
pediatricIQphantoms

Release 0.0.2

Brandon Nelson, Rongping Zeng

Jan 12, 2026

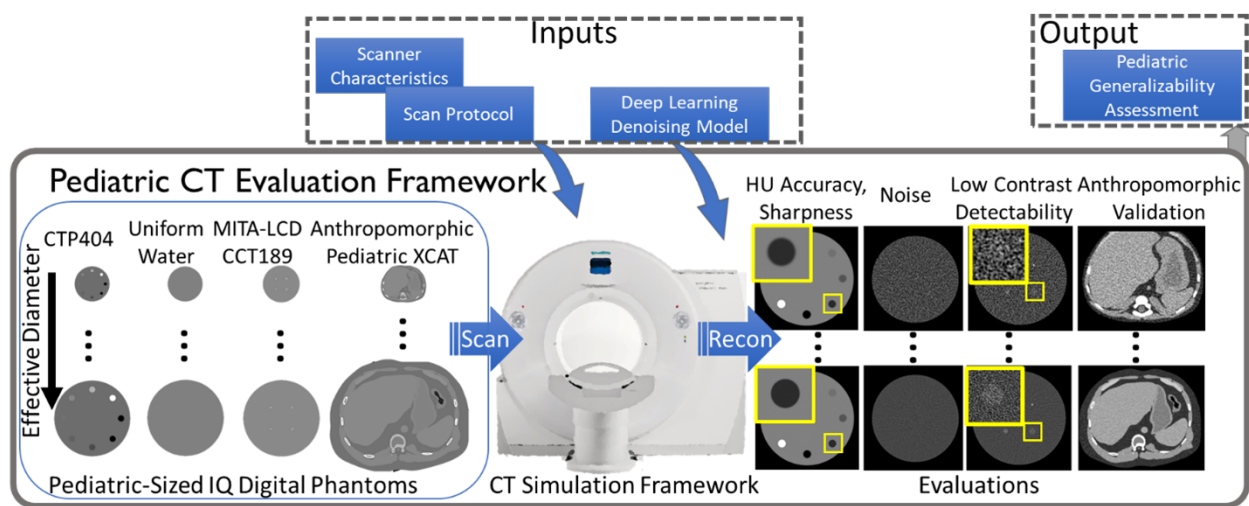
CONTENTS:

1	Introduction	3
2	Installation	5
3	Contents	7
3.1	Usage	7
3.2	Running Simulations	13
3.3	Evaluating Denoising Efficiency	34
3.4	Function Reference	46
3.5	Contributing Guide	48
3.6	Frequently Asked Questions	49
4	Developers	51
5	Indices and tables	53
	Python Module Index	55

This documentation provides information regarding how to download, install, and use the pediatricIQphantoms tools which are designed to simulate computed tomography (CT) image characteristics in a range of pediatric sizes ranging from newborn through adolescence.

INTRODUCTION

DOI 10.5281/zenodo.10064035  pediatricIQphantoms failing



Digital Pediatric Image Quality Phantoms for Evaluating CT Denoising Methods are a set of digital phantoms and simulation methods for generating CT images of standard image quality (IQ) phantoms designed to match the effective diameter of pediatric patients ranging from newborns to teenagers. This repository has [tools](#) for generating [MITA-LCD phantom](#) and a multi-contrast sensitometry module similar to the CTP404 module of the [Catphan 600 phantom](#). Functions are also provided to simulate different acquisition parameters and CT scanner models.

Size is one of the most important patient factors influencing CT performance as it determines the overall x-ray attenuation and noise properties. New deep learning-based denoisers have shown potential to improve image quality for a fixed radiation dose or maintain image quality while reducing dose [Brady 2023](#). Performance Assessment consists of analytical quality assurance phantom models and interfaces to CT simulation frameworks to generate simulated CT images representing different diameters of each phantom.

INSTALLATION

Installation is only required to generate new datasets, a pregenerated dataset can be downloaded from [Zenodo](#), only proceed if you want to generate new simulated datasets.

