DEEPHEALTH

DICOM and NIfTI formats

Costantino Grana - UNIMORE

Winter School 25/01/2022

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Contents

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Medical Imaging File Formats

- The file format describes how the image data are organized inside the image file and how the pixel data should be interpreted by a software for the correct loading and visualization.
- Medical image file formats can be divided in two main categories according to their goal:
  - Standardize the images generated by diagnostic modalities, e.g., DICOM
  - Facilitate and strengthen post-processing analysis, e.g., NIfTI

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM

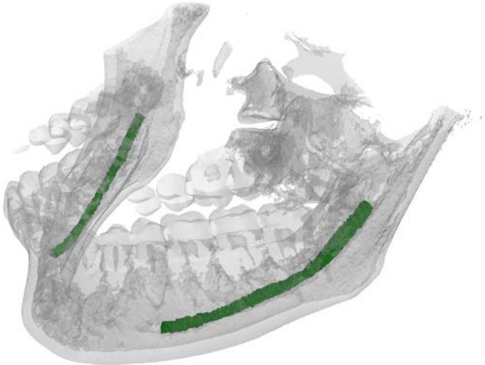
- Digital Imaging and Communications in Medicine (DICOM)
- It has been widely adopted by hospitals and incorporates standards for imaging modalities such as radiography, ultrasonography, computed tomography (CT), magnetic resonance imaging (MRI), and radiation therapy.
- Allows to store both data and metadata: an image that is separate from its metadata becomes "meaningless" as medical image
- The DICOM header, in addition to the information about the image matrix, contains the most complete description of the entire procedure used for the acquisition
- A DICOM file also contains patient information

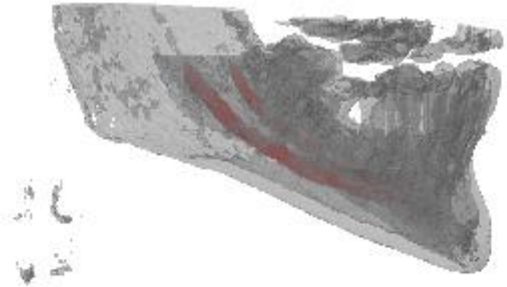
The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM

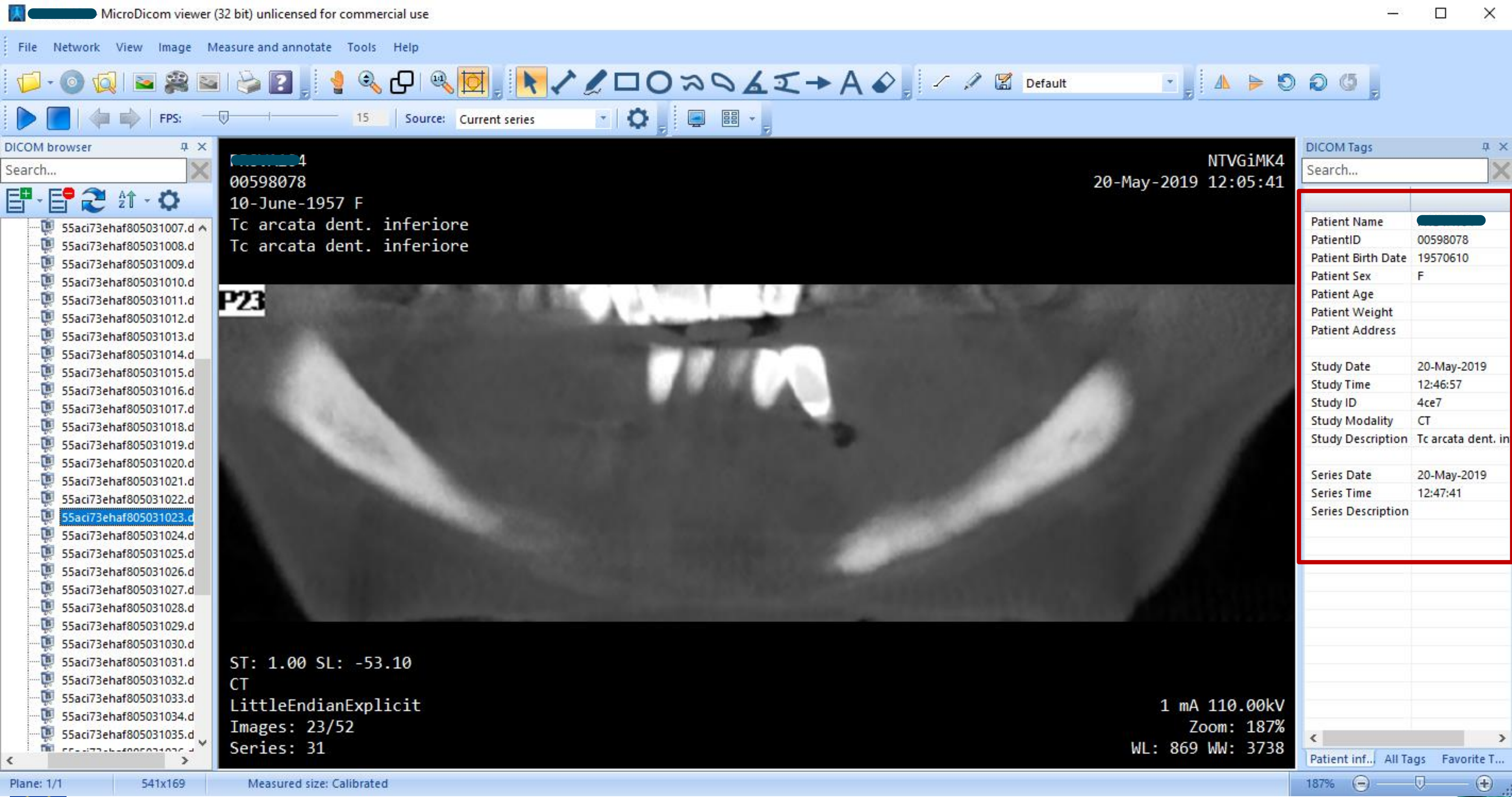
- DICOM includes protocols for image exchange, image compression, 3-D visualization, image presentation, and results reporting.
- There exists plenty of software to open and view with DICOM files. Micro DICOM is on of them (its free for non-commercial usage):

