

fastMONAI

Simplifying deep learning for medical image analysis



Western Norway
University of
Applied Sciences



MMIV HELSE VEST





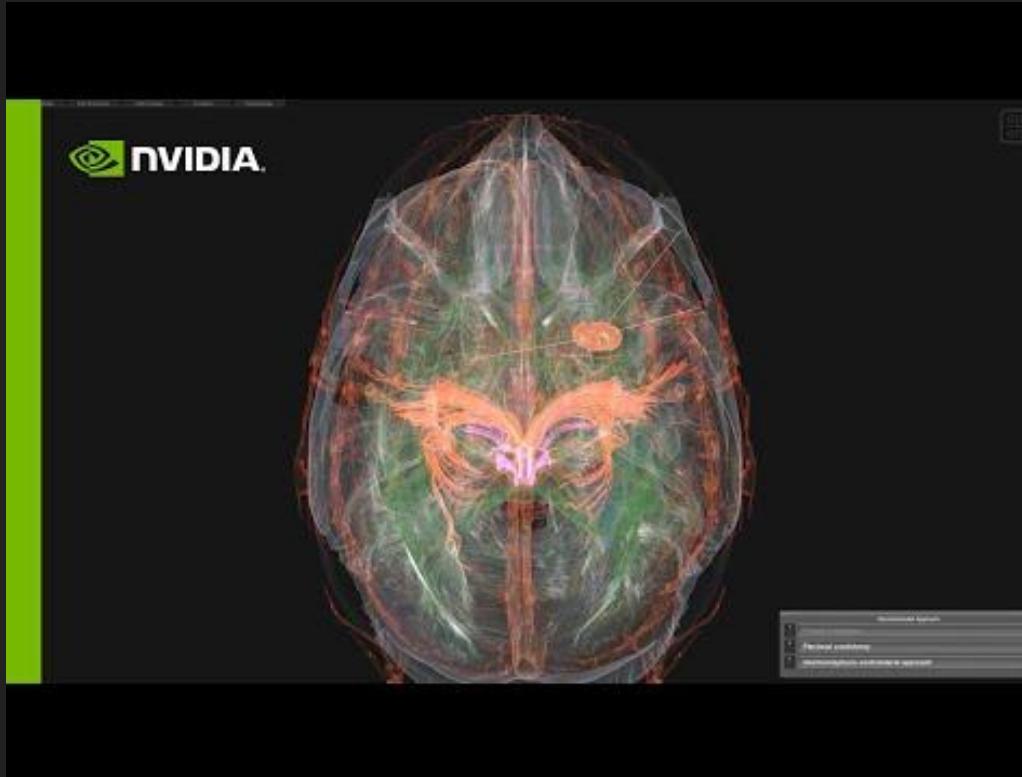




fastai + MONAI = fastMONAI

- **MONAI**
 - Pytorch-based framework for deep learning in 2D and 3D medical imaging
- **TorchIO**
 - Python library for efficient loading, preprocessing and augmentation of 3D medical images
- **fastai**
 - Pytorch-based low-code library for 2D images utilizing state-of-the-art approaches in various domains
- **fastMONAI**
 - Utilizing multiple state-of-the-art techniques in the area of 3D medical image analysis

Why MONAI?



Announced MONAI Core v1.0 few days ago


```

from fastMONAI.vision_core import *
from fastMONAI.vision_data import *
from fastMONAI.vision_augmentation import *
from fastMONAI.vision_metrics import *
from fastai.vision.all import *

NB_DIR = Path.cwd()
STUDY_DIR = NB_DIR/'..''/..''/data''/EC-data-Sat'

df = pd.read_csv(STUDY_DIR/'dataset.csv')
df.shape
(166, 6)

dataset_obj = MedDataset(img_list=df.mask_path.tolist(), dtype=NiiMask ,max_workers=12)

The volumes in this dataset have different orientations. Recommended to pass in the argument reorder=True when creating a MedDataset object for this dataset

data_info_df = dataset_obj.summary()
data_info_df.head()

dim_0 dim_1 dim_2 voxel_0 voxel_1 voxel_2 orientation
7 232 256 88 0.9766 0.9766 1.2000 LPS+
4 192 192 48 1.3021 1.3021 2.0000 RAS+
6 232 256 88 0.9766 0.9766 1.2000 LIP+
3 192 192 48 1.3021 1.3021 2.0000 LPS+
0 192 48 192 1.3021 2.0000 1.3021 RAS+ nnU-Net: Self-adapting Framework
example_path total
/home/sathish/machine_learning/fastMONAI/preprocess_nbs/././data/EC-data-Sat/367/segmented/367segmentedJulie.nii.gz 28
/home/sathish/machine_learning/fastMONAI/preprocess_nbs/././data/EC-data-Sat/011/segmented/011segmentedJulie.nii.gz 26
/home/sathish/machine_learning/fastMONAI/preprocess_nbs/././data/EC-data-Sat/372/segmented/372segmentedJulie.nii.gz 25
/home/sathish/machine_learning/fastMONAI/preprocess_nbs/././data/EC-data-Sat/017/segmented/017segmentedJulie.nii.gz 18
/home/sathish/machine_learning/fastMONAI/preprocess_nbs/././data/EC-data-Sat/021/segmented/021segmentedJulie.nii.gz 17

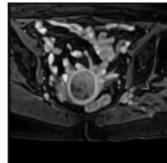
_, reorder = dataset_obj.suggestion()
resample = [1,1,1]

dblock = NiiDataBlock(blocks=(ImageBlock(cls=NiiImage), NiiMaskBlock), splitter=RandomSplitter(seed=42),get_x=ColReader('img_path'),get_y=ColReader('mask_path'),
item_tfms=[ZNormalizationTIO(), ImagePadOrCropTIO([256,256,256])],reorder=reorder,resample=resample)

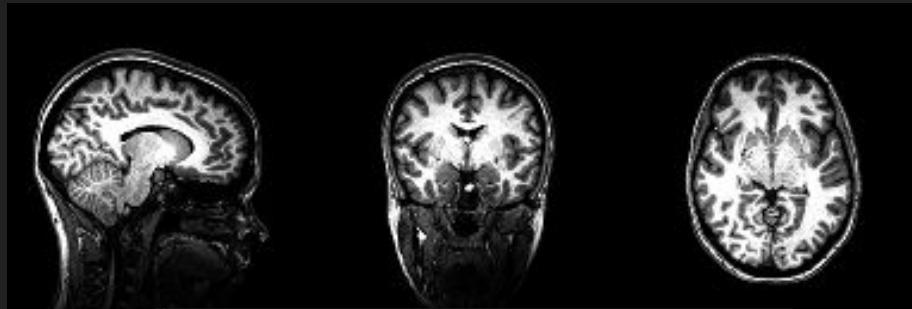
dls = dblock.dataloaders(df, bs=4)

dls.show_batch(figsize=(10, 10), anatomical_plane='A')

