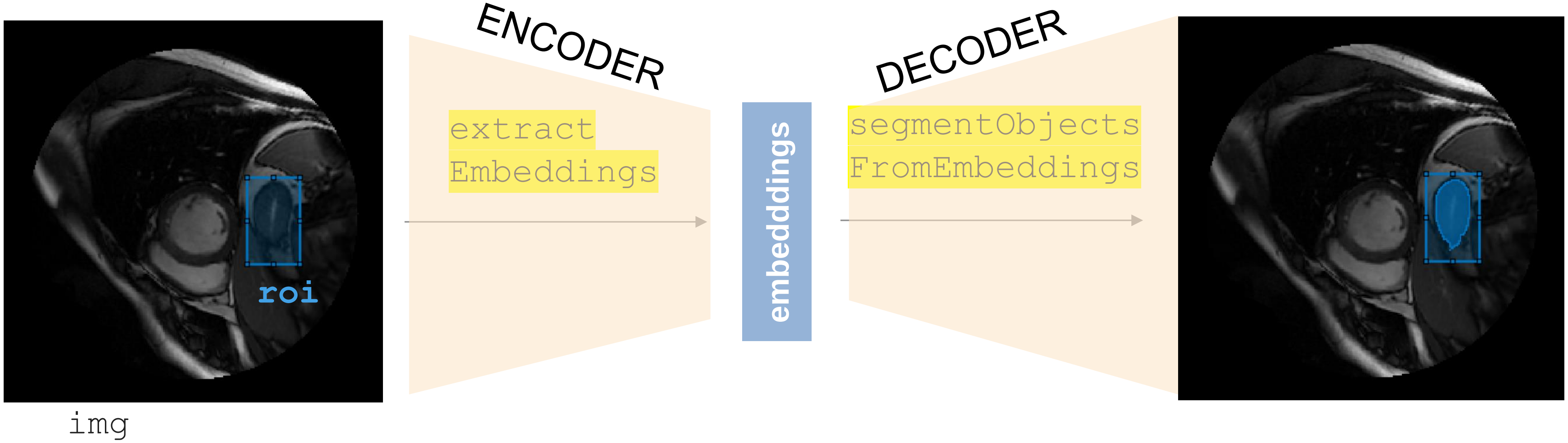


In MATLAB install this support package

MedSAM is a state of the art, **zero-shot**, **foundational, medical image segmentation model.** MedSAM is adapted from SAM to perform segmentation **specifically for medical images** from different modalities including CT, MRI, Endoscopy, X-ray, Ultrasound, Pathology etc. The zero-shot capabilities of MedSAM enable to use this model for image segmentation without the need for re-training or transfer learning

https://www.nature.com/articles/s41467-024-44824-z

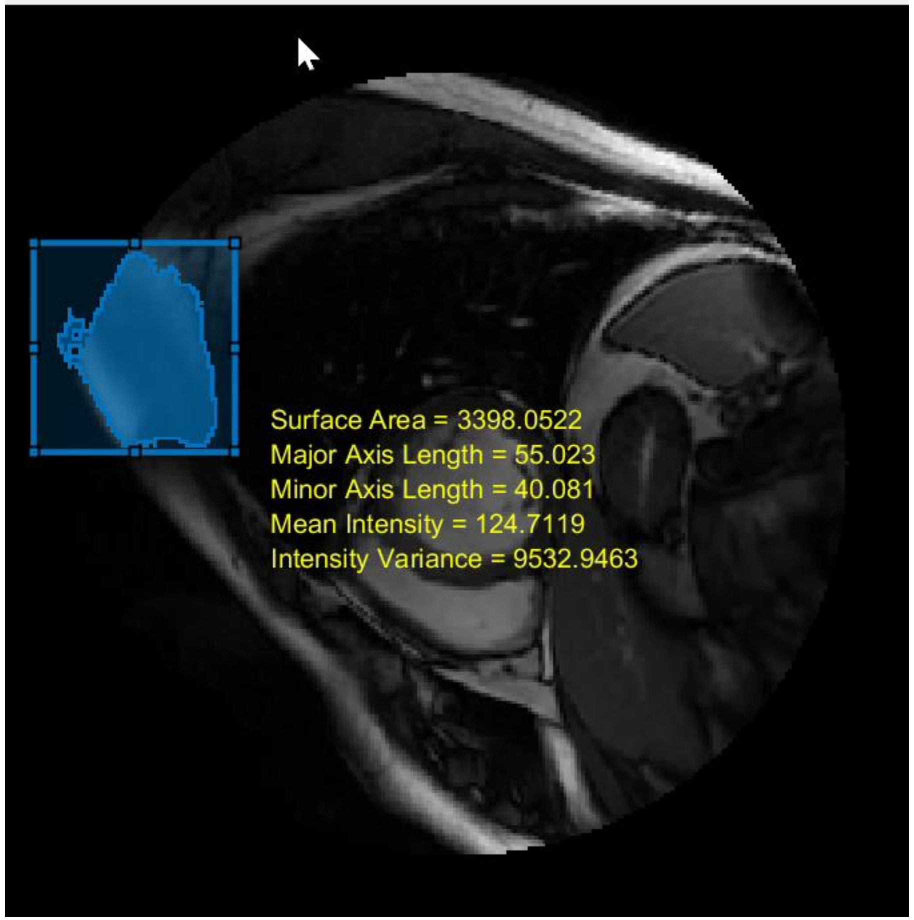
Segment image using medSAM in MATLAB




```
medsam = medicalSegmentAnythingModel;
embeddings = extractEmbeddings(medsam, img);                % ENCODER
roi = drawrectangle(ax);
mask = segmentObjectsFromEmbeddings(medsam, embeddings,    % DECODER
                size(img),BoundingBox = roi.Position);
```