https://www.microdicom.com/downloads.html





The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.



```
55aci73ehaf805031007.d
55aci73ehaf805031008.d
55aci73ehaf805031009.d
55aci73ehaf805031010.d
55aci73ehaf805031011.d
55aci73ehaf805031012.d
55aci73ehaf805031013.d
55aci73ehaf805031014.d
55aci73ehaf805031015.d
55aci73ehaf805031016.d
55aci73ehaf805031017.d
55aci73ehaf805031018.d
55aci73ehaf805031019.d
55aci73ehaf805031020.d
55aci73ehaf805031021.d
55aci73ehaf805031022.d
55aci73ehaf805031023.d
55aci73ehaf805031024.d
55aci73ehaf805031025.d
55aci73ehaf805031026.d
55aci73ehaf805031027.d
55aci73ehaf805031028.d
55aci73ehaf805031029.d
55aci73ehaf805031030.d
55aci73ehaf805031031.d
55aci73ehaf805031032.d
55aci73ehaf805031033.d
55aci73ehaf805031034.d
55aci73ehaf805031035.d
```

```
[redacted]4                                              NTVGiMK4
00598078                                     20-May-2019 12:05:41
10-June-1957 F
Tc arcata dent. inferiore
Tc arcata dent. inferiore

ST: 1.00 SL: -53.10
CT
LittleEndianExplicit                              1 mA 110.00kV
Images: 23/52                                        Zoom: 187%
Series: 31                                      WL: 869 WW: 3738
```

Patient Name	[redacted]
PatientID	00598078
Patient Birth Date	19570610
Patient Sex	F
Patient Age	
Patient Weight	
Patient Address	
Study Date	20-May-2019
Study Time	12:46:57
Study ID	4ce7
Study Modality	CT
Study Description	Tc arcata dent. in
Series Date	20-May-2019
Series Time	12:47:41
Series Description	

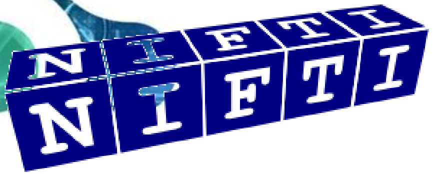
Plane: 1/1   541x169   Measured size: Calibrated

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM

- Especially when working with 3D volumes, being able to correctly visualize data requires to store additional information (e.g. pixel spacing).
- The ECVL Image allows to store all these information to properly handle and work with DICOMs.
- To enable DICOM support in ECVL the external dependencies DCMTK is required.
- DCMTK is a DICOM ToolKit consisting of source code, documentation and installation instructions for a set of software libraries and applications implementing part of the DICOM/MEDICOM Standard.