```



Deep learning for 3D images?

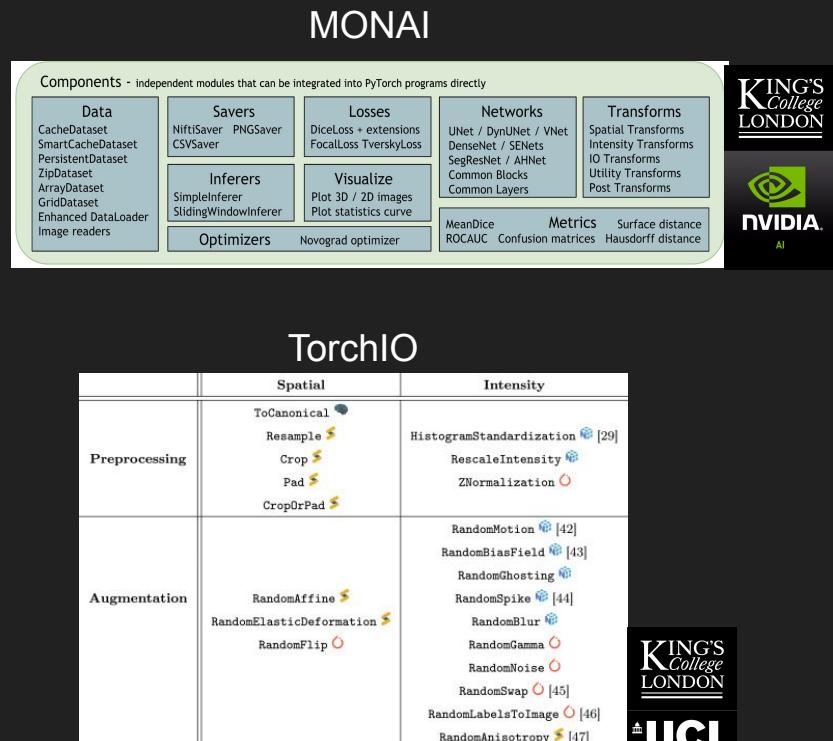
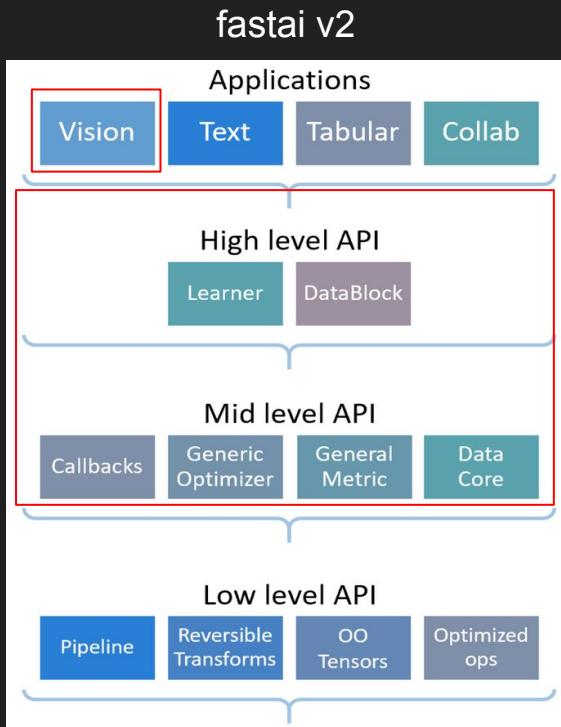


```
img.shape
```

```
torch.Size([1, 194, 256, 256])
```

Not supported in fastai

fastMONAI



fastMONAI



Learner

Callbacks

Datablock

Optimizers



Networks

Metrics



Transforms

Well-documented

The screenshot shows the MONAI documentation page for the UNETR model. It includes a search bar, a navigation menu with links like 'What's New', 'Highlights', 'API Reference', 'Installation Guide', 'Development', and 'More'. Below the menu, there's a section titled 'Return type: `Tensor`' with a note about the `UNet` instance referring to the former. The main content area displays the `class nn.Module` code for `UNET`, which defines a class with various layers and parameters. At the bottom, there are examples of code snippets for different use cases.

