Module Interface Specification for BrainInsight3D: 3D fMRI Visualization & Segmentation

Nada Elmasry

March 20, 2024

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/NadaElmasry/3D-Image-Reconstruction-Segmentation/blob/main/docs/SRS/SRS.pdf

Contents

3 Introduction

The following document details the Module Interface Specifications for BrainInsight3D: 3D fMRI Visualization & Segmentation.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/NadaElmasry/3D-Image-Reconstruction-Segmentation.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by ProgName.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of ProgName uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, ProgName uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding Module	
Behaviour-Hiding Module	Input Format Module Input Verification Module Data Preprocessing Module 2D Slice Module 3D Volume Module Segmentation Module User Interface Module Main Program Module
Software Decision Module	2D Data Visualization Module 3D Data Visualization Module User Interface Rendering Module Segmentation Algorithm Processing Module Optimization Module

Table 1: Module Hierarchy

6 MIS of Input Format Module

6.1 Module

InputFormat

6.2 Uses

None

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

1. read_scan

    - **Input:** NIfTI file containing scan meta-data and scan of dimensions $1 \times 3 \times 3 \times 3$ containg the input scan voxel values at each timepoint.
    - **Output:** scan meta-data in a tuple format and scan voxel values in NumPy ndarray of dimensions $1 \times 3 \times 3 \times 3$.
    - **Exception:** "InvalidFormatError" if the input file is not in NIfTI format. "FileNotFoundError" if the input file cannot be found in file path. "CorruptFileError" if the input scan is corrupted.

6.4 Semantics

6.4.1 State Variables

CurrentScanData: Carries the input scan-metadata and matrix with voxel values.

6.4.2 Environment Variables

None

6.4.3 Assumptions

None

6.4.4 Access Routine Semantics

read_scan(scan_file):

- transition: accepts NIfTI files as input and changes the state variable CurrentScanData to the most recent successfully read NIfTI scan file.
- output: scan meta-data (tuple), scan voxel values (NumPy ndarray of dimensions $1x3x3x3$)
- exception: "InvalidFormatError" if the input file is not in NIfTI format. "FileNotFoundError" if the input file cannot be found in file path. "CorruptFileError" if the input scan is corrupted.

6.4.5 Local Functions

None

7 MIS of Input Verification Module

7.1 Module

Input Verification

7.2 Uses

ImageFormatModule

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

1. validate_scan

    - **Input:** scan meta-data in a tuple format and scan voxel values in NumPy ndarray of dimensions $1 \times 3 \times 3 \times 3$.
    - **Output:** None.
    - **Exception:** "MissingMetaDataError" if metadata is missing attributes required by the software for correct rendering. "MismatchedDimensionsError" if scan dimensions are not as expected by the software application. "InvalidDataRange" if input scan data isn't in the normal fMRI scan ranges or if all values is zero.

7.4 Semantics

7.4.1 State Variables

None

7.4.2 Environment Variables

None

7.4.3 Assumptions

Scan data has successfully passed the checks in ImageFormatModule.

7.4.4 Access Routine Semantics

validate_scan(meta_data, scan_array):

- transition: -
- output: None.
- exception: "MissingMetaDataError" if metadata is missing attributes required by the software for correct rendering. "MismatchedDimensionsError" if scan dimensions are not as expected by the software application. "InvalidDataRange" if input scan data isn't in the normal fMRI scan ranges or if all values is zero.

7.4.5 Local Functions

None.

8 MIS of Data Preprocessing Module

8.1 Module

DataProc

8.2 Uses

None

8.3 Syntax

8.3.1 Exported Constants

None

8.3.2 Exported Access Programs

1. align_scan
    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of scan data aligned.
    - **Exception:** "ImageAlignmentError" if there is an error that causes the alignment algorithm to fail.
2. normalize_scan
    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of scan voxel data normalized with zero mean and unit standard deviation.
    - **Exception:** "ImageNormalizationError" if there is an error that causes the normalization algorithm to fail.
3. remove_noise
    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of scan voxel data with reduced noise.
    - **Exception:** "ImageNoiseRemovalError" if there is an error that causes the noise removal algorithm to fail.
4. register_scan
    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of registered scan voxel data.

- **Exception:** "ImageRegistrationError" if there is an error that causes the registration algorithm to fail.

5. enhance_contrast

   - **Input:** NumPy ndarray of scan voxel data.
   - **Output:** NumPy ndarray of scan voxel data with enhanced contrast.
   - **Exception:** "ImageContrastEnhancementError" if there is an error that causes the contrast enhancement algorithm to fail.

8.4 Semantics

8.4.1 State Variables

- isAligned: boolean to indicate whether the scan has passed through the image alignment step of the preproceesing pipeline.
- isRegistered: boolean to indicate whether the scan has passed through the image registeration step of the preproceesing pipeline.
- isNormalized: boolean to indicate whether the scan has passed through the image normalization step of the preproceesing pipeline.
- isDenoised: boolean to indicate whether the scan has passed through the image denoising step of the preproceesing pipeline.
- isContrastEnhanced: boolean to indicate whether the scan has passed through the contrast enhancement step of the preproceesing pipeline.

8.4.2 Environment Variables

None.

8.4.3 Assumptions

Scan data has successfully passed the tests in the InputVerificationModule.