Requirements

- [Conda](#) package manager e.g. [Miniconda](#)
- [Mac](#), [Linux](#), or [Windows Subsystem for Linux \(WSL\)](#) operating systems described on the [Octave Conda Forge page](#). This package currently uses the Octave-based [Michigan Image Reconstruction Toolbox \(MIRT\)](#)

Installation

```
git clone https://github.com/DIDSR/pediatricIQphantoms
cd pediatricIQphantoms
conda env create --file environment.yml
conda activate pediatricIQphantoms
```

The code block above does the following in 4 lines:

1. Git clones the [pediatricIQphantoms](#) repository
2. Changes the active directory to the repo
3. Creates a new conda environment called “pediatricIQphantoms”
4. Activates the conda environment. This makes the phantom creation library *pediatricIQphantoms* accessible in scripts (see [examples](#)) and via command line calls (see [demo 01](#) and [demo 02](#)).

Test the Installation

```
pytest
```

This runs the [unit tests](#) to verify that installation was successful.

Running Notebooks

Example [computational notebooks](#) have been provided to demonstrate

- [running simulations interactively with python](#)
- [viewing the simulated dataset](#)
- [evaluating pediatric generalizability of denoisers](#)

To run these Notebooks you will need to have [jupyter](#) installed

```
conda install jupyterlab -y
```

Running Shell Scripts

Example shell scripts and config files have been provided to demonstrate non-interactive use of the tool to generate simulated datasets.

- Command line usage for phantom creation and simulation
 - demonstrates command line usage including simulating different scanner configurations and acquisition protocols
 - run with `bash demo_01_phantom_creation.sh`
- Command line usage for more complex simulation experiments with changing acquisition parameters

After Installation

Check out the *Usage* section for detailed information on customizing dataset running_simulations.

Computational notebooks have also been provided to demonstrate how to use pediatricIQphantoms dataset including:

- running CT simulations
- using the dataset to assess denoising performance in pediatric subgroups

CONTENTS

3.1 Usage

3.1.1 Intended Purpose

The pediatricIQphantom phantom generation and CT simulation tool is intended for evaluating the patient size dependence of nonlinear, data-driven image denoising and processing algorithms by providing digital versions of standard image quality phantoms, the [MITA-LCD phantom](#) and CTP404 module of the [Catphan 600 phantom](#). Evaluating the patient size dependence of denoising algorithms is important when considering the performance of these devices in pediatric populations[1]. This is due to the smaller fields of view (FOV) associated with pediatric protocols which alters the image texture, an important factor when training and testing data-drive denoising methods.

Intended users are CT device developers and image denoising and processing software developers. Advanced nonlinear CT image reconstruction and denoising methods (products code [JAK](#), [QIH](#), [LLZ](#) among others) includes statistically iterative, model-based iterative and deep learning-based image reconstruction and denoising methods.

3.1.2 Command Line interface

The quickest and simplest way to start generating new datasets with different parameters is using the command line interface tool `make_phantoms` and modifying the provided configuration toml files. The examples below illustrate different scenarios that can be experimented with using the command line tool and config files.

- [example_01_multiple_recon_kernels.sh](#)

The purpose of this example is to illustrate how to batch several simulations into a single config .toml file and how to update parameters while keeping everything else the same. This example is available in the [examples folder](#) or alternatively can be run using the following terminal command:

```
python make_phantoms.py configs/multiple_recon_kernels.toml
```

The key difference in this config file compared to the [default](#) or [test](#) configs is that multiple simulations are batched in a single config file by repeating the `[[simulation]]` toml header for each new simulation to add to the batch. Note in toml this is referred to as a [nested table](#).

```
# multiple_recon_kernels.toml
# This config file demonstrates how to run 4 unique simulations procedurally varying_
↪ parameters

[[simulation]]
# directories
image_directory = "results/multiple_recon_kernels"
# phantoms
model = ['CCT189'] # <-- current options include ['CTP404', 'MITA-LCD', 'UNIFORM']
```