The screenshot shows the fastai documentation site. The top navigation bar includes links for 'Installing', 'About fastai', 'Migrating from other libraries', 'Windows Support', 'Tests', 'Contributing', 'Docker Containers', and a search bar. The main content area features a large heading 'Welcome to fastai' with the subtext 'fastai simplifies training fast and accurate neural nets using modern best practices'. Below this are sections for 'Quick start', 'Tutorials', 'Interpretation of Predictions', 'Data', 'Core', and 'Transforms'. A sidebar on the right lists 'On this page' sections such as 'Getting started', 'Data structures', 'Batch-based pipelines', 'Transforms', 'Preprocessing', 'Augmentation', 'Others', and 'Medical image datasets'. There are also sections for 'Additional interfaces', 'Examples gallery', and 'GitHub repository'.

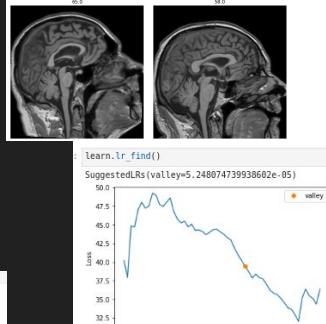
The screenshot shows the TorchIO documentation site. The top navigation bar includes links for 'Installing', 'About fastai', 'Migrating from other libraries', 'Windows Support', 'Tests', 'Contributing', 'Docker Containers', and a search bar. The main content area features a large heading 'Welcome to fastai' with the subtext 'fastai simplifies training fast and accurate neural nets using modern best practices'. Below this are sections for 'Quick start', 'Tutorials', 'Interpretation of Predictions', 'Data', 'Core', and 'Transforms'. A sidebar on the right lists 'On this page' sections such as 'Getting started', 'Data structures', 'Batch-based pipelines', 'Transforms', 'Preprocessing', 'Augmentation', 'Others', and 'Medical image datasets'. There are also sections for 'Additional interfaces', 'Examples gallery', and 'GitHub repository'.

fastMONAI has step-by-step tutorials using open datasets

Classification



Regression



Segmentation

```

from fastmri.vision import import_
from monai.data import DecathlonDataset
from sklearn.model_selection import train_test_split

Download external data
We use the MONAI function DecathlonDataset to download the data and generate items for training.

path = Path("./data")
path.mkdir(exist_ok=True)

training_data = DecathlonDataset(root=path, challenge="BrainTumour", section="training", download=True,
                                 num_workers=16)

2022-09-01 17:36:43,099 - INFO - Verified 'Task01_BrainTumour.tgz' file exists: ./data/Task01_BrainTumour/
2022-09-01 17:36:43,100 - INFO - Non-empty folder exists in ./data/Task01_BrainTumour, skipping extracting.

df = pb.Dataset.from_dataframe(data=data)
df.shape

(388, 2)

Splt the labeled data into training and test

train_df, test_df = train_test_split(df, test_size=0.1, random_state=42)
train_df.shape, test_df.shape

(340, 2), (38, 2)

Look at training data

med_dataset = MedDataset(img_list=train_df['label'].tolist(), dtype=MedMask, max_workers=12)

summary = med_dataset.summary()

summary[0].head()

dim_0 dim_1 dim_2 voxel_n voxel_1 voxel_2 orientation example_path total
0 100 240 155 10 10 10 RAD51noseithBRB_devtastMONAUdata7s01_BrainTumourLabelsTrBRBS_302.nii.gz 348
```

resample, reorder = med_dataset.suggestion()
resample, reorder
(1.0, 1.0, 1.0, False)
img_size = med_dataset.get_largest_img_size(resample=resample)
img_size
[240, 240, 155, 0]
bs=4
size=[224, 224, 128]
item_tfm = [Normalize(), PadOrCropsize(), RandomAffine(scales=0.6, degrees=5, isotropic=True)]
dblock = MedDataBlock(blocks=[ImageBlock(modality='ModT1g'), MedianBlock],
 get_y=True, get_x=True, get_xc=True, get_yc=True, get_xr=True,
 get_yr=True, get_xc_r=True, get_yc_r=True, recompute_reorder=True,
 resample=resample)
dts = dblock.dataseter(train_df, bs=bs)
training and validation
(len(dts.train_dts.items), len(dts.valid_dts.items))
(200, 68)
dts.show_batch(mosaic=True)



Scientific publications

Screenshot of a GitHub repository for "Pulmonary nodule classification in lung cancer from 3D thoracic CT scans".

Repository details:

- Branch: master
- Commits: 22
- Last commit: c55bd2f on Feb 1, 2021

File list:

- figures
- notebooks
- src
- README.md
- environment.yml

README.md content:

Pulmonary nodule classification in lung cancer from 3D thoracic CT scans

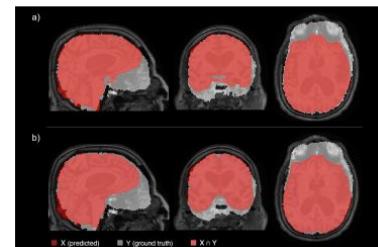
CNN model diagram:

Preprocessing: 3x3x3 dilation, Normalization and center of mass.