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.



NIfTI

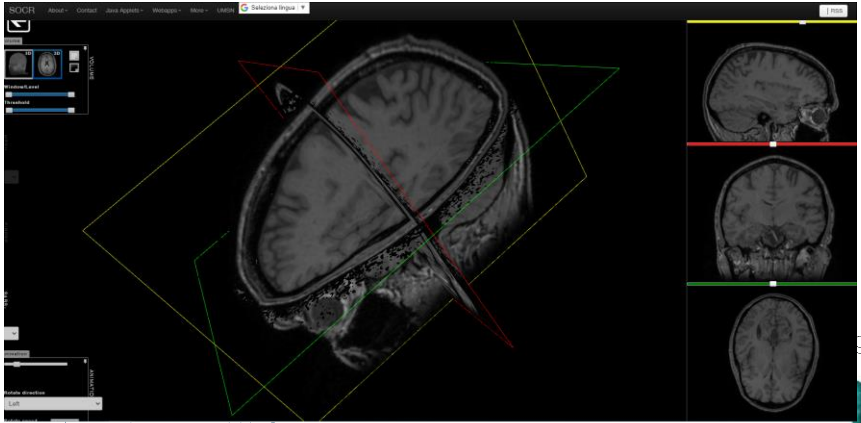
- Neuroimaging Informatics Technology Initiative (Nifti)
- Specifically designed for neuroimaging, it stores additional information w.r.t. DICOM (e.g. image orientation) with the intent to avoid the left–right ambiguity in brain study.
- The format allows the storage of the header and pixel data in separate files. Anyway, the images are typically saved as a single ".nii" file in which the header and the pixel data are merged.
- An update version of the standard, the NifTi-2, developed to manage larger data set has been defined in the 2011.
- NifTi-2 encode each of the dimensions of an image matrix with a 64-bit integer instead of using a 16-bit (as in the Nifti-1).

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

View NIfTI Files

- To view NiFTI files, you can use different programs:
  - ImageJ, a java-based program that runs on most operating systems, including MAC OSX
  - MBAT, from the Laboratory of Neural Imaging at USC
  - Mango, from Neuroimaging Informatics Tools and Resources Clearinghouse, offers desktop, web, and iPad-compatible versions
  - MRIcro, which runs on Windows and Linux systems
- Or web services:
  - Brain Viewer Webapp - SOCR

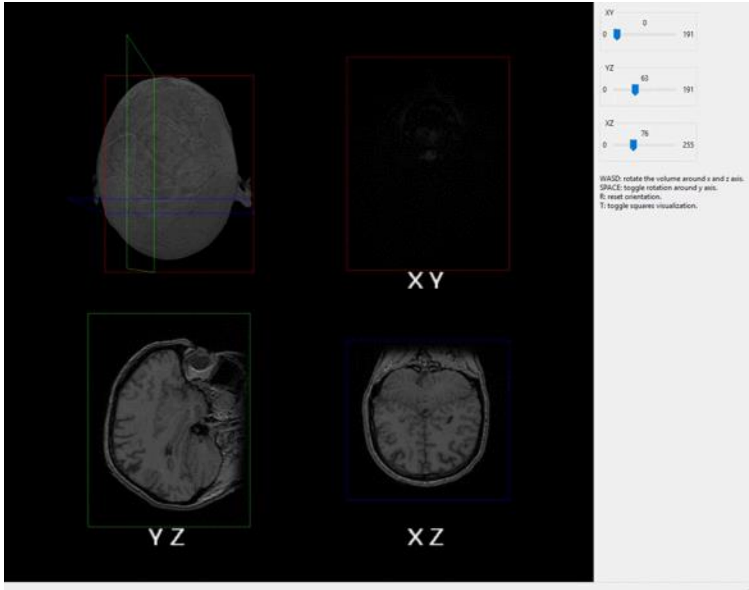
  https://socr.umich.edu/HTML5/BrainViewer/



The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

View NIfTI Files

- In the ecvl-applications GitHub repository you can find a C++ software based on ECVL and wxWidgets (a Cross-Platform GUI Library) and OpenGL (a cross-language, cross-platform API for rendering 2D and 3D vector graphics).
- This application also contains a volume rendering tool for NIfTI images.



WASD: rotate the volume around x and z axis.
SPACE: toggle rotation around y axis.
R: reset orientation.
T: toggle squares visualization.

