MACQUARIE
University
SYDNEY · AUSTRALIA

CNS LAB

Practical Python workflows: from neuroimaging to quantitative features

Ghasem Azemi - WCCNS 2026

1st World Conference of
Computational Neurosurgery
13-15 February 2026
Sydney, Australia | Hyatt Regency Sydney

AESCULAP ACADEMY

Acknowledgements

The Computational NeuroSurgery (CNS) Lab

- ❑ **All pipelines presented** were developed and implemented in the **CNS Lab**

- ❑ Work conducted under the **supervision of Prof Di Ieva**

- ❑ **Focus on computational neuroimaging** for research and clinical translation

MACQUARIE University

Topics covered in this session:

- ❑ Managing computational environments
- ❑ Neuroradiomics pipeline
- ❑ Fractal-based features for brain imaging
- ❑ Predictive modelling and clinical interpretation
- ❑ Model explainability and building clinical trust
- ❑ Key takeaways

Before we begin

Scope and context

- ❑ **Radiomics pipelines** are demonstrated using **MRI**
- ❑ **Fractal-based analysis** is also shown for **MRI**
- ❑ All implementations are explained in **Python**
- ❑ **Colab notebook:** https://colab.research.google.com/drive/1I2N6Vo7Y6x1yH2FoZictvVjDfqc2Z7nC?usp=sharing
- ❑ **Google Drive files (required to run notebook):**
- ❑ https://drive.google.com/drive/folders/1LCTgspFYa-aDHI4kE0_miTYUh_jz0ECt?usp=sharing

Managing computational environments

Why reproducible software environments matter

Challenges in computational imaging

- ❑ Medical imaging analysis uses **many specialised tools**
- ❑ Each tool may require a **specific software version**
- ❑ Different computers often have **different setups**
- ❑ This can lead to:
  - o code that works on one machine but not another
  - o results that cannot be reproduced
- ❑ Goal:
  - 👉 Run the same analysis, get the same result, on any computer

Think of it like needing the correct scanner protocol to reproduce an MRI study

Setting up a reproducible software environment

Conda

- ❑ **Several tools exist** to install research software
- ❑ Not all tools manage **software versions and dependencies** equally well
- ❑ **Conda** is an environment and package manager
  - o A **package manager** locates and installs requested software and dependencies
  - o **Environments** are isolated collections of software, allowing different projects to use different versions safely

Conda is like sharing an MRI protocol instead of just describing it in words

Preparing your computing environment

Installing Conda

- ❑ **Miniconda (recommended)**: Lightweight installer with only Conda
  https://docs.conda.io/en/latest/miniconda.html

- ❑ **Anaconda**: Includes Conda + a large collection of preinstalled packages and tools
  https://docs.conda.io/projects/conda/en/stable/user-guide/install/index.html

- ❑ Not sure which one to choose?
  https://docs.conda.io/projects/conda/en/stable/user-guide/install/download.html#anaconda-or-miniconda

- ❑ **During installation**
  - Accept default settings
  - Allow Conda to initialise your shell when prompted

Setting up the workshop environment

Verify, create, and activate Conda env

- ❑ **Open a terminal** (macOS) or **Anaconda Powershell Prompt** (Windows)

- ❑ **Run** the following commands:

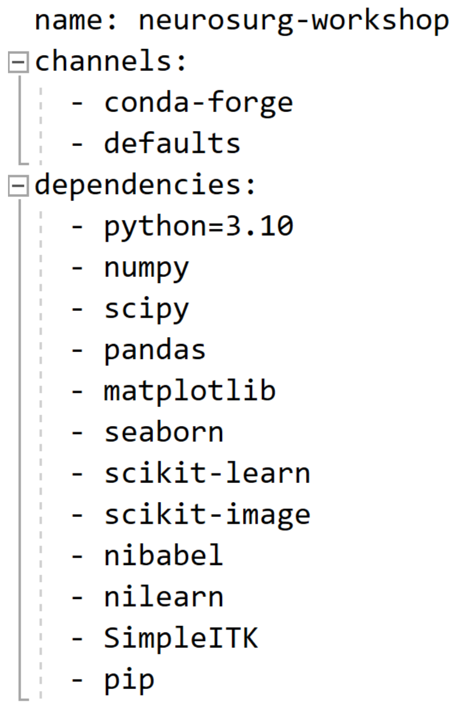
```
# Verify Conda installation
conda --version

# Create the workshop environment
conda env create -f neurosurg-workshop.yaml

# Activate the environment
conda activate neurosurg-workshop
```