(continues on next page)

(continued from previous page)

```

diameter = [112, 131, 151, 185, 200, 292, 350] # <-- units in mm
reference_diameter = 200 # <-- diameter in mm of the real phantom for comparison

# acquisition
framework = "MIRT" # CT simulation framework options include ['MIRT']
↳<https://github.com/JeffFessler/mirt>
nsims = 200 # <-- number of simulations to perform with different noise instantiations
ndetectors = 880 # number of detector columns (set it to be large enough to
↳cover the projected FOV to avoid truncation)
nangles = 1160 # <-- number of views in a rotation (na=1160 based on
↳ZengEtAl2015-IEEE-NuclearScience-v62n5:"A Simple Low-Dose X-Ray CT Simulation From
↳High-Dose Scan")
aec_on = true # (aec built in to ped xcat) <-- 'aec' = automatic exposure control,
↳when `true`, it ensures constant noise levels for all `patient_diameters` (see
↳`reference_dose_level` for more info)
add_noise = true # <-- if true adds Poisson noise, noise magnitude set by
↳`reference_dose_level`, noise texture set by reconstructed field of view (currectly
↳fov = 110# patient_diameter)
full_dose = 3e5 # <-- units of photons per pixel
dose_level = [0.1, 0.25, 1.00] # <-- units of photons, this expression is evaluated
↳by matlab, so keep in this format '[xx, yy, zz]`

# acquisition geometry # CT geometry (the following parameter values simulate Siemens
↳Force)
sid = 595 # (mm) source-to-isocenter distance (value based on AAPM LDCT data
↳dicom header)
sdd = 1085.6 # source-to-detector distance
# isocenter-to-detector distance dod = sdd - sid

detector_size = 1 # detector column size
detector_offset = 1.25 # lateral shift of detector

# reconstruction
fov = 340 # <-- FOV in mm of adult protocol used in scanning real physical phantom
↳for comparison
matrix_size = 512 # <-- reconstructed matrix size in pixels (square, equal on both
↳sides)
fbp_kernel = 'hanning,2.05' # 'hanning,xxx', xxx = the cutoff frequency, see
↳fbp2_window.m in MIRT for details.

[[simulation]]

fbp_kernel = 'hanning,0.85'

[[simulation]]

model = ['CTP404'] # <-- current options include [CCT189, CTP404]
fbp_kernel = 'hanning,2.05'
dose_level = [1.0]

[[simulation]]

```

(continues on next page)

(continued from previous page)

```
fbp_kernel = 'hanning,0.85'
```

Here multiple simulations are run, note the repeated header blocks `[[simulation]]` indicate the start of a new experiment. Any parameters set in the first simulation, (the first `[[simulation]]` above), override the default parameters. In each subsequent `[[simulation]]` an new provided settings will update the scan settings, otherwise all other parameters will carry over from the previous simulation.

For example:

```
{'image_directory': 'results/multiple_recon_kernels',
 'model': ['CCT189'],
 'diameter': [112, 131, 151, 185, 200, 292, 350],
 'framework': 'MIRT',
 'nsims': 200,
 'nangles': 1160,
 'aec_on': True,
 'add_noise': True,
 'full_dose': 300000.0,
 'dose_level': [0.1, 0.25, 1.0],
 'sid': 595,
 'sdd': 1085.6,
 'ndetectors': 880,
 'detector_size': 1,
 'detector_offset': 1.25,
 'fov': 340,
 'matrix_size': 512,
 'fbp_kernel': 'hanning,2.05'}
```

In the second simulation in the config file only the *fbp_kernel* is updated

```
[[simulation]]

fbp_kernel = 'hanning,0.85'
```

This results in only updating the *fbp_kernel* element leaving all other elements the same from the previous simulation.