Note: image taken from 45 cine-MRI images, DICOm format, acquired from multiple patients with various cardiac pathologies (healthy, hypertrophy, heart failure with infarction and heart failure without infarction).

Interactive setup to recompute segmentation for moving roi



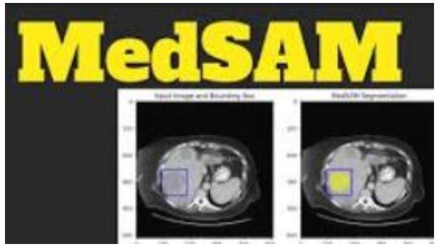

```
addlistener(roi,"ROIMoved",
  @(src,evt) ...
segmentAndAnalyzeROI(evt.CurrentPosition,med
sam,embeddings,img,dispIm,t));


function segmentAndAnalyzeROI (callback)
    apply decoder segmentObjectsFromEmbeddings
    Fill the region with insertObjectMask
    Compute radiomics features
    Visualize radiomics features in yellow
end
```

Conclusions

We have learnt …

MATLAB makes Medical Imaging easier

- Interactive apps for Registration, Image Labeling
- Integration of medical models for Image segmentation



Examples:

1. Classification of Tumors using Radiomics features
2. Segmentation of Brain MRI using a MONAI model
3. Segmentation of CT scans using different models of MONAI
4. Segmentation of heart ROI using MedSAM

References

- Medical Imaging Toolbox
- Medical Imaging Toolbox™ Interface for MONAI Label Library support package
- Medical Imaging Toolbox Model for Medical Segment Anything Model
- Example: Classify Breast Tumors from Ultrasound Images using Radiomics features
- Example: Segment and Analyse Brain MRI scan using AI
- Example: Segment CT scan using MONAI Label
- Example: Interactively Segment and analyze ROI using MedSAM and Radiomics
- Get Started with Radiomics.
- Get Started with MONAI Label in Medical Image Labeler.
- Get Started with Medical Segment Anything Model for Medical Image Segmentation.
- MATLAB Portal for University of Parma