Workshop Conda environment

Contents of the *.yaml* file



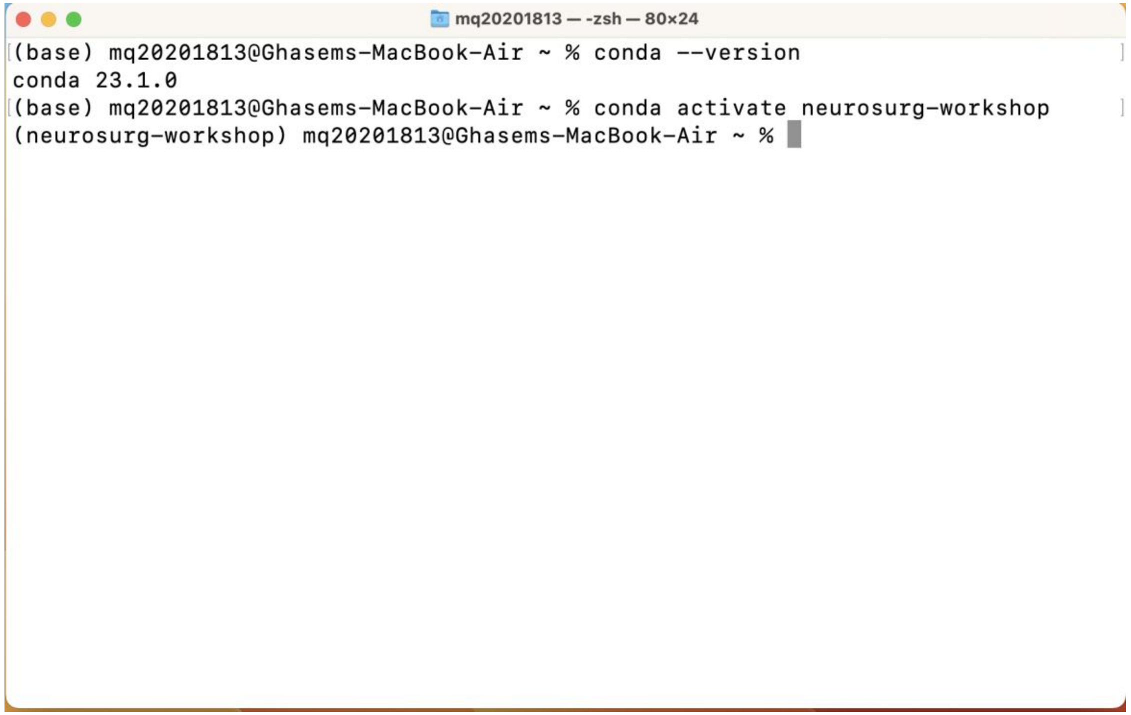
```yaml
name: neurosurg-workshop
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.10
  - numpy
  - scipy
  - pandas
  - matplotlib
  - seaborn
  - scikit-learn
  - scikit-image
  - nibabel
  - nilearn
  - SimpleITK
  - pip
```

Key points:

- ❑ **name:** environment name to activate
- ❑ **channels:** sources to download packages from
- ❑ **dependencies:** exact versions of Python and libraries for reproducibility

Terminal preview

Running Conda commands



mq20201813 — -zsh — 80×24

```
(base) mq20201813@Ghasems-MacBook-Air ~ % conda --version
conda 23.1.0
(base) mq20201813@Ghasems-MacBook-Air ~ % conda activate neurosurg-workshop
(neurosurg-workshop) mq20201813@Ghasems-MacBook-Air ~ %
```

MACQUARIE University

Computational framework for neuroradiomics

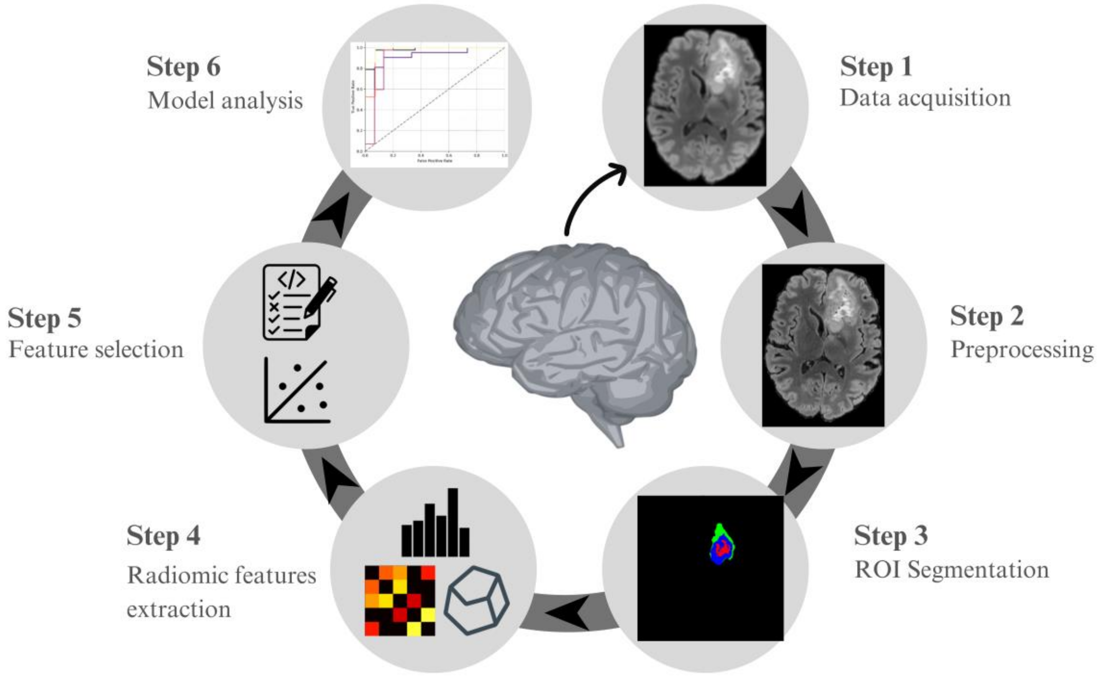
Computational neuroradiomics pipeline

What is neuroradiomics?

