framework for deep learning in healthcare imaging

Michał Sokólski

Agenda

1. What is MONAI
2. Overview of MONAI modules
3. Usage Example
4. My subjective overview of the package
5. Q&A

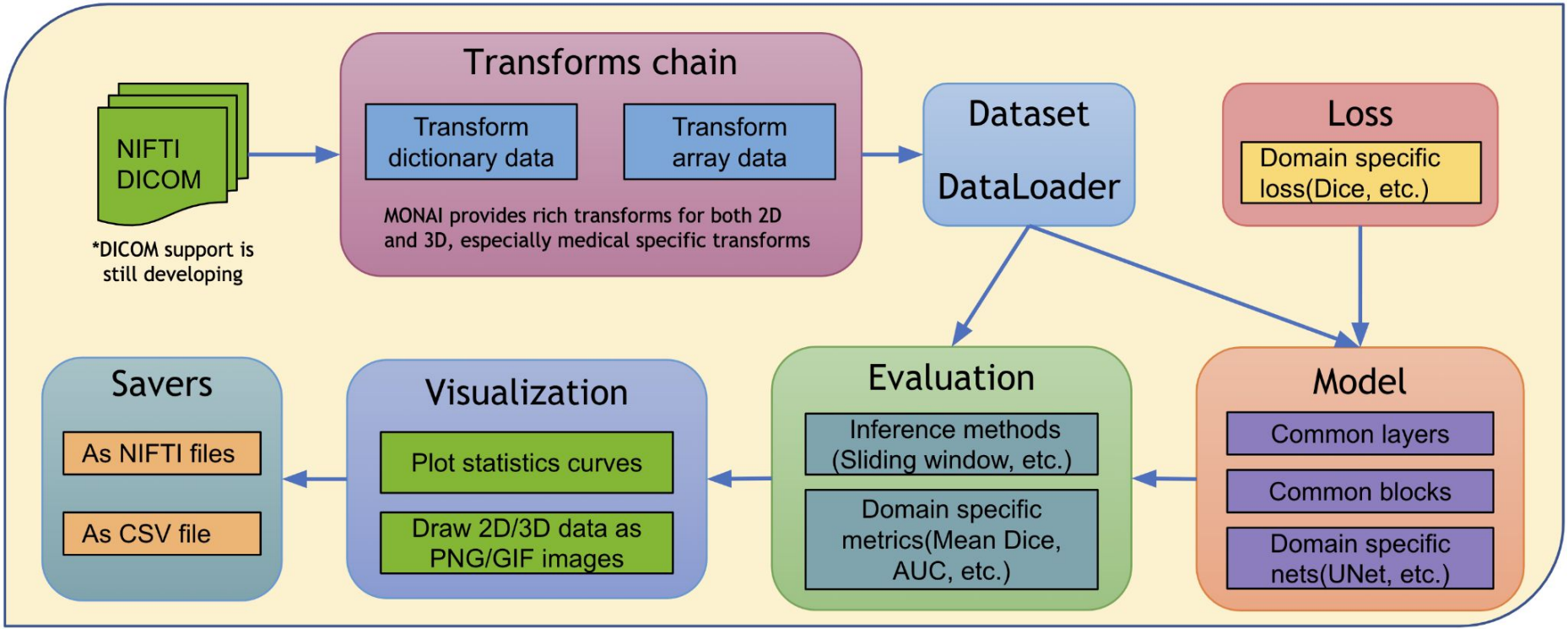
MONAI

MONAI (https://monai.io/) 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.