Data source: A1, A2, A3, A4.

Visual explanations: CAM, Grad CAM.

2D and 3D U-Nets for skull stripping in a large and heterogeneous set of head MRI using fastai



This code is written by Alexander Selvikvåg Lundervold and Satheshkumar Kalyugarasan .

V1

Screenshot of a GitHub repository for "Fully automatic whole-volume tumor segmentation in cervical cancer".

Repository details:

- Branch: main
- Commits: 3
- Last commit: f7181f3 on Mar 22

File list:

- figs
- notebooks
- src
- ignore
- README.md
- environment.yml

README.md content:

Fully automatic whole-volume tumor segmentation in cervical cancer

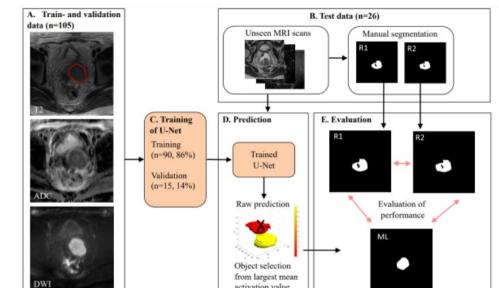
A. Train+validation data (n=105)

B. Test data (n=26)

C. Training of U-Net

D. Prediction

E. Evaluation



V2

Other projects are currently being pursued

Let's look at how we developed fastMONAI

Nbdev: tool for exploratory programming

Create delightful software with Jupyter Notebooks

Write, test, document, and distribute software packages and technical articles — all in one place, your notebook.

[Get started](#)

Trusted in industry

NETFLIX transform Outerbounds NOVETTA

Netflix Technology Blog
Aug 16, 2018 · 13 min read · Listen

[...](#)

Beyond Interactive: Notebook Innovation at Netflix

By [Michelle Ufford](#), [M Pacer](#), [Matthew Seal](#), and [Kyle Kelley](#)

Notebooks have rapidly grown in popularity among data scientists to become the de facto standard for quick prototyping and exploratory analysis. At Netflix, we're pushing the boundaries even further, reimagining what a notebook can be, who can use it, and what they can do with it. And we're making big investments to help make this vision a reality.

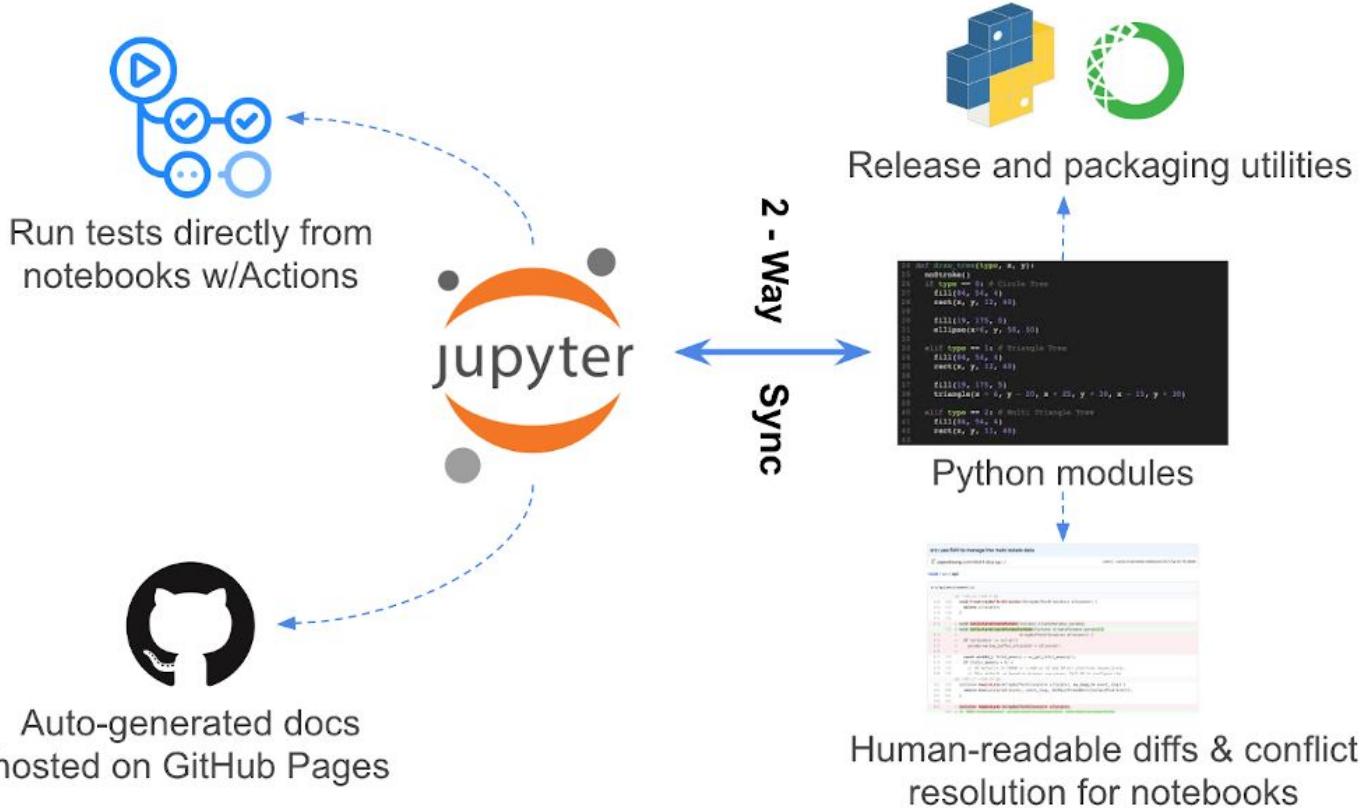
In this post, we'll share our motivations and why we find Jupyter notebooks so compelling. We'll also introduce components of our notebook infrastructure and explore some of the novel ways we're using notebooks at Netflix.