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

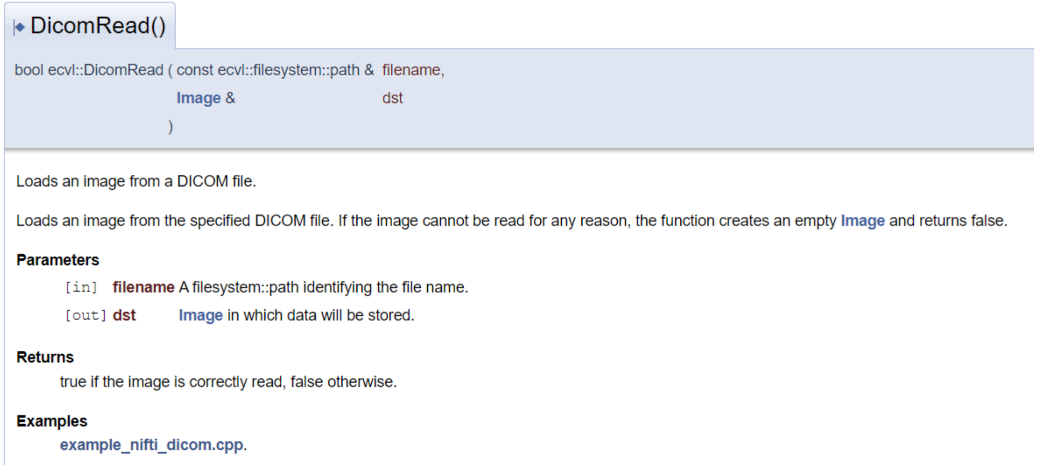
DICOM vs NIfTI

- The DICOM image (or file) is always a 2D image. 3D volumes are stored in multiple images each containing one slice of a MRI, CT, TEP, etc
- The NifTi file can contain the full set of images.
- A NifTi volume is a whole representation of a part of the human body, that contains all the slices took using the X-rays or any other method.
- NIfTi was designed to be simpler than DICOM and to ensure more efficiency for image processing analysis.
- Several other Medical Image File formats exists (e.g. Analyze, Minc), but they are less common than DICOM and NifTI

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM and NIfTI with ECVL

- ECVL library exports four different APIs to read and write DICOM and NifTI files
- Specifically, DicomRead and DicomWrite are respectively responsible for reading and writing DICOM file from and to disc.



◆ DicomRead()

```
bool ecvl::DicomRead ( const ecvl::filesystem::path &  filename,
                       Image &                          dst
                     )
```

Loads an image from a DICOM file.

Loads an image from the specified DICOM file. If the image cannot be read for any reason, the function creates an empty Image and returns false.

Parameters

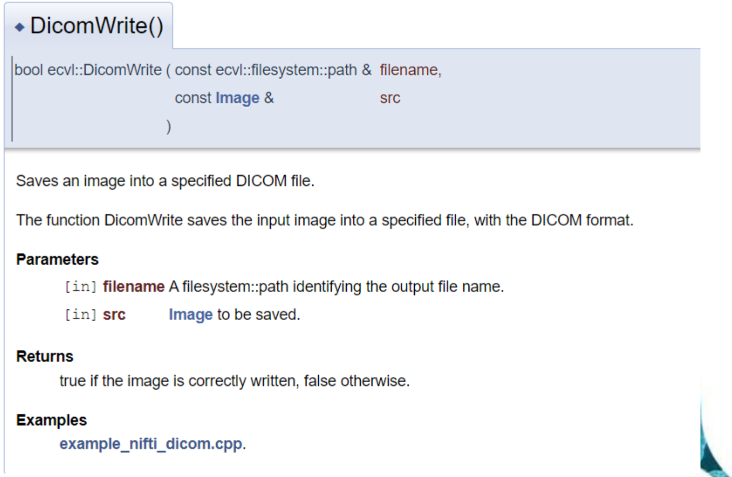
[in] **filename** A filesystem::path identifying the file name.
[out] **dst** Image in which data will be stored.

Returns

true if the image is correctly read, false otherwise.

Examples

example_nifti_dicom.cpp.



◆ DicomWrite()

```
bool ecvl::DicomWrite ( const ecvl::filesystem::path &  filename,
                        const Image &                    src
                      )
```

Saves an image into a specified DICOM file.

The function DicomWrite saves the input image into a specified file, with the DICOM format.

Parameters

[in] **filename** A filesystem::path identifying the output file name.
[in] **src** Image to be saved.

Returns

true if the image is correctly written, false otherwise.