End-to-end training workflow



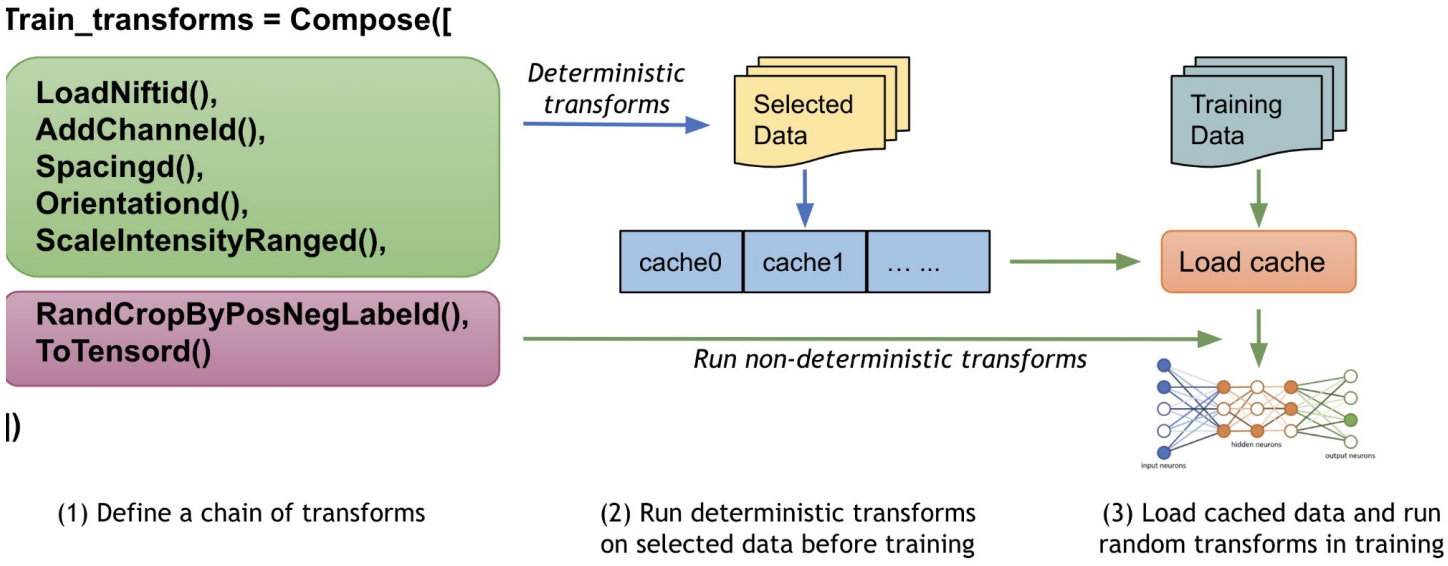
Modules: I/O processing

- Pre-defined data readers for: NIfTI, DICOM, PNG, JPG, BMP, NPY/NPZ
- ITKReader for other formats (a lot of them...)
- 2D and 3D data handling
- Many, many data transforms like:
  - Crop and pad
  - Intensity
  - IO
  - Post-processing
  - Spatial

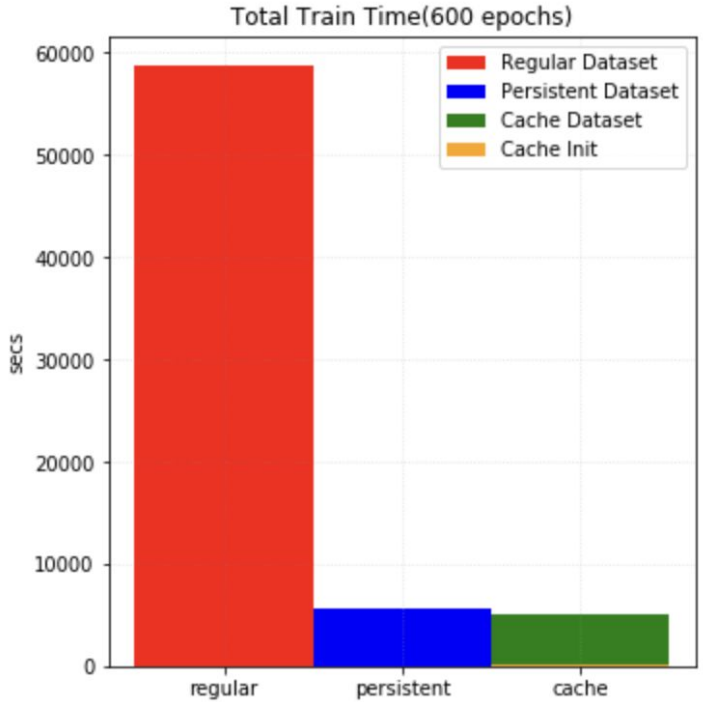
https://docs.monai.io/en/latest/transforms.html#vanilla-transforms