Then by third simulation a new phantom is introduced, CTP404, and we wish to only image it at full dose and with the first of the two kernels being investigated (sharp and smooth):

```
[[simulation]]

model = ['CTP404']
dose_level = [1.0]
fbp_kernel = 'hanning,2.05'
```

Finally by the fourth we repeat the previous simulation but with the second kernel, the smooth kernel

```
[[simulation]]

fbp_kernel = 'hanning,0.85'
```

This is done in parsing the config files using the python dict update method `pediatricIQphantoms.make_phantoms()`

- Reproducing the `pediatricIQphantoms` dataset

warning: this may take several hours to complete (14 hours on my system), but is recommended if you want to reproduce the [pediatricIQphantoms dataset](#) with different scanner or phantom characteristics using the config file editing processes described in the previous example.

```
make_phantoms configs/pediatricIQphantoms.toml
```

Note that this example of the executable `make_phantoms` that is installed and added to your python after installing the `pediatricIQphantoms` python package (see [install instructions](#) for details)

3.1.3 Viewing images

The outputs of the simulation are [DICOM CT images](#). The notebook [01_viewing_images.ipynb](#) discusses ways to view and interact with DICOM images produced in [00_running_simulations.ipynb](#)

`run_batch_sim` outputs simulated datasets in the following directory structure:

```
$ tree results/test -P *_000.dcm | head -n 15
```

```
results/test
├── CCT189
│   ├── diameter112mm
│   │   ├── dose_025
│   │   │   └── fbp hanning205
│   │   │       └── 11.2 cm CCT189_000.dcm
│   │   └── dose_100
│   │       └── fbp hanning205
│   │           └── 11.2 cm CCT189_000.dcm
│   └── diameter292mm
│       ├── dose_025
│       │   └── fbp hanning205
│       │       └── 29.2 cm CCT189_000.dcm
│       └── dose_100
│           └── fbp hanning205
```

CSV files are output with each batched simulation which include file path and all relevant acquisition parameters, for example:

Table 1: Image metadata summary from batch mode simulations

Name	series	effective diameter [cm]	pediatric subgroup	phantom	scanner	Dose [%]	kernel	FOV [cm]	file
11.2 cm CTP404	simulation	11.2	infant	CTP404	Siemens Definition AS+ (simulated)	25	fbp D45	34	pediatricIQphantoms/CTP404/diameter112mm/dose_025/fbp_hanning205/11.2 cm CTP404_000.dcm
11.2 cm CTP404	simulation	11.2	infant	CTP404	Siemens Definition AS+ (simulated)	25	fbp D45	34	pediatricIQphantoms/results/pediatricIQphantoms/CTP404/diameter112mm_hanning205/11.2 cm CTP404_001.dcm
35.0 cm CTP404	noise free	35	infant	CTP404	Siemens Definition AS+ (simulated)		fbp D45	34	pediatricIQphantoms/CTP404/diameter350mm/35.0 cm CTP404_noisefree.dcm
35.0 cm CTP404	ground truth	35	infant	CTP404	Siemens Definition AS+ (simulated)		fbp D45	34	pediatricIQphantoms/CTP404/diameter350mm/35.0 cm CTP404_groundtruth.dcm

Drag and drop the unzipped dataset into [3D Slicer](#) to automatically load the dataset into [Slicer's DICOM database](#)