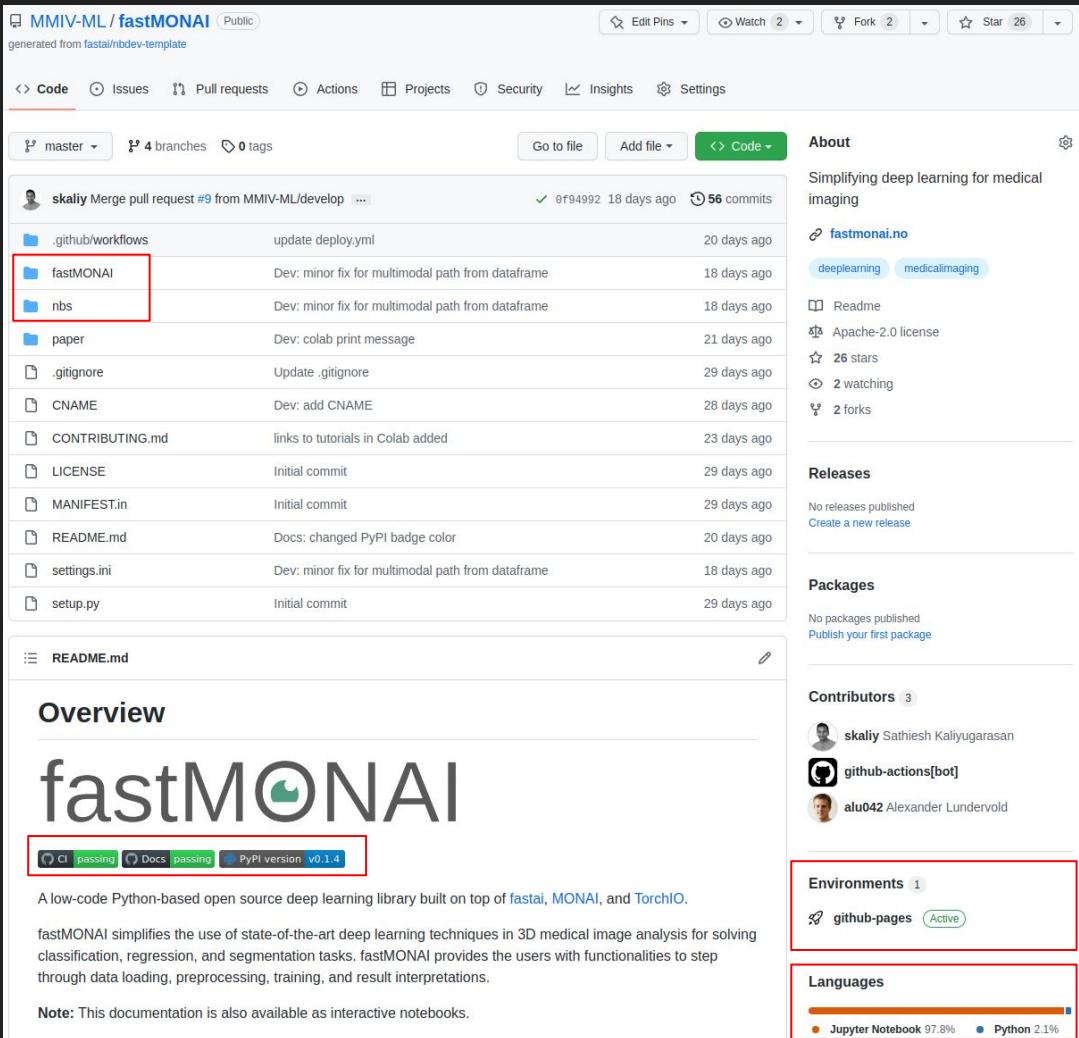
If you're short on time, we suggest jumping down to the Use Cases section.

David Berg
Software Engineer, Netflix

Prior to using nbdev, documentation was the most cumbersome aspect of our software development process... Using nbdev allows us to spend more time creating rich prose around the many code snippets guaranteeing the whole experience is robust.

nbdev has turned what was once a chore into a natural extension of the notebook-based testing we were already doing.