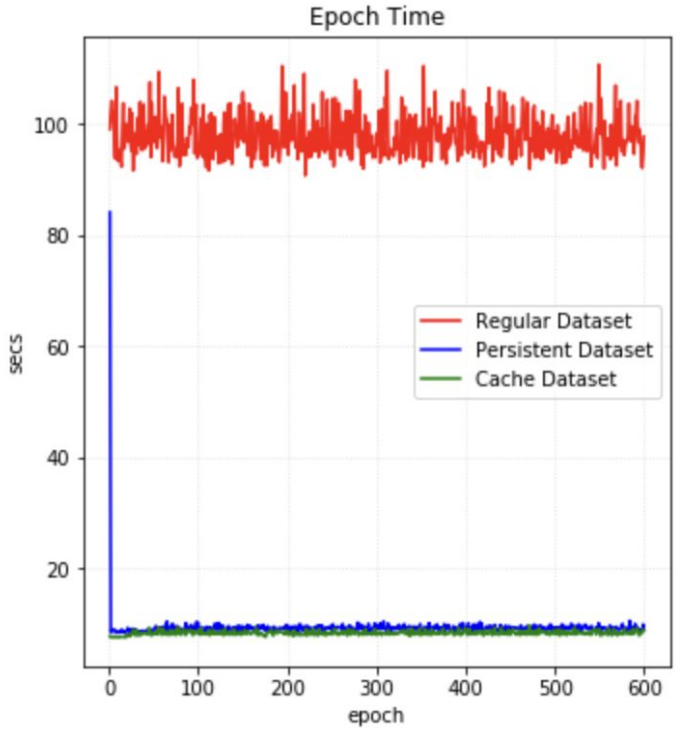
Modules: Datasets & Data Loaders

SmartCacheDataset, PersistentDataset, ...



Modules: Datasets & Data Loaders





Modules: Datasets & Data Loaders

Predefined datasets for public medical data

- MedNISTDataset
  - 58k small images (64x64 px) splitted to 6 categories: AbdomenCT, HeadCT, Hand, CXR, BreastMRI, ChestCT

- DecathlonDataset
  - 2.5k 3D images with 10 different semantic segmentation tasks (CT & MRI for different organs)

Modules: Losses & Optimizers

- Domain-specific loss functions in the medical imaging research:
  - DiceLoss, GeneralizedDiceLoss, MaskedDiceLoss, TverskyLoss, FocalLoss, DiceCELoss

- LearningRateFinder
  - Convinience tool that finds “good” learning rate

Modules: Network architectures

- Ready-to-use popular neural network building blocks and layers
  - Convolution blocks, Squeeze-and-Excitation, Transformer blocks,...
- Implementations of: UNet, DynUNet, DenseNet, GAN, AHNet, VNet, SENet, SegResNet, EfficientNet
  - All networks are 2D and 3D data compatible
  - Almost no pretrained models! Only 3 architecture types have proper pre-trained weights and all of them come from Torchvision
- Torchvision model adapters

Data loading usage example: part 1

- Here is example of data loading using MONAI
- It is based on CheXpert dataset, that contains X-ray image data with 14 labels for each image

```
loader = CheXpertSmallDatasetLoader(
    dataset_dir=Path("/home/alltr/Downloads/CheXpert-v1.0-small"),
    uncertain_as="positive",
    unmentioned_as_negative=True
)
```

```
next(iter(loader.train_dicts))
```

```
{'img': '/home/alltr/Downloads/CheXpert-v1.0-small/train/patient00001/study1/view1_frontal.jpg',
 'labels': tensor([0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1.])}
```

Data loading usage example: part 2

```
from monai import transforms

transform = transforms.Compose(
    [
        transforms.LoadImaged(keys='img'),
        transforms.AddChanneld(keys='img'),
        transforms.HistogramNormalized(
            keys='img',
            num_bins=256,
            min=0,
            max=1
        ),
        transforms.Resized(
                keys='img',
                spatial_size=(128, 128),
                mode='bilinear',
                align_corners=False,
        ),
    ]
)
```

```
train_dataset = monai.data.Dataset(
            data=loader.train_dicts,
            transform=transform
        )
```