3D Slicer 5.6.2

File Edit View Help

DATA DCM SAVE Modules: >> >> >> >> >> >> >> >>

DICOM database

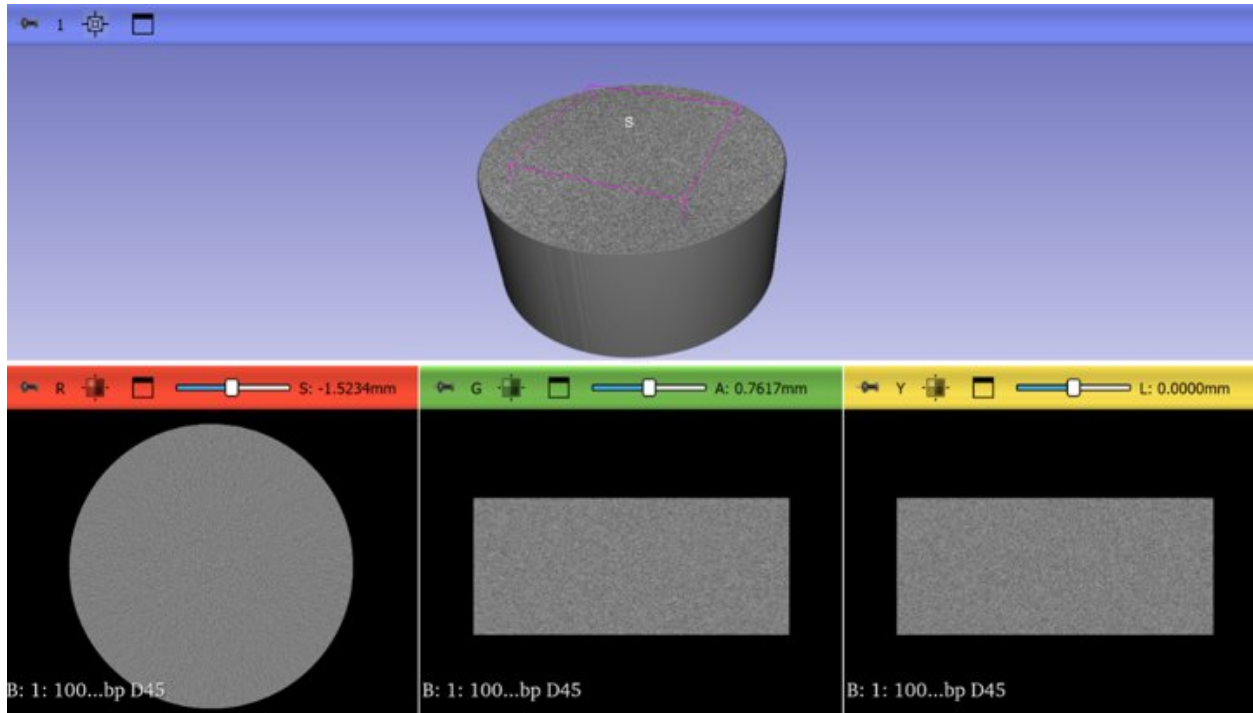
Patients: Studies: Series:

Patient name	Patient ID	Birth date	Sex	Studies	Last study date	Date added
35.0 cm uniform	023		O	4	2024-05-23	2024-05...:11.264
29.2 cm uniform	022		O	4	2024-05-23	2024-05...:10.863
21.6 cm uniform	021		O	4	2024-05-23	2024-05...:10.454
20.0 cm uniform	020		O	4	2024-05-23	2024-05...:10.043
18.5 cm uniform	019		O	4	2024-05-23	2024-05...:05.048
15.1 cm uniform	018		O	4	2024-05-23	2024-05...:04.720
13.1 cm uniform	017		O	4	2024-05-23	2024-05...:04.355
11.2 cm uniform	016		O	4	2024-05-23	2024-05...:03.978
35.0 cm MITA-LCD	015		O	4	2024-05-23	2024-05...:03.581
29.2 cm MITA-LCD	014		O	4	2024-05-23	2024-05...:03.281
21.6 cm MITA-LCD	013		O	4	2024-05-23	2024-05...:02.845
20.0 cm MITA-LCD	012		O	4	2024-05-23	2024-05...:02.590

Study date	Study ID	Study description	Series	Date added
2024-05-23	63	35.0 cm MITA-LCD_noisefree fbp D45		2024-05...:03.583
2024-05-23	62	35.0 cm MITA-LCD_groundtruth		2024-05...:03.582
20240523	61	100% dose 35.0 cm MITA-LCD fbp D45	1	2024-05...:03.763
20240523	60	25% dose 35.0 cm MITA-LCD fbp D45	1	2024-05...:03.584

Series #	Series description	Modality	Size	Count	Date added
1	25% dose 35.0 cm MITA-LCD fbp D45	CT	512x512	166	2024-05...:03.584
1	100% dose 35.0 cm MITA-LCD fbp D45	CT	512x512	168	2024-05...:03.763
1	35.0 cm MITA-LCD_groundtruth	CT			2024-05...:03.582

Advanced



Examples of other viewers:

1. [Radiant](#) *Note*: this is a freemium product
2. [SNAP ITK](#) Originally developed for 3D medical imaging segmentation
3. [Fiji/ImageJ](#) originally developed for 2D biomedical imaging analysis

3.1.4 Conclusions

This section introduced several ways to interact with pediatricIQphantom simulation tools and view the DICOM images produced by `run_batch_sim`.