A screenshot of a GitHub repository page for "MMIV-ML/fastMONAI". The repository is public and was generated from "fastai/nbdev-template". The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The repository has 4 branches and 0 tags. A pull request by skaliy titled "Merge pull request #9 from MMIV-ML/develop" has been merged, showing 56 commits. The commit history lists changes in .github/workflows, fastMONAI, and nbs. The fastMONAI folder is highlighted with a red box. Other files listed include .gitignore, CNAME, CONTRIBUTING.md, LICENSE, MANIFEST.in, README.md, settings.ini, and setup.py. The README.md file contains an overview of fastMONAI, which is described as a low-code Python-based open source deep learning library built on top of fastai, MONAI, and TorchIO. It highlights its use for state-of-the-art deep learning techniques in 3D medical image analysis. The page also shows CI status (CI passing), documentation (Docs passing), and PyPI version (v0.1.4). It includes sections for Environments (github-pages Active), Languages (Jupyter Notebook 97.8%, Python 2.1%), and Contributors (skaliy, github-actions[bot], alu042). The right sidebar provides information about the repository, such as simplifying deep learning for medical imaging, the Apache-2.0 license, and 26 stars.

MMIV-ML / **fastMONAI** Public

generated from fastai/nbdev-template

<> Code Issues Pull requests Actions Projects Security Insights Settings

master 4 branches 0 tags Go to file Add file <> Code

skaliy Merge pull request #9 from MMIV-ML/develop ... ✓ 0f94992 18 days ago 56 commits

.github/workflows update deploy.yml 20 days ago

fastMONAI Dev: minor fix for multimodal path from dataframe 18 days ago

nbs Dev: minor fix for multimodal path from dataframe 18 days ago

paper Dev: colab print message 21 days ago

.gitignore Update .gitignore 29 days ago

CNAME Dev: add CNAME 28 days ago

CONTRIBUTING.md links to tutorials in Colab added 23 days ago

LICENSE Initial commit 29 days ago

MANIFEST.in Initial commit 29 days ago

README.md Docs: changed PyPi badge color 20 days ago

settings.ini Dev: minor fix for multimodal path from dataframe 18 days ago

setup.py Initial commit 29 days ago

README.md

Overview

fastMONAI

CI passing Docs passing PyPI version v0.1.4

A low-code Python-based open source deep learning library built on top of [fastai](#), [MONAI](#), and [TorchIO](#).

fastMONAI simplifies the use of state-of-the-art deep learning techniques in 3D medical image analysis for solving classification, regression, and segmentation tasks. fastMONAI provides the users with functionalities to step through data loading, preprocessing, training, and result interpretations.

Note: This documentation is also available as interactive notebooks.

About

Simplifying deep learning for medical imaging

fastmonai.no

deeplearning medicalimaging

Readme Apache-2.0 license 26 stars 2 watching 2 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Contributors 3

skaliy Sathiesh Kaliyugaran
github-actions[bot]
alu042 Alexander Lundervold

Environments 1

github-pages Active

Languages

Jupyter Notebook 97.8% Python 2.1%

Notebooks folder

master		fastMONAI / nbs /	Go to file	Add file	...
 skaliv	Dev:	minor fix for multimodal path from dataframe	✓ f685928	18 days ago	
..					
 images		Added all files		29 days ago	
 .gitignore		Added all files		29 days ago	
 00_vision_core_plot.ipynb		updated README		29 days ago	
 01_vision_core.ipynb		Dev: minor fix for multimodal path from dataframe		18 days ago	
 02_vision_data.ipynb		Added all files		29 days ago	
 03_vision_augment.ipynb		Dev: fixed fastai version		25 days ago	
 04_vision_loss_functions.ipynb		Added all files		29 days ago	
 05_vision_metrics.ipynb		Added all files		29 days ago	
 06_utils.ipynb		Dev: colab print message		21 days ago	
 07_dataset_info.ipynb		Added all files		29 days ago	
 08_external_data.ipynb		Dev: Colab compatibility		22 days ago	
 09a_tutorial_classification.ipynb		Dev: colab print message		21 days ago	
 09b_tutorial_regression.ipynb		Dev: colab print message		21 days ago	
 09c_tutorial_binary_segmentation.ipynb		Dev: colab print message		21 days ago	
 09d_tutorial_multiclass_segmentation.ipynb		Dev: colab print message		21 days ago	
 09e_tutorial_inference.ipynb		Dev: Colab compatibility		22 days ago	

Notebooks folder

master fastMONAI / nbs /

skaliv Dev: minor fix for multimodal path from dataframe ✓ f685928 18 days ago ⏲ History

```
[ ]: #! default_exp vision_core
```

```
[ ]: #! hide
from nbdev.showdoc import *
```

```
[ ]: #! export
from fastMONAI.vision_plot import *
from fastai.data.all import *
from torchio import ScalarImage, LabelMap, ToCanonical, Resample
import pickle
import warnings
```

```
***
```

```
***
```

Load images

```
***
```

```
***
```

```
[ ]: #! export
def med_img_reader(fn:(str, Path), # Image path
                  dtype=torch.Tensor, # Datatype (MedImage, MedMask, torch.Tensor)
                  resample:list=None, # Wheter to resample image to different voxel sizes and image dimensions.
                  reorder:bool=False, # Wheter to reorder the data to be closest to canonical (RAS+) orientation.
                  only_tensor:bool=True # Wheter to return only image tensor. If False, return ScalarImage or LabelMap object.
                  ):
    '''Load a medical image data. 4D tensor is returned as `dtype` if `only_tensor` is True, otherwise return ScalarImage or LabelMap.'''
    if ';' in fn:
        img_fns = fn.split(';')
        return multi_channel(img_fns, reorder, resample, dtype=dtype)
        #add return metadata option to get original information

    o = _load(fn, dtype=dtype)
    o, org_metadata = _preprocess(o, reorder, resample)

    if only_tensor: return dtype(o.data.type(torch.float))

    return o, org_metadata
```

Go to file Add file ...