- ❑ Extracts **quantitative features** (intensity, texture, shape) from neuroimaging data
- ❑ Replaces subjective image reading **with objective, reproducible analysis**
- ❑ Supports **diagnosis, prognosis, and monitoring** in neurological and neurosurgical diseases
- ❑ Illustrated using **MRI**, but applicable to **CT, X-ray**, and other imaging modalities
- ❑ Pipeline implementation demonstrated using **Python-based tools**

Computational neuroradiomics pipeline

Typical neuroradiomics workflow



MRI preprocessing workflow

Key preprocessing steps

- ❑ **DICOM → NIfTI:** standardises raw scanner data for analysis
- ❑ **Bias field correction:** corrects intensity inhomogeneity across the image
- ❑ **Spatial normalisation:** aligns images to a common reference space
- ❑ **Skull stripping:** removes non-brain tissues
- ❑ (Optional) **Resampling to BraTS resolution:** required when using BraTS-trained tumour segmentation models

MRI preprocessing tools

Software used in this workshop

- **dcm2niix**
  Converts DICOM files to NIfTI format for downstream neuroimaging analysis
  https://github.com/rordenlab/dcm2niix

- **ANTs (Advanced Normalization Tools)**
  Provides state-of-the-art algorithms for bias field correction and image registration
  https://github.com/ANTsX/ANTs

- **FSL (FMRIB Software Library)**
  Offers widely used tools for skull stripping, registration, and MRI preprocessing
  https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/

Install required tools

Installation on macOS

- ❑ **Launch terminal and run the following commands:**

```
brew install dcm2niix
conda install -c conda-forge ants
curl -Ls https://fsl.fmrib.ox.ac.uk/fsldownloads/fslconda/releases/getfsl.sh | sh -s
```

- ❑ **Verify the installation of each tool:**

```
dcm2niix --version
fslmaths --help
antsRegistration --help
```

Executing MRI preprocessing

mri_preprocessing.sh

- ❑ **Open a terminal** (macOS) **or Anaconda PowerShell Prompt** (Windows)
- ❑ **Run** the preprocessing script by executing:

```
chmod +x mri_preprocessing.sh
./mri_preprocessing.sh
```

Preprocessing script contents

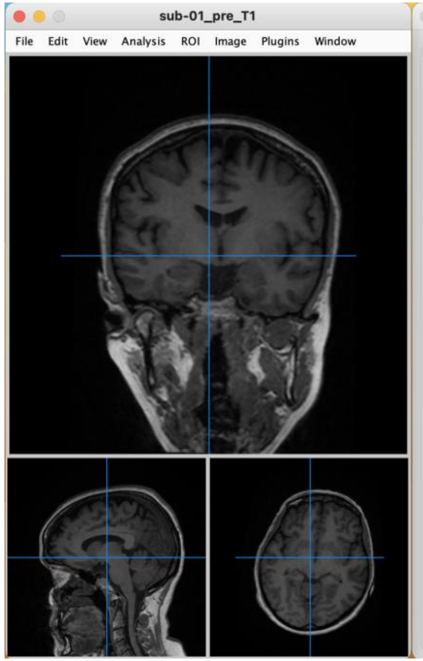
mri_preprocessing.sh

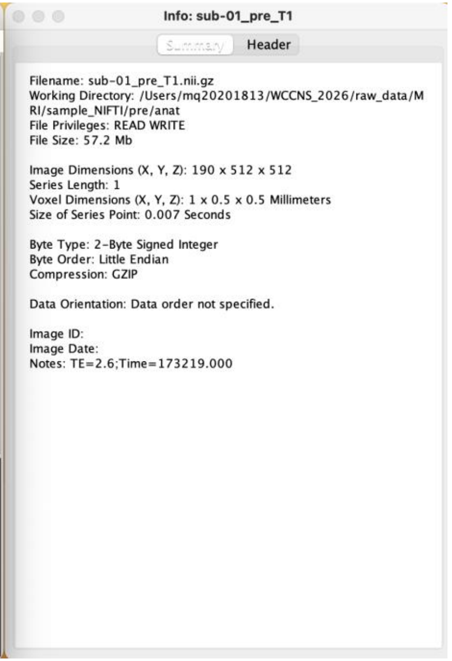
```
f="raw_data/MRI/sample_NIFTI/pre/anat"
template="$FSLDIR/data/standard/MNI152_T1_1mm.nii.gz"
brats_template="BraTS2021_00000_t1.nii.gz"
input_dicom="raw_data/MRI/sample_DICOM/pre/anat"

dcm2niix -z y -o "$f" "$input_dicom"
N4BiasFieldCorrection -d 3 -i "$f/sub-01_pre_T1.nii.gz" -s 3 -c [100x50x20,0.000001] \
-o "$f/sub-01_pre_T1_BFC.nii.gz"
flirt -in "$f/sub-01_pre_T1_BFC.nii.gz" -ref "$template" -out "$f/sub-01_pre_T1_BFC_norm.nii.gz" \
-omat "$f/sub-01_pre_T1_BFC_trans.mat"
bet "$f/sub-01_pre_T1_BFC_norm.nii.gz" "$f/sub-01_pre_T1_BFC_brain.nii.gz" -R -f 0.5 \
-g 0
antsApplyTransforms -d 3 -i "$f/sub-01_pre_T1_BFC_brain.nii.gz" -r "$brats_template" \
-o "$f/sub-01_pre-01_BFC_BraTS.nii.gz" -n Linear
```

Raw vs preprocessed MRI

Visual comparison and metadata (using Mango)





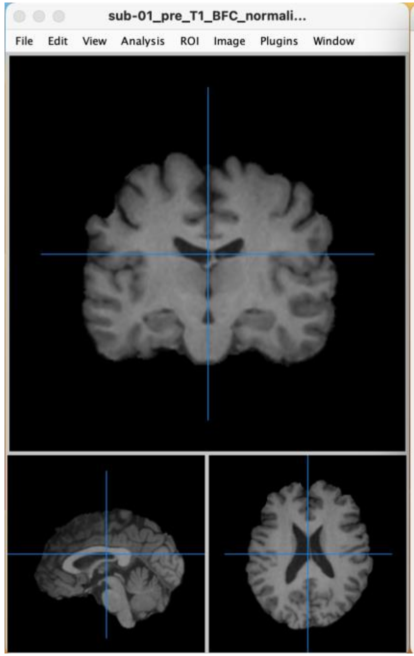
Filename: sub-01_pre_T1.nii.gz
Working Directory: /Users/mq20201813/WCCNS_2026/raw_data/MRI/sample_NIFTI/pre/anat
File Privileges: READ WRITE
File Size: 57.2 Mb