Examples

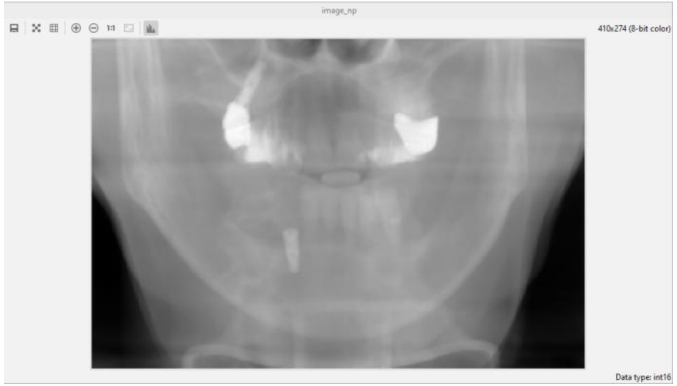
example_nifti_dicom.cpp.

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM and NIfTI with ECVL

- You can read a DICOM image simply using the generic ecvl.ImRead function

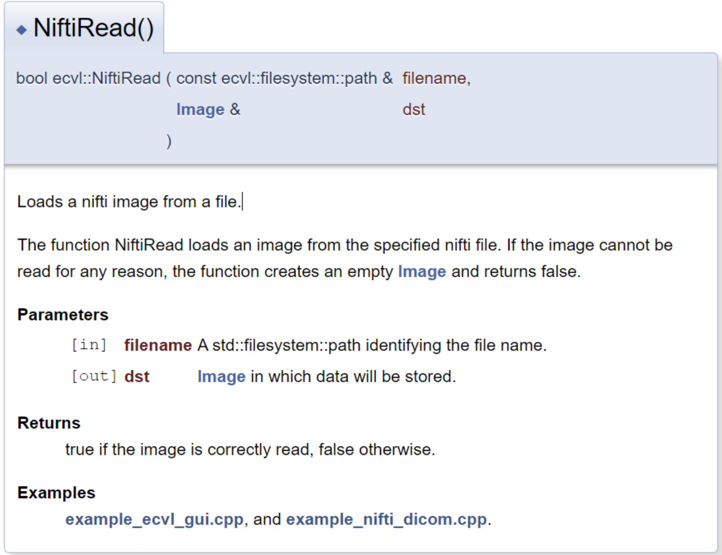
```
import numpy as np
import pyecvl.ecvl as ecvl

image = ecvl.DicomRead("data/DICOM/55aci73ehaf802001000.dcm")
image_np = np.asarray(image).transpose([1,0,2])

image = ecvl.ImRead("data/DICOM/55aci73ehaf802001000.dcm")
image_np = np.asarray(image).transpose([1,0,2])

ecvl.DicomWrite("data/out_55aci73ehaf802001000.dcm", image)
```



The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM and NIfTI with ECVL

- NiftyRead and NiftyWrite are respectively responsible for reading and writing NifTI file from and to disc.



◆ NiftiRead()

```
bool ecvl::NiftiRead ( const ecvl::filesystem::path &  filename,
                       Image &                          dst
                     )
```

Loads a nifti image from a file.

The function NiftiRead loads an image from the specified nifti file. If the image cannot be read for any reason, the function creates an empty Image and returns false.

Parameters

[in] **filename** A std::filesystem::path identifying the file name.

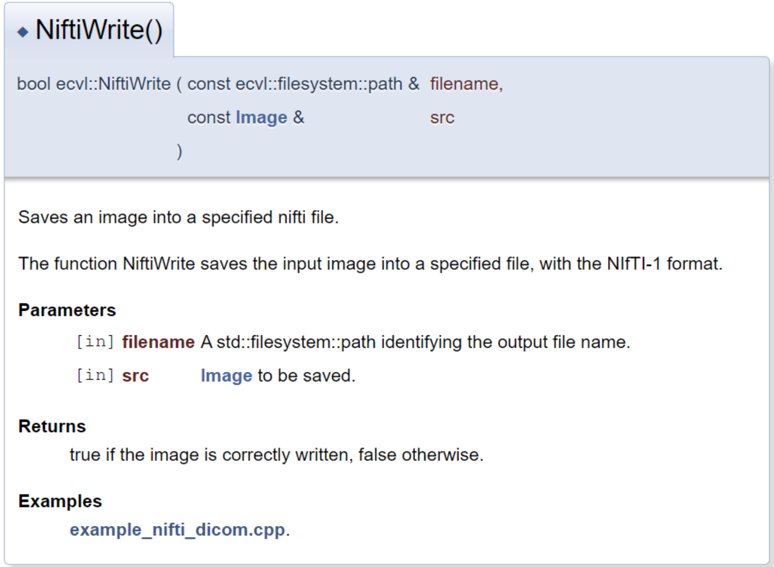
[out] **dst** Image in which data will be stored.