Source code

fastMONAI / fastMONAI /		
		Go to file
		Add file
...		...
 skaliy	Dev: minor fix for multimodal path from dataframe	✓ f685928 18 days ago
..		History
 __init__.py	Dev: minor fix for multimodal path from dataframe	18 days ago
 _modidx.py	Dev: minor fix for multimodal path from dataframe	18 days ago
 core.py	init	29 days ago
 dataset_info.py	Added all files	29 days ago
 external_data.py	Dev: Colab compatibility	22 days ago
 utils.py	Dev: colab print message	21 days ago
 vision_all.py	Dev: Colab compatibility	22 days ago
 vision_augmentation.py	Dev: fix of docstrings for vision_augment	26 days ago
 vision_core.py	Dev: minor fix for multimodal path from dataframe	18 days ago
 vision_data.py	Added all files	29 days ago
 vision_loss.py	Added all files	29 days ago
 vision_metrics.py	Added all files	29 days ago
 vision_plot.py	Added all files	29 days ago

Source code

master fastMONAI/fastMONAI/

Go to file Add file ...

```
# %% ..../nbs/01_vision_core.ipynb 8
def med_img_reader(fn: str, Path,
                   dtype=torch.Tensor, # Datatype (MedImage, MedMask, torch.Tensor)
                   resample:list=None, # Wheter to resample image to different voxel sizes and image dimensions.
                   reorder:bool=False, # Wheter to reorder the data to be closest to canonical (RAS+) orientation.
                   only_tensor:bool=True # Wheter to return only image tensor. If False, return ScalerImage or LabelMap object.
                   ):
    """Load a medical image data. 4D tensor is returned as `dtype` if `only_tensor` is True, otherwise return ScalerImage or LabelMap."""
    if ';' in fn:
        img_fns = fn.split(';')
        return _multi_channel(img_fns, reorder, resample, dtype=dtype)
        #add return metadata option to get original information

    o = _load(fn, dtype=dtype)
    o, org_metadata = _preprocess(o, reorder, resample)

    if only_tensor: return dtype(o.data.type(torch.float))

    return o, org_metadata
```

Documentation page: fastmonai.no

Overview
Vision core plot
Vision core
Vision data
Data augmentation
Custom loss functions
Vision metrics
Utils
Dataset information
External data
Tutorials
Single label classification
Regression
Binary semantic segmentation
Multi-class semantic segmentation

[fastMONAI](#) is a low-code Python-based open source deep learning library built on top of [fastai](#), [MONAI](#), and [TorchIQ](#). fastMONAI simplifies the use of state-of-the-art deep learning techniques in 3D medical image analysis for solving classification, regression, and segmentation tasks. fastMONAI provides the users with functionalities to step through data loading, preprocessing, training, and result interpretations.

Note: This documentation is also available as interactive notebooks.

Installing

From PyPI

```
pip install fastMONAI
```

From Github

If you want to install an editable version of fastMONAI run:

- `git clone https://github.com/MMIV-ML/fastMONAI`
- `pip install -e 'fastMONAI[dev]'`

Getting started

The best way to get started using fastMONAI is to read the [paper](#) and look at the step-by-step tutorial-like notebooks to learn how to train your own models on different tasks (e.g., classification, regression, segmentation). See the docs at <https://fastmonai.no> for more information.

1-Click	Notebook
Notebook	Notebook
09a_tutorial_classification.ipynb	Open in Colab

shows how to construct a binary classification model based on MRI data.

Documentation page: [fastmonai.no](#)

Overview
Vision core plot
Vision core
Vision data
Data augmentation
Custom loss functions
Vision metrics
Utils
Dataset information
External data
Tutorials >
Single-label classification
Regression
Binary semantic segmentation
Multi-class semantic segmentation

Vision core

Load images

med_img_reader

[source](#)

```
med_img_reader (fn:<class'str'>,<class'pathlib.Path'>), dtype=<class
    'torch.Tensor'>, resample:list=None, reorder:bool=False,
    only_tensor:bool=True)
```

Load a medical image data. 4D tensor is returned as `dtype` if `only_tensor` is True, otherwise return ScalerImage or LabelMap.

Type	Default	Details
fn	(<class 'str'>, <class 'pathlib.Path'>)	Image path
dtype	_TensorMeta	Datatype (MedImage, MedMask, torch.Tensor)
resample	list	Wheter to resample image to different voxel sizes and image dimensions.
reorder	bool	Wheter to reorder the data to be closest to canonical (RAS+) orientation.
only_tensor	bool	Wheter to return only image tensor. If False, return ScalerImage or LabelMap object.

On this page

- Load images
- med_img_reader**
- MetaResolver
- MedBase
- MedImage
- MedMask

[Report an issue](#)

Documentation page: fastmonai.no

Overview
Vision core plot
Vision core
Vision data
Data augmentation
Custom loss functions
Vision metrics
Utils
Dataset information
External data
Tutorials >
Single-label classification
Regression
Binary semantic segmentation
Multi-class semantic segmentation

Vision metrics

binary_dice_score [source](#) [Report an issue](#)

Calculate the mean Dice score for binary semantic segmentation tasks.

Type	Details
act	Activation tensor [B, C, W, H, D]
targ	Target masks [B, C, W, H, D]

Returns **Tensor**

multi_dice_score [source](#)

Calculate the mean Dice score for each class in multi-class semantic segmentation tasks.