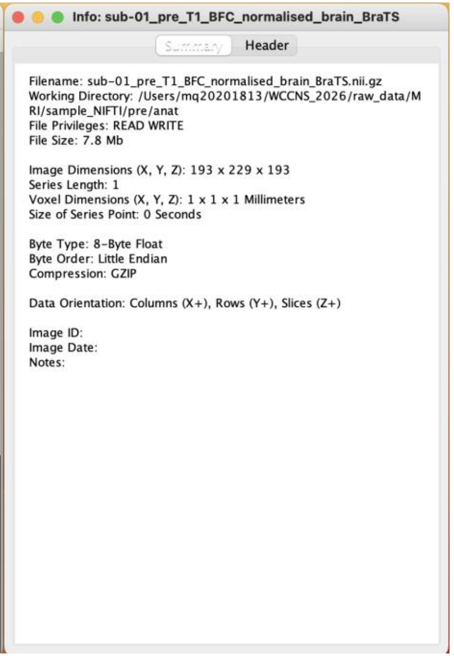
Image Dimensions (X, Y, Z): 190 x 512 x 512
Series Length: 1
Voxel Dimensions (X, Y, Z): 1 x 0.5 x 0.5 Millimeters
Size of Series Point: 0.007 Seconds

Byte Type: 2-Byte Signed Integer
Byte Order: Little Endian
Compression: GZIP

Data Orientation: Data order not specified.

Image ID:
Image Date:
Notes: TE=2.6;Time=173219.000


sub-01_pre_T1_BFC_normali...
File Edit View Analysis ROI Image Plugins Window



Filename: sub-01_pre_T1_BFC_normalised_brain_BraTS.nii.gz
Working Directory: /Users/mq20201813/WCCNS_2026/raw_data/MRI/sample_NIFTI/pre/anat
File Privileges: READ WRITE
File Size: 7.8 Mb

Image Dimensions (X, Y, Z): 193 x 229 x 193
Series Length: 1
Voxel Dimensions (X, Y, Z): 1 x 1 x 1 Millimeters
Size of Series Point: 0 Seconds

Byte Type: 8-Byte Float
Byte Order: Little Endian
Compression: GZIP

Data Orientation: Columns (X+), Rows (Y+), Slices (Z+)

Image ID:
Image Date:
Notes:

ROI delineation

Defining regions of interest

- ❑ **Identify** regions within neuroimaging data **for focused analysis**
- ❑ **In brain tumour studies,** ROIs often correspond to tumour subregions (enhancing core, necrotic core, oedema)
- ❑ Can be done **manually or fully automatically** using segmentation tools

Brain tumour segmentation

Example: BraTS 2021 winning model

- ❑ **Requirements:**

  1. **NVIDIA GPU with CUDA support**

  2. **Docker installed and configured**

  3. **PyTorch + CUDA (via Conda)**

  1. **nnunetv2 Python package**

Brain TS using BraTS 2021 winning model

Steps 1-3

❑ **Steps 1-3 to run the BraTS 2021 model:**

1. **Verify Docker:**

```
docker --version
```

2. **Check NVIDIA toolkit & CUDA:**

```
dpkg -l | grep nvidia-container-toolkit
nvcc --version
```

3. **Install PyTorch with CUDA:**

```
conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
```