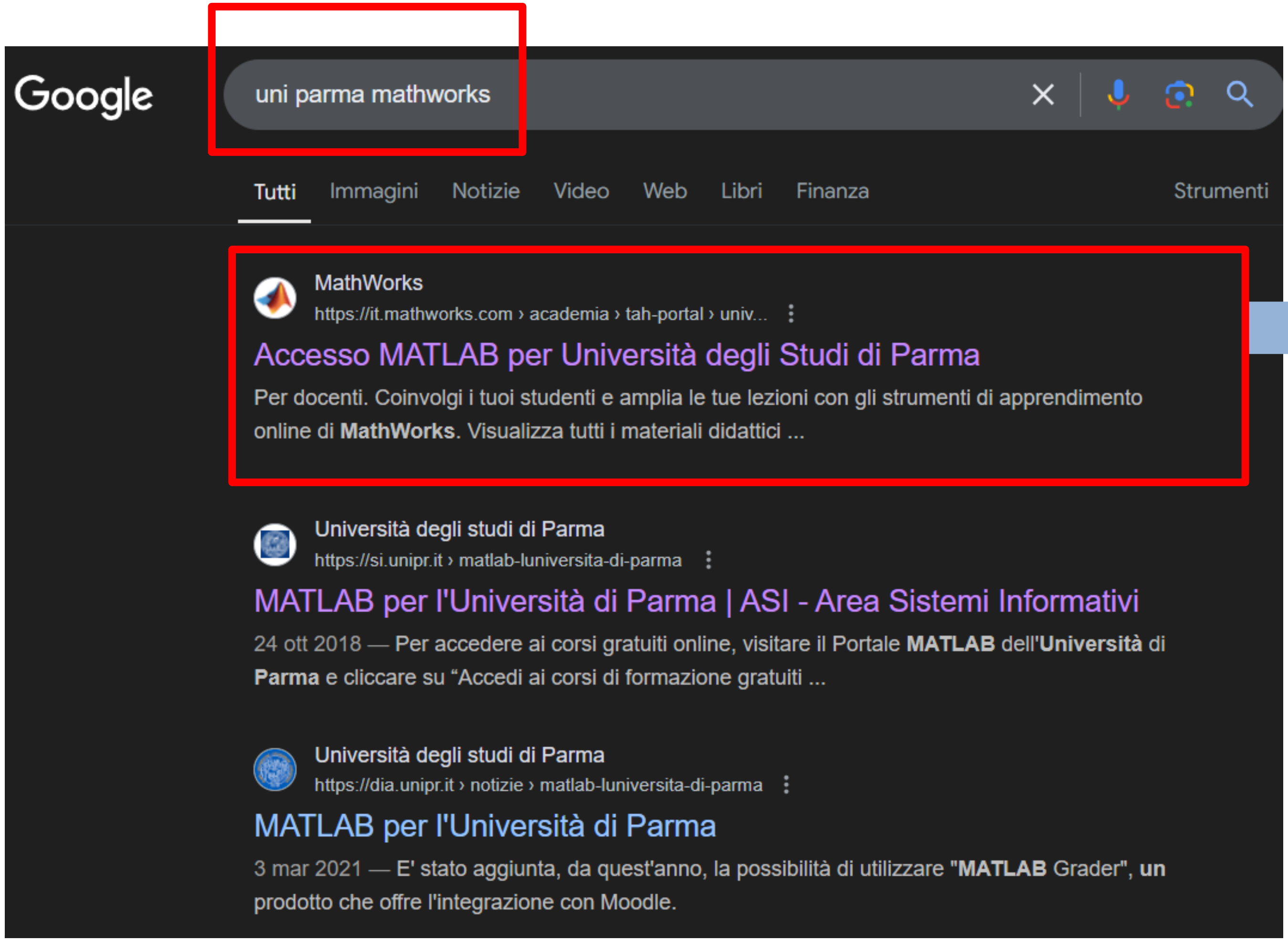
For any technical question: please

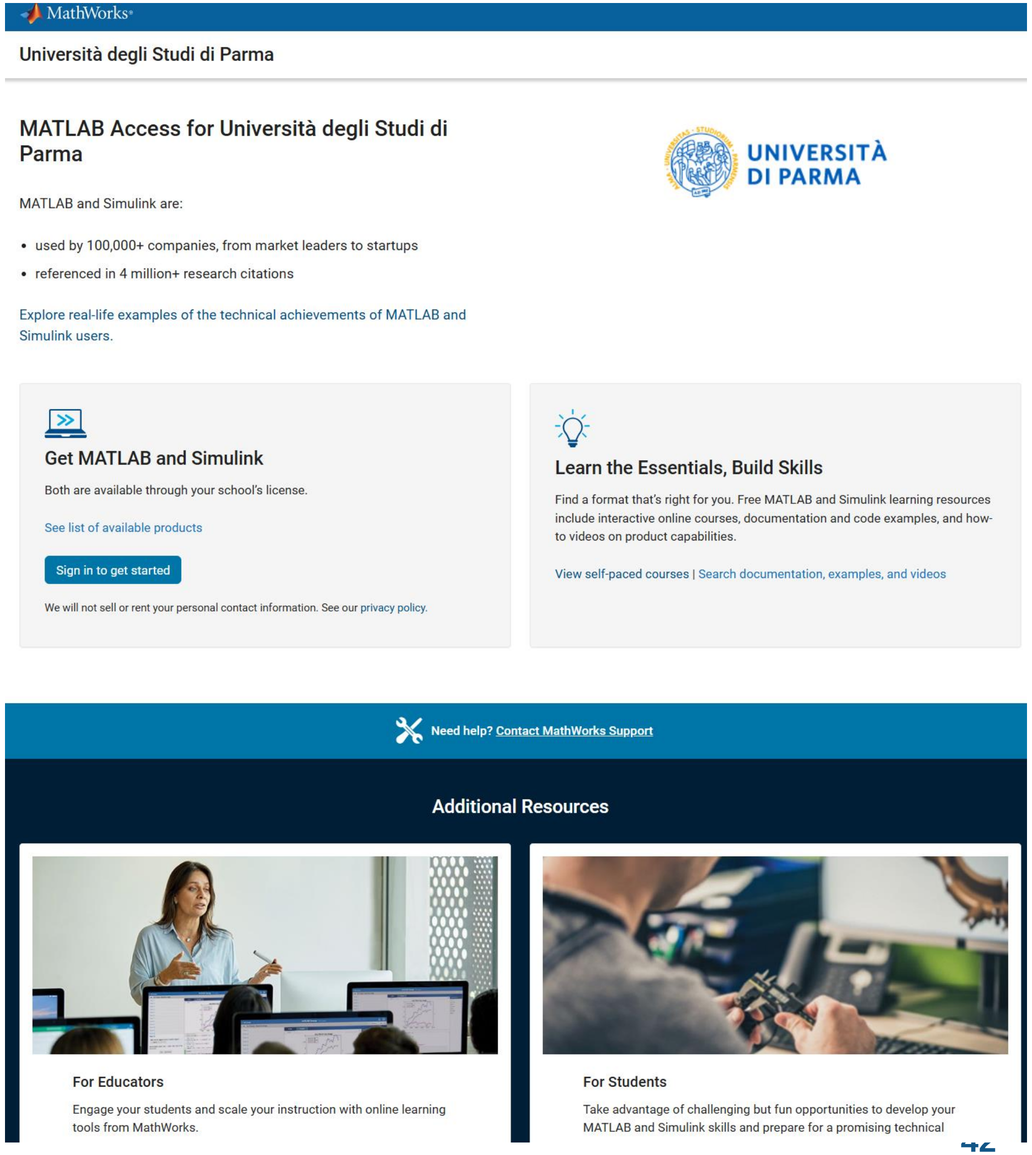
- contact MathWorks Tech Support
- write to Paolo Panarese: ppanares@mathworks.com

Free access to your MATLAB Campus Wide License

Anytime, **Anywhere Access** for Faculty, Staff, Students, and Visitors

https://www.mathworks.com/products.html





MathWorks

Free courses (onramp) with certification

https://matlabacademy.mathworks.com/

Self-Paced Courses

- Getting Started (22)
- MATLAB (6)
- Simulink (10)
- AI, Machine Learning, and Deep Learning (6)
- Physical Modeling (5)
- Math and Optimization (7)
- Image and Signal Processing (6)
- System Engineering (1)
- Online Training Suite (12)

Explore over 50 virtual and in-person **classroom courses**

AI, Machine Learning, and Deep Learning

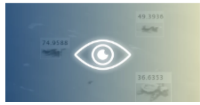
Machine Learning Onramp
6 modules | 2 hours | Languages
Learn the basics of practical machine learning methods for classification problems.

Machine Learning with MATLAB
6 modules | 12 hours | Languages
Explore data and build predictive models.

Deep Learning Onramp
5 modules | 2 hours | Languages
Get started quickly using deep learning methods to perform image recognition.

Deep Learning with MATLAB
11 modules | 7 hours | Languages
Learn the theory and practice of building deep neural networks with real-life image and sequence data.

Reinforcement Learning Onramp
100%
5 modules | 2.5 hours | Languages
Master the basics of creating intelligent controllers that learn from experience.



Computer Vision Onramp
6 modules | 2 hours | Languages
Learn the basics of computer vision to design an object detector and tracker.

Self-Paced Courses

- Getting Started (22)
- MATLAB (6)
- Simulink (10)
- AI, Machine Learning, and Deep Learning (6)
- Physical Modeling (5)
- Math and Optimization (7)
- Image and Signal Processing (6)
- System Engineering (1)
- Online Training Suite (12)

Explore over 50 virtual and in-person **classroom courses**

Image and Signal Processing

Image Processing Onramp
6 modules | 2 hours | Languages
Learn the basics of practical image processing techniques in MATLAB.