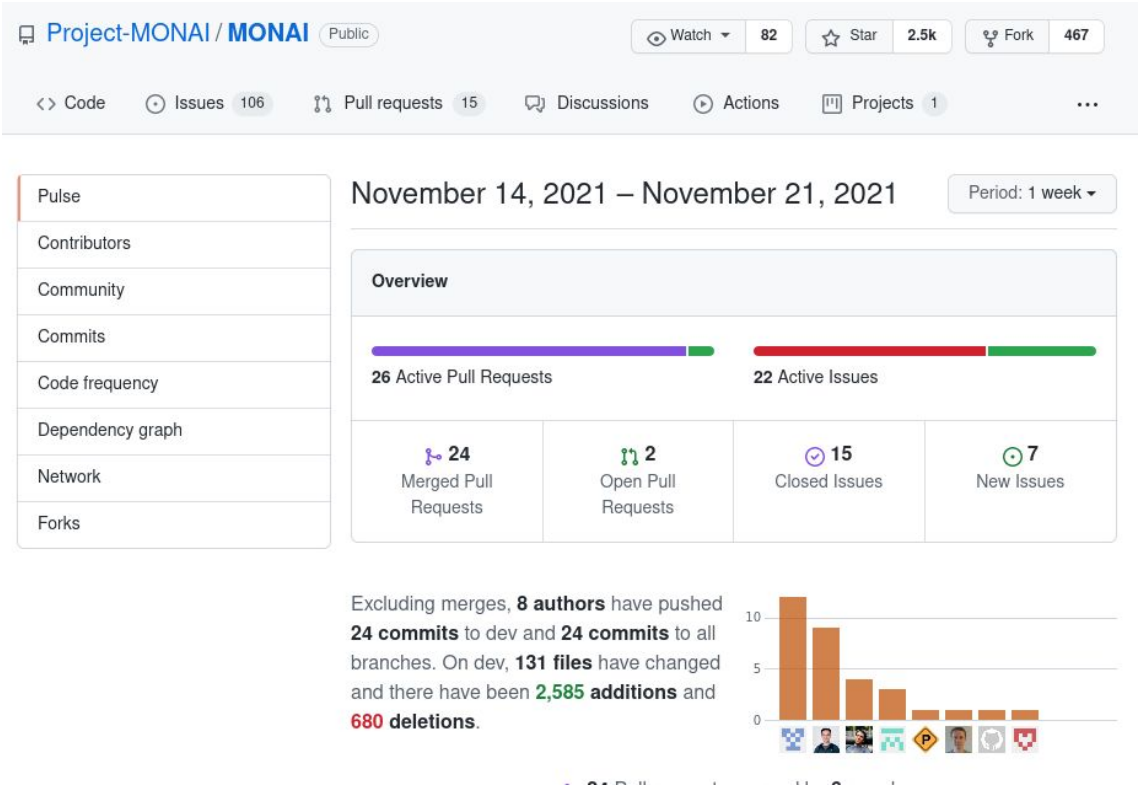
Data loading usage example: part 3

```
train_dataset[0]
```

```
{'img': array([[[4.8967707e-03, 1.1496857e-02, 7.5493730e-03, ...,
          5.9709007e-01, 6.2116456e-01, 6.9390053e-01],
         [1.8202408e-01, 2.2265975e-01, 2.2646448e-01, ...,
          5.8236504e-01, 6.5855801e-01, 6.6892320e-01],
         [1.9316415e-01, 5.4358566e-01, 2.7534008e-01, ...,
          6.5688235e-01, 6.8536568e-01, 6.9557846e-01],
         ...,
         [0.0000000e+00, 0.0000000e+00, 1.2426265e-03, ...,
          9.3348217e-01, 9.1857481e-01, 9.6831173e-01],
         [1.1698164e-03, 0.0000000e+00, 7.2810144e-05, ...,
          9.4151521e-01, 9.9126703e-01, 9.4060457e-01],
         [1.2426265e-03, 0.0000000e+00, 0.0000000e+00, ...,
          9.6733844e-01, 9.6048635e-01, 9.1872561e-01]]], dtype=float32),
 'labels': tensor([0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1.]),
 'img_meta_dict': {'format': 'JPEG',
  'mode': 'L',
  'width': 389,
  'height': 320,
  'spatial_shape': array([389, 320]),
  'original_channel_dim': 'no_channel',
  'filename_or_obj': '/home/alltr/Downloads/CheXpert-v1.0-small/train/patient00001/study1/view1_frontal.jpg'},
 'img_transforms': [{'class': 'Resized',
   'id': 140364319217024,
   'orig_size': (389, 320),
   'extra_info': {'mode': 'bilinear', 'align_corners': False}}]}
```

MONAI is still in the development

Current stable version is 0.7



Subjective overview of the package

- Great data I/O tools
- A lot of small utilities for optimization
- Mature enough for the serious usage
- Lacking in the pretrained models
- Lacking in ready-to-use datasets (tensorflow-datasets)

Thank You!

Q & A