Brain TS using BraTS 2021 winning model

Steps 4-6

- **Steps 4-6 to run the BraTS 2021 model:**

  4. **Install nnunetv2:**

     ```
     pip install nnunetv2
     ```

  5. **Pull Docker image:**

     ```
     docker pull rixez/brats21nnunet
     ```

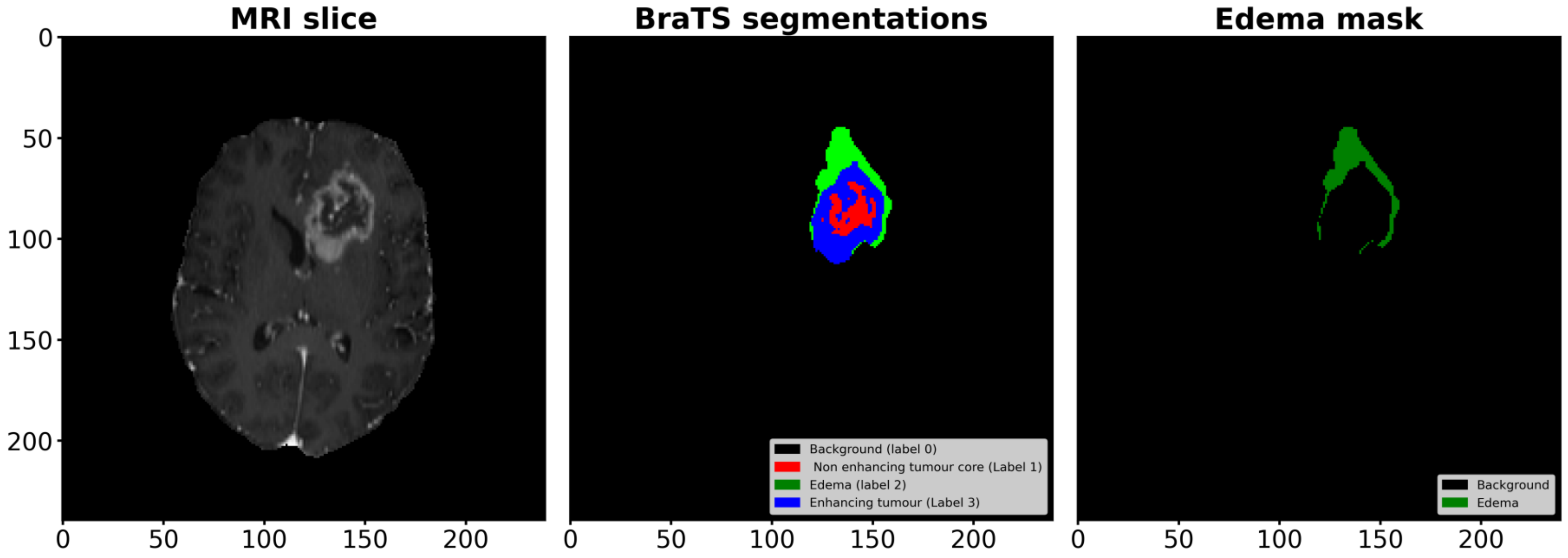
  4. **Run segmentation:**

     ```
     docker run -it --rm --gpus device=2 --name nnunet \
       -v "/absolute/path/to/dataset":"/input" \
       -v "/absolute/path/to/output":"/output" \
       rixez/brats21nnunet
     ```

Regions of interest

Brain tumour and edema masks

- Example MRI slice **showing tumour segmentation** and edema, which can be used for **radiomics feature extraction**



Radiomic feature extraction

Using Python and PyRadiomics

- ❑ Measure quantitative properties of brain tissue from **preprocessed images and ROIs**
- ❑ **Features capture** intensity, texture, shape, and spatial patterns
- ❑ **PyRadiomics** is an open-source Python library **for standardised and reproducible** radiomic feature extraction
- ❑ Accurate preprocessing and ROI delineation are critical for **reproducibility and interpretability**

Installing PyRadiomics

macOS / Apple Silicon

MACQUARIE University

- ❑ **Visit:** https://pyradiomics.readthedocs.io/en/latest/installation.html
- ❑ **Clone the repository:**

```
git clone https://github.com/Radiomics/pyradiomics.git
```

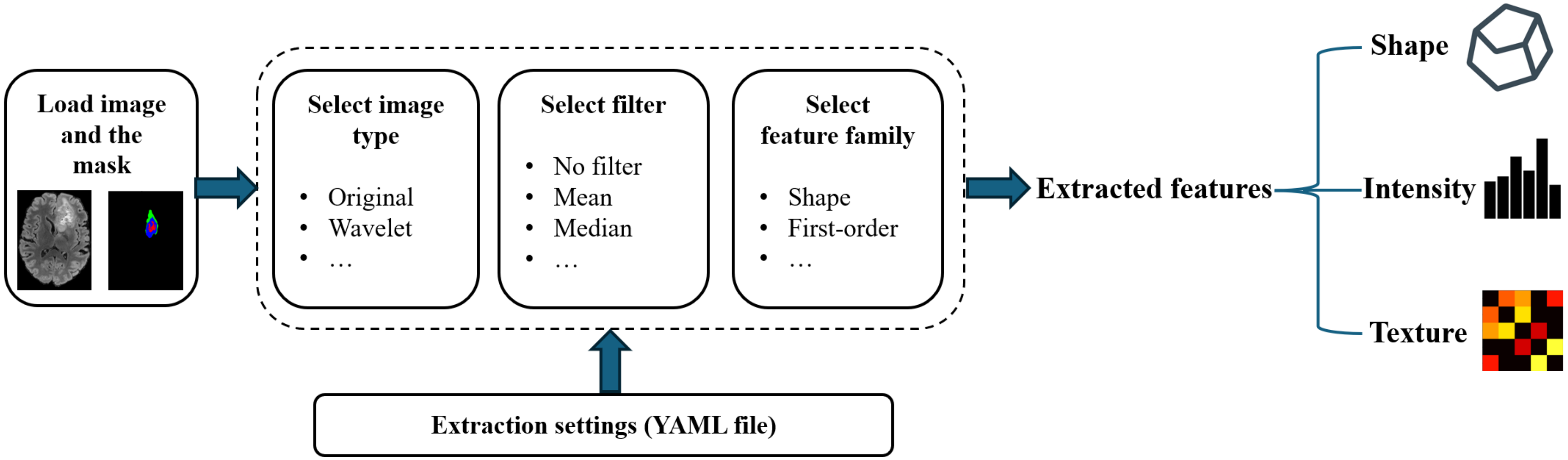
- ❑ **Install using pip:**

```
python -m pip install .
```

Radiomics extraction workflow

PyRadiomics configuration

- ❑ Illustrates **selecting image type**, **filters**, and **feature families**, with settings saved in a **YAML** file for standardised and reproducible feature extraction



Extracting radiomic features

Minimal Python example

- ❑ Example Python script to **extract radiomic features from an MRI using a given mask** with default settings (no filters, no image types).

```python
from radiomics import featureextractor
# Paths to the image and mask
image_path = "sub-01_pre_T1_BFC_brain.nii.gz"
mask_path = "sub-01_pre_T1_BFC_brain_mask.nii.gz"
# Initialise extractor with default settings
extractor = featureextractor.RadiomicsFeatureExtractor()
# Extract features
features = extractor.execute(image_path, mask_path)
```

Radiomic feature groups

Types and approximate counts

- **Shape features (~14):** geometry of ROI (volume, surface, compactness, sphericity)