Returns

true if the image is correctly read, false otherwise.

Examples

example_ecvl_gui.cpp, and **example_nifti_dicom.cpp**.



◆ NiftiWrite()

```
bool ecvl::NiftiWrite ( const ecvl::filesystem::path &  filename,
                        const Image &                    src
                      )
```

Saves an image into a specified nifti file.

The function NiftiWrite saves the input image into a specified file, with the NIfTI-1 format.

Parameters

[in] **filename** A std::filesystem::path identifying the output file name.

[in] **src** Image to be saved.

Returns

true if the image is correctly written, false otherwise.

Examples

example_nifti_dicom.cpp.

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DICOM and NIfTI with ECVL

- You can read a NifTI volume simply using the generic ecvl.ImRead function

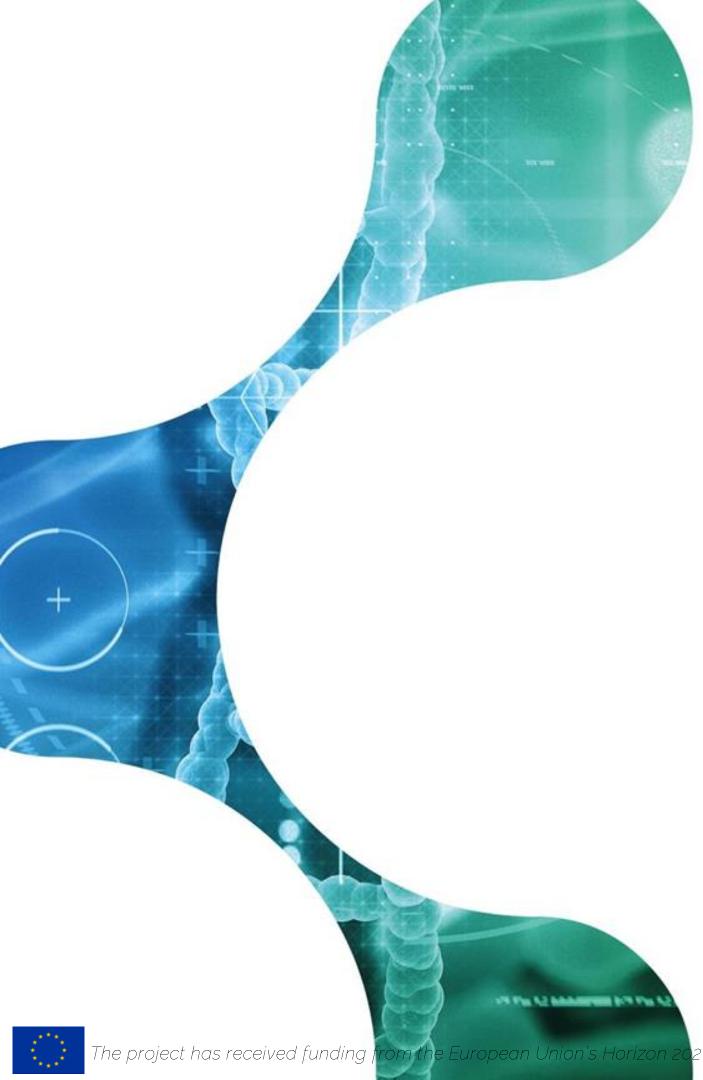
```
import numpy as np
import pyecvl.ecvl as ecvl

image = ecvl.NiftiRead("data/brain_mr_01.nii")
image_np = np.asarray(image).transpose([1,0,2])

image = ecvl.ImRead("data/brain_mr_01.nii")
image_np = np.asarray(image).transpose([1,0,2])

ecvl.NiftiWrite("data/out_brain_mr_01.nii", image)
```

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Metadata

- When reading DICOM and NifTI files, ECVL imports also metadata
- Through the .getMeta() and .setMeta() function methods you can get or set metadata of an Image.
- This functionality is fully implemented in C++, but still missing in the python API.
- Do you want to contribute? You can do it with a pull-request.

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.



DEEPHEALTH

Thank you!

Costantino Grana
costantino.grana@unimore.it

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.