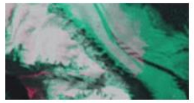
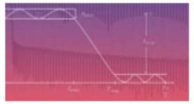


Image Processing with MATLAB
11 modules | 11 hours | Languages
Learn practical image processing workflows in MATLAB.

Computer Vision Onramp
6 modules | 2 hours | Languages
Learn the basics of computer vision to design an object detector and tracker.

Signal Processing Onramp
7 modules | 1.5 hours | Languages
An interactive introduction to signal processing methods for spectral analysis.



Signal Processing with MATLAB
8 modules | 7.5 hours | Languages
Learn how to perform signal processing in MATLAB.

Wireless Communications Onramp
6 modules | 1 hour | Languages
Learn the basics of simulating a wireless communications link in MATLAB.

Given Imaging (Medtronic) Develops Camera-in-a-Capsule Using MATLAB to Improve the Diagnosis of Gastrointestinal Disorders

Challenge

Create an alternative to endoscopy and other invasive gastrointestinal imaging procedures

Solution

Use MATLAB and companion toolboxes to develop and implement a swallowable video capsule

Results

- Fast, efficient development
- Easy access to precise diagnostic information
- Improved patient care

The PillCam® SB 2 video capsule.

“With MATLAB, we simulated the intended system and fine-tuned it at the early stages of implementation, enabling us to develop critical engineering programs that met requirements on the first iteration.”

- Rafi Nave, Given Imaging

Link to user story

Philips Healthcare Develops Smart Digital RF Power Subsystem for MRI Systems

Challenge

Develop a novel digital RF power subsystem for use in MRI systems

Solution

Use Simulink to model, simulate, and verify the design, and use HDL Coder to generate consistent and predictable VHDL code for the FPGA implementation

Results

- Design issues resolved early in development
- Tradeoffs rapidly assessed and implemented
- Process consistency and predictability improved

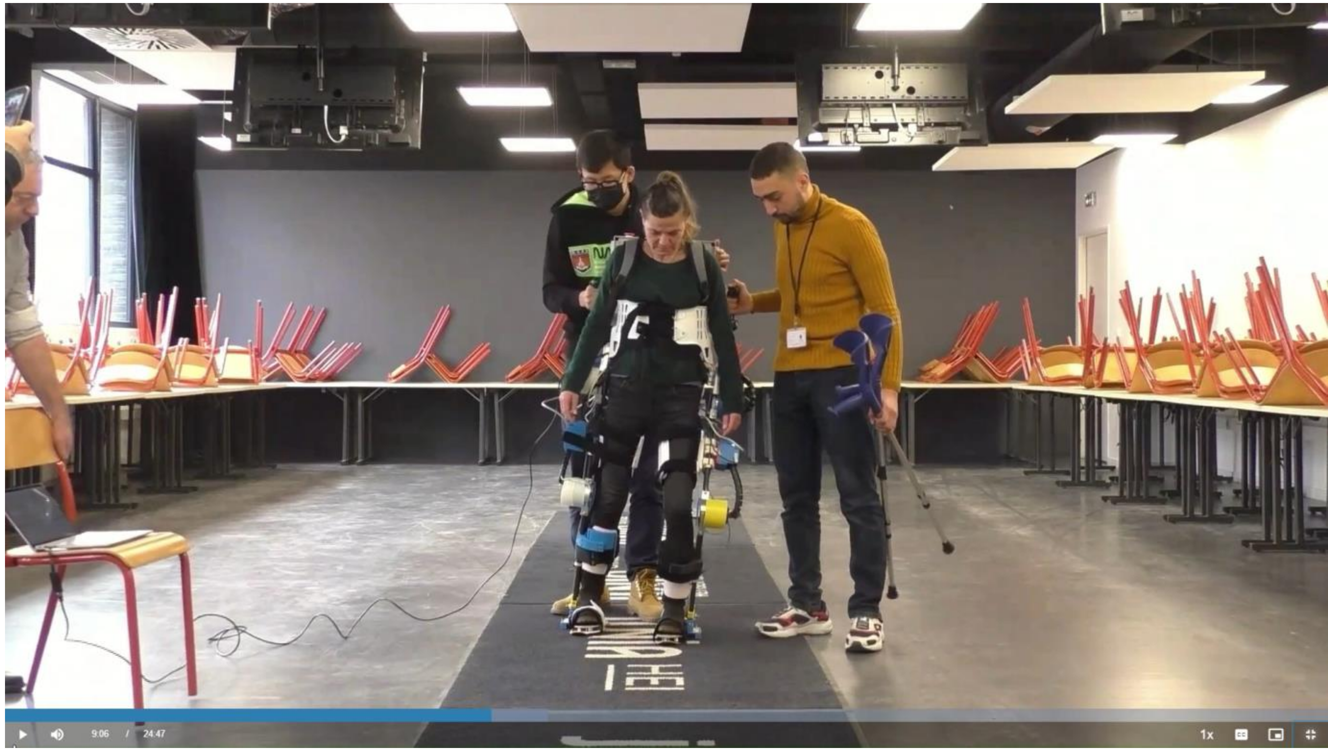
Van Helvoort (left) and van Bakel with a Philips Healthcare MRI scanner.