- **First-order features (~18):** intensity distribution (mean, median, variance, skewness, kurtosis, entropy)

- **Texture features (~68–100):** spatial patterns of voxel intensities
  - **GLCM:** Co-occurrence of intensity pairs (contrast, correlation, homogeneity)
  - **GLRLM:** Run lengths of same intensity (granularity)
  - **GLSZM:** Size of connected regions with same intensity (uniformity/heterogeneity)
  - **GLDM:** Dependence between neighbouring voxels (complexity, smoothness)

Fractal-based features for brain imaging

Fractal-based features

Seeing complexity in the brain

- ❑ Fractal analysis quantifies **how complex and irregular** brain structures or signals are

- ❑ Many biological systems follow **repeating patterns across scales**, from cells to networks

- ❑ These patterns are often missed by standard measurements but are clinically meaningful

Intuition first

A simple visual analogy

- ❑ Like **coastlines or tree branches**, the brain shows similar patterns at different scales
- ❑ Zooming in reveals new details, but the overall structure remains familiar
- ❑ Fractal measures capture this **scale-independent organisation** in brain data

Fractal metrics

Key fractal features

- **Fractal Dimension (FD):** measures spatial complexity and self-similarity of structures or signals
- **Lacunarity:** quantifies heterogeneity and the distribution of gaps or variability within a structure
- Together, FD and lacunarity provide complementary information about **organisation, variability, and complexity**, enriching standard radiomic features

Fractals across modalities

Not just MRI

- ❑ **Structural MRI:** complexity of anatomy, lesions, or tissue organisation
- ❑ **fMRI:** temporal complexity of brain activity and functional dynamics
- ❑ **EEG / MEG:** irregularity and self-similarity in neural signals over time
- ❑ Enables a **common quantitative language** across imaging and signal modalities

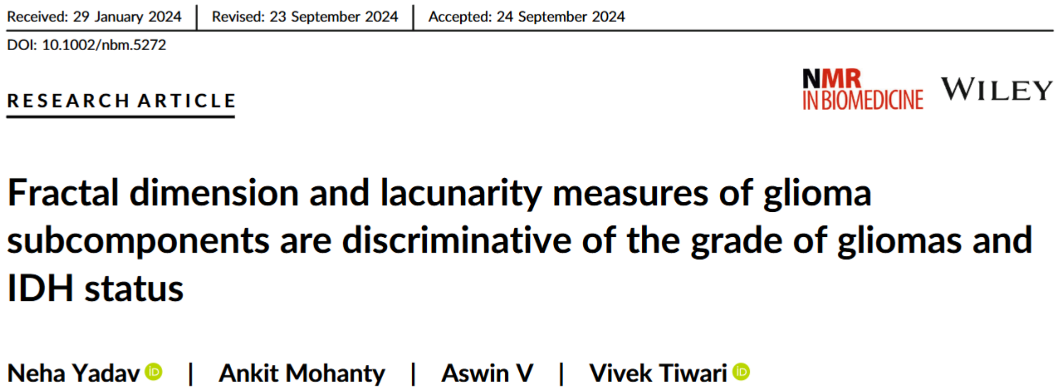
Fractal feature extraction workflow

Measuring tumour complexity from MRI: an example

- ❑ MRI scans are used to describe how complex and heterogeneous different tumour regions are

- ❑ Fractal measures help distinguish tumour subtypes and clinically relevant characteristics, such as grade and molecular status

- ❑ See details in:



Received: 29 January 2024 | Revised: 23 September 2024 | Accepted: 24 September 2024

DOI: 10.1002/nbm.5272

RESEARCH ARTICLE

NMR IN BIOMEDICINE WILEY

Fractal dimension and lacunarity measures of glioma subcomponents are discriminative of the grade of gliomas and IDH status

Neha Yadav | Ankit Mohanty | Aswin V | Vivek Tiwari

Predictive modelling and clinical interpretation

Radiomics + fractals

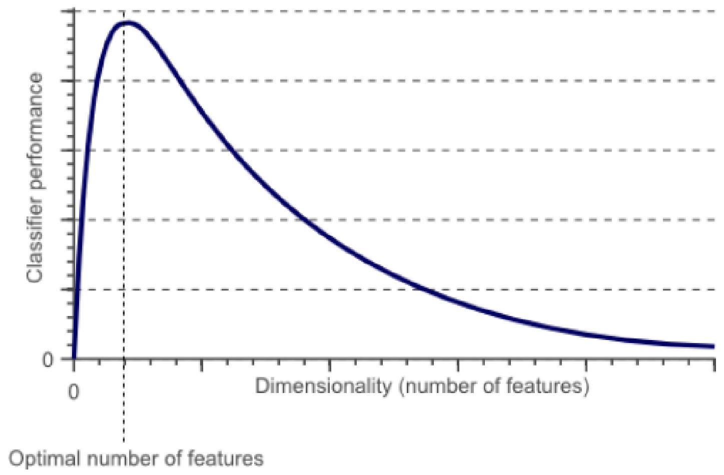
Complementary information

- **Radiomics:** what the tissue looks like (intensity, texture, shape)
- **Fractal features:** how complex and heterogeneous the structure or signal is
- **Combined features** provide a richer description for classification and prediction
- Supports more informative models for diagnosis, prognosis, and monitoring

Managing high-dimensional feature vectors

Feature reduction needed

- ❑ Combining radiomic features with fractal-based measures quickly leads to **hundreds of extracted features**
- ❑ **More features** require exponentially **more data** for accurate model generalisation
- ❑ Large feature vectors reduce Classifier Performance



