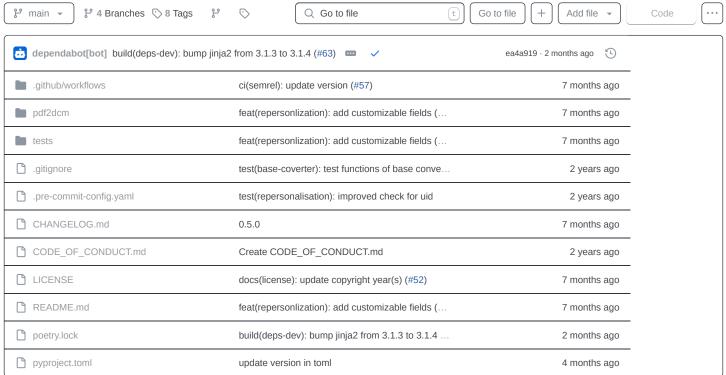


Public repository





To the check the setup, simply check the version number of the pdf2dcm package by

```
python -c 'import pdf2dcm; print(pdf2dcm.__version__)'
```

Poppler Setup

Poppler is a popular project that is used for the creation of Dicom RGB Secondary Capture. You can check if you already have it installed by calling pdftoppm -h in your terminal/cmd. To install poppler these are some of the recommended ways-

Q

Conda

```
conda install -c conda-forge poppler
```

Ubuntu

```
sudo apt-get install poppler-utils
```

MacOS

```
brew install poppler
```

PDF to Encapsulated DCM

Usage

```
from pdf2dcm import Pdf2EncapsDCM

converter = Pdf2EncapsDCM()
converted_dcm = converter.run(path_pdf='tests/test_data/test_file.pdf', path_template_dcm='tests/test_data/CT_small.c
print(converted_dcm)
# [ 'tests/test_data/test_file.dcm' ]
```

Parameters converter.run:

- path_pdf (str) : path of the pdf that needs to be encapsulated
- path_template_dcm (str, optional): path to template for getting the repersonalisation of data.
- suffix (str, optional): suffix of the dicom files. Defaults to ".dcm".

Returns:

• List[Path] : list of path of the stored encapsulated dcm

PDF to RGB Secondary Capture DCM

Usage

```
from pdf2dcm import Pdf2RgbSC

converter = Pdf2RgbSC()
converted_dcm = converter.run(path_pdf='tests/test_data/test_file.pdf', path_template_dcm='tests/test_data/CT_small.c
print(converted_dcm)
# [ 'tests/test_data/test_file_0.dcm', 'tests/test_data/test_file_1.dcm' ]
```

Parameters converter.run:

• path_pdf (str): path of the pdf that needs to be converted

- path_template_dcm (str, optional): path to template for getting the repersonalisation of data.
- suffix (str, optional): suffix of the dicom files. Defaults to ".dcm".

Returns:

• List[Path]: list of paths of the stored secondary capture dcm

Notes

- The name of the output dicom is same as the name of the input pdf
- If no template is provided no repersonalisation takes place
- It is possible to produce dicoms without a suffix by simply passing suffix="" to the converter.run()

Repersonalisation

It is the process of copying over data regarding the identity of the encapsualted pdf from a template dicom. Currently, the fields that are repersonalised by default are-

- PatientName
- PatientID
- PatientSex
- StudyInstanceUID
- SeriesInstanceUID
- SOPInstanceUID

The fields SeriesInstanceUID and SOPInstanceUID have been removed from the repersonalization by copying as it violates the DICOM standards.

You can set the fields to repersonalize by passing repersonalisation_fields into Pdf2EncapsDCM(), or Pdf2Rgbsc()

Example:

```
fields = [
    "PatientName",
    "PatientID",
    "PatientSex",
    "StudyInstanceUID",
    "AccessionNumber"
]
converter = Pdf2RgbSC(repersonalisation_fields=fields)
```

☐ README

Code of conduct

MIT license



≔

(\$)

Releases 8

v0.5.0 Latest on Jan 31

+7 releases

Packages

No packages published Publish your first package



Deployments 8

github-pages 7 months ago

+ 7 deployments

Languages

Python 100.0%