"Simulink helps system architects and hardware designers communicate. It is like a shared language that enables us to exchange knowledge, ideas, and designs. Simulink and HDL Coder enable us to focus on developing our algorithms and refining our design via simulation, not on checking VHDL syntax and coding rules."

- Juha Inberg, Ponsse

[Link to user story](#)

JUNIA Develops Autonomous Pediatric Exoskeleton for Children with Severe Neurological Disorders

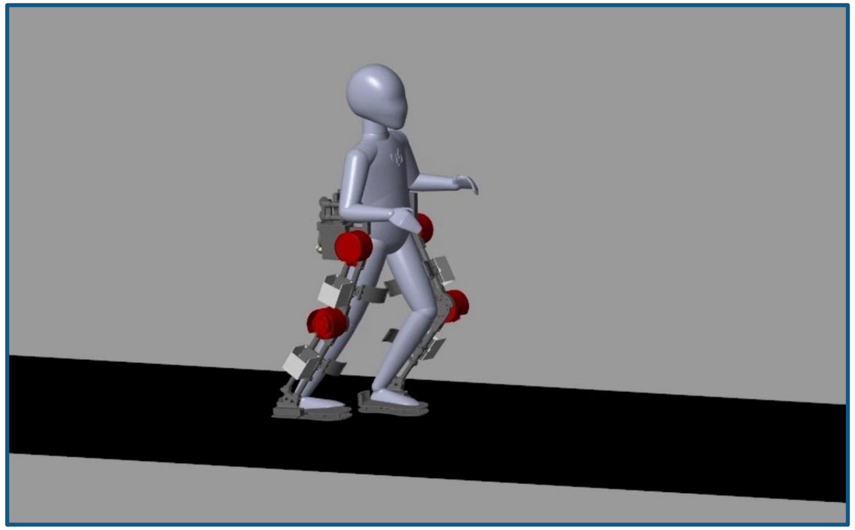


Challenge

Develop an exoskeleton to be used as a physical therapy tool for children with cerebral palsy.

Solution

Use MATLAB, Simulink, and Simscape Multibody to model motor dynamics and design motor controllers. Conduct real-time testing of prototype hardware with Speedgoat.



Model of the JUNIA exoskeleton. (Image credit: JUNIA HEI)

"We found that Simulink Real-Time is the best choice because you can design whatever model you want, whatever control algorithm you want, and you can apply it quickly to your prototype." — Yang Zhang, postdoctoral researcher at JUNIA HEI

Link to feature story

University of Twente Develops Software for Visualizing Reduced Blood Circulation with Augmented Reality and Deep Learning

Challenge

Enable clinicians and diabetic patients to visualize reduced blood circulation using an affordable, handheld imaging device

Solution

Use MATLAB to develop algorithms that construct a 3D representation of blood circulation and project that representation on the skin's surface

Results

- Data acquisition, localization, mapping, and AR projection algorithms run in real time
- Deep learning models trained to detect poor blood flow before it becomes visible
- Low-cost, fully functional prototype developed and tested

Augmented reality (AR) visualization of blood flow in the wrist and hand.

"One of the biggest advantages of using MATLAB in my research is the ability to use a single platform for all aspects of the project, including image processing and computer vision, SLAM, and deep learning."

- Dr. Beril Sirmacek, University of Twente

Link to article

Thank you!

- Q&A

Accelerating the pace of engineering and science

For any technical question: please

- contact MathWorks Tech Support
- write to Paolo Panarese: ppanares@mathworks.com