Feature selection & dimensionality reduction

Techniques for reducing features

- With **limited patient samples**, we need to reduce features for robust, interpretable models
- **Feature selection:** retain the most predictive features without altering their representation
  - Examples: **SelectKBest, Boruta, Relief-F**
- **Dimensionality reduction:** transform features into **lower-dimensional space** while preserving variance
  - Example: **PCA**

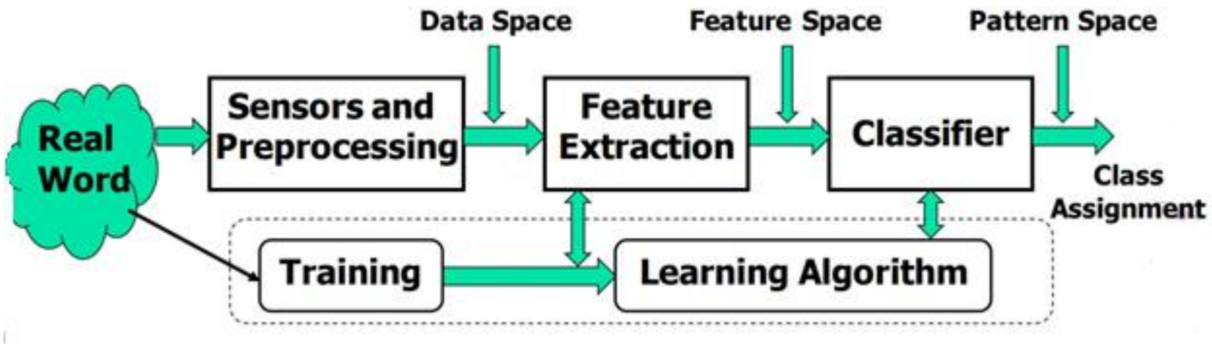
From features to clinical decisions

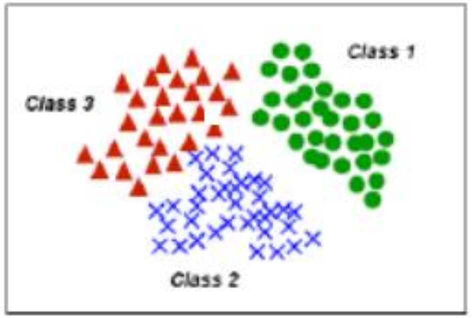
Classification and prediction

- ❑ Quantitative features extracted from images (radiomics and fractal-based) are used to **predict or classify clinical outcomes**
- ❑ These features can be combined to describe **tumour shape, texture, heterogeneity, and complexity**
- ❑ The goal is to support tasks such as **tumour grading, outcome prediction, or treatment response**
- ❑ **Visual analogy:** the model learns from past patients to help interpret new cases

Machine learning-based classifiers

Data → model → clinical output





Building predictive models

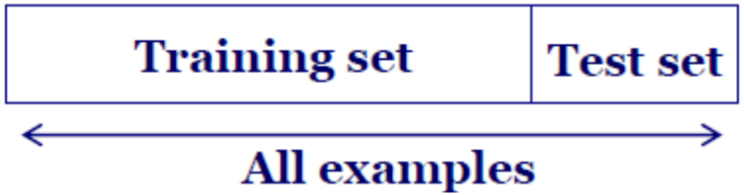
How classification is done

- ❑ Data are divided into **training and testing groups** to ensure fair evaluation
- ❑ The computer is shown **many past examples** with known outcomes (**training data**)
- ❑ It **learns patterns** that distinguish different clinical groups
- ❑ **New patients (test set)** are then assessed based on what the model has learned
- ❑ Performance is checked using **clinically meaningful measures** (e.g. accuracy, sensitivity)

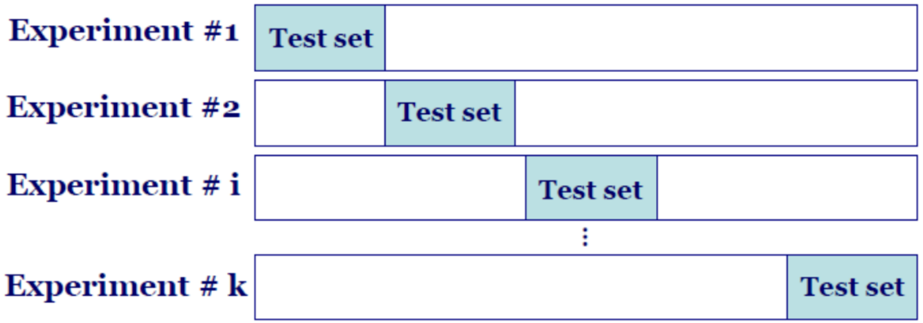
Performance evaluation

Data partitioning

- **Train/Test Split:** data is randomly partitioned **into two independent sets**



- **K-Fold Cross-Validation:** data is divided into K subsets; the model is trained on K-1 subsets and tested on the remaining one, repeated K times to assess stability



Predictive modelling in Python

Common libraries used

- ❑ **scikit-learn** – the most widely used library for building classification and prediction models (e.g. logistic regression, random forest, support vector machines)
- ❑ **NumPy & pandas** – handle numerical data and feature tables
- ❑ **SciPy** – supports statistical analysis and optimisation
- ❑ **pickle** – save trained models for reuse and reproducibility

Model explainability matters

Building clinical trust in AI predictions

- ❑ Clinicians need to understand **why** a model makes a prediction **before trusting** and using it in practice
- ❑ **Interpretability** is especially important **when AI outputs may influence diagnosis, prognosis, or treatment planning**
- ❑ In radiomics and fractal-based models, predictions are driven by hand-crafted features
- ❑ Feature attribution methods (e.g. **SHAP**) can highlight **which features contribute most to a prediction**