8.4.4 Access Routine Semantics

align_scan(scan_array):

- transition: performs image alignment process on all slices of the input scan volumes and sets the isAligned state variable to true if the function terminates successfully.
- output: NumPy ndarray of scan data aligned.

- exception: "ImageAlignmentError" if there is an error that causes the alignment algorithm to fail.

register_scan(scan_array):

- transition: performs image registration process on all slices of the input scan volumes and sets the isRegistered state variable to true if the function terminates successfully.
- output: NumPy ndarray of registered scan data.
- exception: "ImageRegistrationError" if there is an error that causes the registration algorithm to fail.

normalized_scan(scan_array):

- transition: performs image normalization process on all slices of the input scan volumes and sets the isNormalized state variable to true if the function terminates successfully.
- output: NumPy ndarray of normalized scan data.
- exception: "ImageNormalizationError" if there is an error that causes the normalization algorithm to fail.

remove_noise(scan_array):

- transition: performs noise removal process on all slices of the input scan volumes and sets the isDenoised state variable to true if the function terminates successfully.
- output:NumPy ndarray of scan voxel data with reduced noise.
- exception: "ImageNoiseRemovalError" if there is an error that causes the noise removal algorithm to fail.

enhance_contrast(scan_array):

- transition: performs contrast enhancement process on all slices of the input scan volumes and sets the isContrastEnhanced state variable to true if the function terminates successfully.
- output: NumPy ndarray of scan data with enhanced contrast.
- exception: "ImageContrastEnhancementError" if there is an error that causes the contrast enhancement algorithm to fail.

8.4.5 Local Functions

9 MIS of 2D Slice Module

9.1 Module

2DSlicer

9.2 Uses

DataPreprocessingModule

9.3 Syntax

9.3.1 Exported Constants

9.3.2 Exported Access Programs

1. get_slice_data

    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of slice data where each slice is a two-dimensional NumPy ndarray.
    - **Exception:** "SlicingError" if the slicing algorithm cannot perform slicing properly due to input incorrect dimensions or an error or condition causes the slicing algorithm to terminate without completing the slicing process.

9.4 Semantics

9.4.1 State Variables

None

9.4.2 Environment Variables

None

9.4.3 Assumptions

Input scan is preprocessed and is in correct format.

9.4.4 Access Routine Semantics

get_slice_data(scan_array):

- transition: -

- output: NumPy ndarray of slice data where each slice is a two-dimensional NumPy ndarray.
- exception: "SlicingError" if the slicing algorithm cannot perform slicing properly due to input incorrect dimensions or an error or condition causes the slicing algorithm to terminate without completing the slicing process.

9.4.5 Local Functions

None.

10 MIS of 3D Volume Module

10.1 Module

3DVolumeReconstruct

10.2 Uses

Data Preprocessing Module

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

1. get_volume_data

    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of scan voxel data after compression and mapping.
    - **Exception:** "MappingError" if the voxel mapping algorithm fails due to scan properties. "CompressionError" if the input scan cannot be compressed to the desired size without significant loss of data. "ReconstructionError" if the 3D volume reconstruction fails to terminate successfully and return non-corrupt volume data.

10.4 Semantics

10.4.1 State Variables

None

10.4.2 Environment Variables

None

10.4.3 Assumptions

Input scan is preprocessed and is in correct format.

10.4.4 Access Routine Semantics

get_volume_data(scan_array):

- transition: -

- output: NumPy ndarray of scan voxel data after compression and mapping.
- exception: "MappingError" if the voxel mapping algorithm fails due to scan properties. "CompressionError" if the input scan cannot be compressed to the desired size without significant loss of data. "ReconstructionError" if the 3D volume reconstruction fails to terminate successfully and return non-corrupt volume data.

10.4.5 Local Functions

None.

11 MIS of Segmentation Module

11.1 Module

Seg

11.2 Uses

Data Preprocessing Module

11.3 Syntax

11.3.1 Exported Constants

11.3.2 Exported Access Programs

1. segment_volume

    - **Input:** NumPy ndarray of scan voxel data.
    - **Output:** NumPy ndarray of scan voxel data and the associated segmentation mask value for each voxel.
    - **Exception:** "SegmentationError" if model fails to produce segmentation mask from input scan due to model internal error or network connectivity error if the connection with the web application has been reset.

11.4 Semantics

11.4.1 State Variables

- InputScan:
- SegmentationMask:

11.4.2 Environment Variables

None.

11.4.3 Assumptions

Input data has passed through the data preprocessing pipeline.

11.4.4 Access Routine Semantics

get_volume_data(scan_array):

- transition: -
- output: NumPy ndarray of scan voxel data and the associated segmentation mask value for each voxel.
- exception: "SegmentationError" if model fails to produce segmentation mask from input scan due to model internal error or network connectivity error if the connection with the web application has been reset.

11.4.5 Local Functions

None.

12 MIS of User Interface Module

[Use labels for cross-referencing —SS]

12.1 Module

UI

12.2 Uses

Main Program Module

12.3 Syntax

12.3.1 Exported Constants

12.3.2 Exported Access Programs

12.4 Semantics

12.4.1 State Variables

DisplayedTab: Indicates which tab (3D volume visualization, 2D slices visualization) is currently displayed.

12.4.2 Environment Variables

None.

12.4.3 Assumptions

None.

12.4.4 Access Routine Semantics

12.4.5 Local Functions

None.

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.