Type	Details
act	Activation values [B, C, W, H, D]
targ	Target masks [B, C, W, H, D]

Returns **Tensor**

Write tests in notebooks

Vision metrics

```
[ 1]: #| export
def _calculate_dsc(pred, targ):
    """ MONAI `compute_meandice`:
        https://docs.monai.io/en/stable/_modules/monai/metrics/meandice.html#compute_meandice
    """

    return torch.Tensor([compute_meandice(p[None], t[None]) for p, t in list(zip(pred,targ))])

[ 1]: #| export
def _calculate_haus(pred, targ):
    """ MONAI `compute_hausdorff_distance`:
        https://docs.monai.io/en/stable/_modules/monai/metrics/hausdorff_distance.html#compute_hausdorff_distance
    """

    return torch.Tensor([compute_hausdorff_distance(p[None], t[None]) for p, t in list(zip(pred,targ))])

[ 1]: #| export
def binary_dice_score(act, # Activation tensor [B, C, W, H, D]
                      targ # Target masks [B, C, W, H, D]
                      ) -> torch.Tensor:
    """Calculate the mean Dice score for binary semantic segmentation tasks."""

    pred = pred_to_binary_mask(act)
    dsc = _calculate_dsc(pred.cpu(), targ.cpu())

    return torch.mean(dsc)

***  

***  

***  

[ 1]: #| hide
# Test Dice score and Hausdorff distance
pred = torch.zeros((1,1,10,10,10))
pred[:, :, :5, :5, :5] = 1

targ = torch.zeros((1,1,10,10,10))
targ[:, :, :5, :5, :5] = 1

dsc = float(_calculate_dsc(pred, targ))
haus = float(_calculate_haus(pred,targ))

assert dsc == 1.0
assert haus == 0.0
```

Continuous integration with GitHub Actions

The screenshot shows a GitHub Actions CI run titled "Dev: minor fix for multimodal path from dataframe CI #64". The summary indicates a success status with a green checkmark icon. The "test" job is expanded, showing its history of steps:

- > ✓ Set up job
- > ✓ Run fastai/workflows/nbdev-ci@master
- > ✓ Post Run fastai/workflows/nbdev-ci@master
- > ✓ Complete job

Each step is marked with a green checkmark icon, indicating they all succeeded. The entire run was completed 18 days ago in 1m 50s.

Publish to PyPI

The screenshot shows the PyPI website interface. At the top, there is a search bar with the placeholder "Search projects" and a magnifying glass icon. To the right of the search bar are links for "Help", "Sponsors", "Log in", and "Register". Below the header, the title "fastMONAI 0.1.4" is displayed in large white text. To the right of the title is a green button with a checkmark icon and the text "Latest version". Below the title, there is a "pip install fastMONAI" button with a clipboard icon. On the left side of the page, there is a dark sidebar containing a configuration file snippet:

```
1 [DEFAULT]
2 # All sections below are required unless otherwise specified
3 # see https://github.com/fastai/nbdev/blob/master/settings.ini for examples
4
5 ### Python Library ###
6 lib_name = fastMONAI
7 min_python = 3.7
8 version = 0.1.4
9 ### OPTIONAL ###
10
11 requirements = fastai==2.7.9 monai==0.8.1 torchio==0.18.76 xlrd>=1.2.0
12 dev_requirements = ipywidgets nbdev tabulate
13
14 ### nbdev ###
15 nbs_path = nbs
16 doc_path = _docs
17 recursive = False
18 tst_flags = nostest
19
```

At the bottom right of the main content area, the text "Released: Sep 9, 2022" is visible.

1-click Colab notebooks

Getting started

The best way to get started using fastMONAI is to read the [paper](#) and look at the step-by-step tutorial-like notebooks to learn how to train your own models on different tasks (e.g., classification, regression, segmentation). See the docs at <https://fastmonai.no> for more information.

Notebook	1-Click Notebook
09a_tutorial_classification.ipynb shows how to construct a binary classification model based on MRI data.	 Open in Colab
09b_tutorial_regression.ipynb shows how to construct a model to predict the age of a subject from MRI scans ("brain age").	 Open in Colab
09c_tutorial_binary_segmentation.ipynb shows how to do binary segmentation (extract the left atrium from monomodal cardiac MRI).	 Open in Colab
09d_tutorial_multiclass_segmentation.ipynb shows how to perform segmentation from multimodal MRI (brain tumor segmentation).	 Open in Colab



TorchIO @TorchIOLib · Aug 31

Nice distillation of modern medical imaging libraries for deep learning. Tools like fastMONAI will be helpful for clinicians getting started with training neural nets on multidimensional medical images.

...

 **Sathiesh Kaliyugaranan** @skaliy3 · Aug 30

Finally! We have released version 0.1 of fastMONAI, a low-code Python-based deep learning library for medical imaging built on top of @fastdotai, @ProjectMONAI, and @TorchIOLib. You can install the library using pip and download the tutorial notebooks here:github.com/MMIV-ML/fastMO...

[Show this thread](#)



Jeremy Howard @jeremyphoward · Sep 1

Replies to [@skaliy3](#)

Awesome job!

Thank you!



Western Norway
University of
Applied Sciences



MMIV



TROND
MOHN
FOUNDATION



<https://skaliy.no>