Limits of feature-based explanations

When “important features” are hard to interpret

- ❑ Many radiomic features lack direct clinical meaning
- ❑ E.g., a feature such as **GLCM entropy** may be **influential but difficult to interpret** clinically
- ❑ Techniques like **SHAP**, help identify what the model is using, but **not always what it means clinically**
- ❑ **Feature importance alone** may therefore be **insufficient** to support clinical confidence

Making predictions more clinically useful

Beyond binary outputs

- ❑ **Report predicted probabilities** (e.g. 85% likelihood) rather than only binary labels (yes/no)
- ❑ **Probabilities better reflect uncertainty** and align with how clinicians reason about risk
- ❑ Show the predicted probability alongside the distribution of probabilities from correctly classified training cases
- ❑ This provides context: “**How confident is this prediction compared to similar patients?**”

Towards clinically meaningful explanations

Linking features to clinical knowledge

- After identifying the most influential features, an **additional interpretation layer** can be applied
- Large Language Models can be used to **review the literature and summarise known clinical associations**
- The output can be a short, **clinician-readable report linking feature behaviour to** known biological or **clinical findings**

Key take aways and references

Take aways

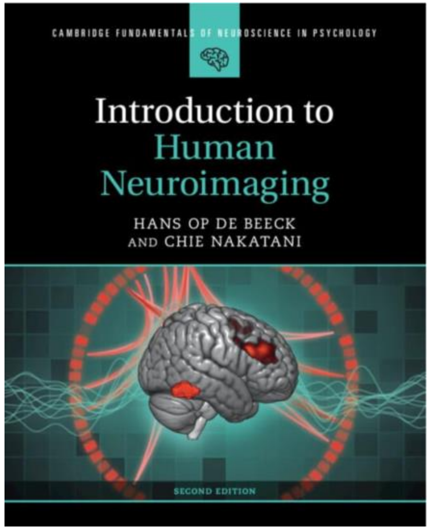
Linking features to clinical knowledge

- ❑ Computational pipelines turn medical images into quantitative, reproducible data
- ❑ Careful preprocessing and ROI definition are critical for reliable results
- ❑ Radiomics and fractal-based features capture complementary tissue information
- ❑ Predictive models can support clinical decisions, not replace them
- ❑ Explainability is essential to build trust and enable clinical adoption
- ❑ Combining feature attribution, uncertainty estimates, and clinical context improves trust and usability
- ❑ **The goal is not to replace clinicians, but to support informed decision-making**

Further reading

Recommended (text)books

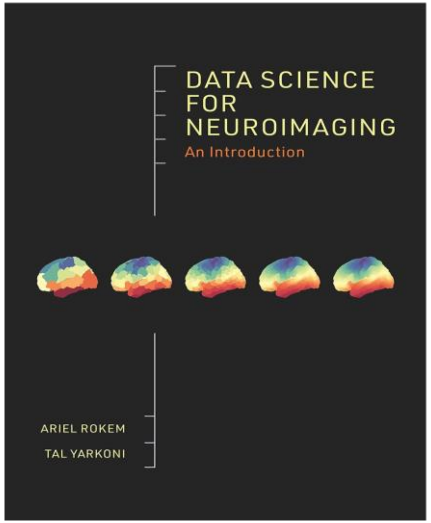
- ❑ **Foundational and accessible references** covering human neuroimaging, data science methods for neuroimaging, and fractal analysis of the brain


CAMBRIDGE FUNDAMENTALS OF NEUROSCIENCE IN PSYCHOLOGY

Introduction to Human Neuroimaging

HANS OP DE BEECK AND CHIE NAKATANI

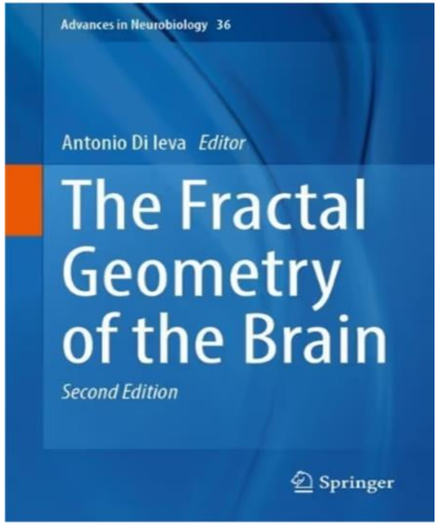
SECOND EDITION


DATA SCIENCE FOR NEUROIMAGING

An Introduction

ARIEL ROKEM

TAL YARKONI

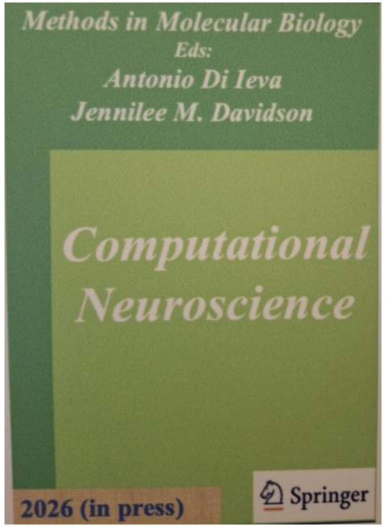

Advances in Neurobiology 36

Antonio Di Ieva Editor

The Fractal Geometry of the Brain

Second Edition

Springer


Methods in Molecular Biology

Eds:

Antonio Di Ieva

Jennilee M. Davidson

Computational Neuroscience

2026 (in press)

Springer

MACQUARIE University

Thank you!