Next see the notebooks for practical examples `00_running_simulations.ipynb`

3.1.5 References

1. Nelson BJ, Kc P, Badal A, Jiang L, Masters SC, Zeng R. Pediatric evaluations for deep learning CT denoising. *Medical Physics*. 2024;51(2):978-990. doi:10.1002/mp.16901
2. Zeng R, Lin CY, Li Q, et al. Performance of a deep learning-based CT image denoising method: Generalizability over dose, reconstruction kernel, and slice thickness. *Med Phys*. 2022;49(2):836-853. doi:10.1002/mp.15430

3.2 Running Simulations

3.2.1 Introduction

This notebook demonstrates how to use the CT simulation tools included in the `pediatricIQphantoms` repo and how these tools can be used to build pediatric IQ phantom datasets. These tools were used to generate the `pediatricIQphantoms` dataset using the `pediatricIQphantoms.toml` configuration file.

This tutorial will introduce:

1. *individual simulations* - this section discusses key technique parameters and how to change them

2. *batch mode simulation* - this section introduces how to systematically iterate through parameters to build a dataset
3. *command line tool* this section combines themes from the previous sections into a command line tool using configuration files consisting of the same technique parameters discussed in sections *individual simulations* and *batch mode simulation*

3.2.2 Individual Simulations

The main tool for running individual simulations is the `CTobj` class which both holds the key parameters for running and describing the scan as well as having methods to [run the scan](#) and [write the output to dicom](#)

As shown in the Class signature above, most attributes have default values provided, these values correspond to a Siemen's Definition AS+ scanner as described in [Zeng et al 2022](#). Thus it can be defined and `run()` with no further initialization attributes. For this example we'll specify the CTP404 phantom and keep all other defaults.

```
[1]: !pip install seaborn -q
!pip install matplotlib -q
```

```
[2]: from pediatricIQphantoms import CTobj
ct = CTobj('CTP404')
ct
```

```
[2]: <class 'pediatricIQphantoms.make_phantoms.CTobj'> 20.0 cm CTP404 I0: 300000.0
```

You can also experiment with changing the lesion size for different phantoms, note CCT189 expects a length 4 list since the lesions are each different sizes, whereas CTP404 takes a single values since all inserts are equally sized. See [documentation](#) for details.

```
import numpy as np
lesion_diameter = np.array([0.015, 0.025, 0.035, 0.05])*2 # in order of 14 HU, 7 HU, ↵
↵5 HU, and 3 HU
# lesion_diameter = [3, 5, 7, 10]; # in mm
ct.lesion_diameter = lesion_diameter
```

```
[3]: ct.run()
```

```
[3]: <class 'pediatricIQphantoms.make_phantoms.CTobj'> 20.0 cm CTP404 I0: 300000.0
Recon: (1, 512, 512) 34.0 cm FOV
Projections: (900, 580)
```

You can access the reconstruction and projection data as numpy arrays via the object attributes:

```
[4]: type(ct.recon), ct.recon.shape, type(ct.projections), ct.projections.shape
```

```
[4]: (numpy.ndarray, (1, 512, 512), numpy.ndarray, (900, 580))
```

```
[5]: fname = ct.write_to_dicom('test.dcm')
```

```
import pydicom
dcm = pydicom.read_file(fname[0])
dcm.PatientName
```

```
[5]: '20.0 cm CTP404'
```

```
[6]: !pip install ipywidgets -q
```

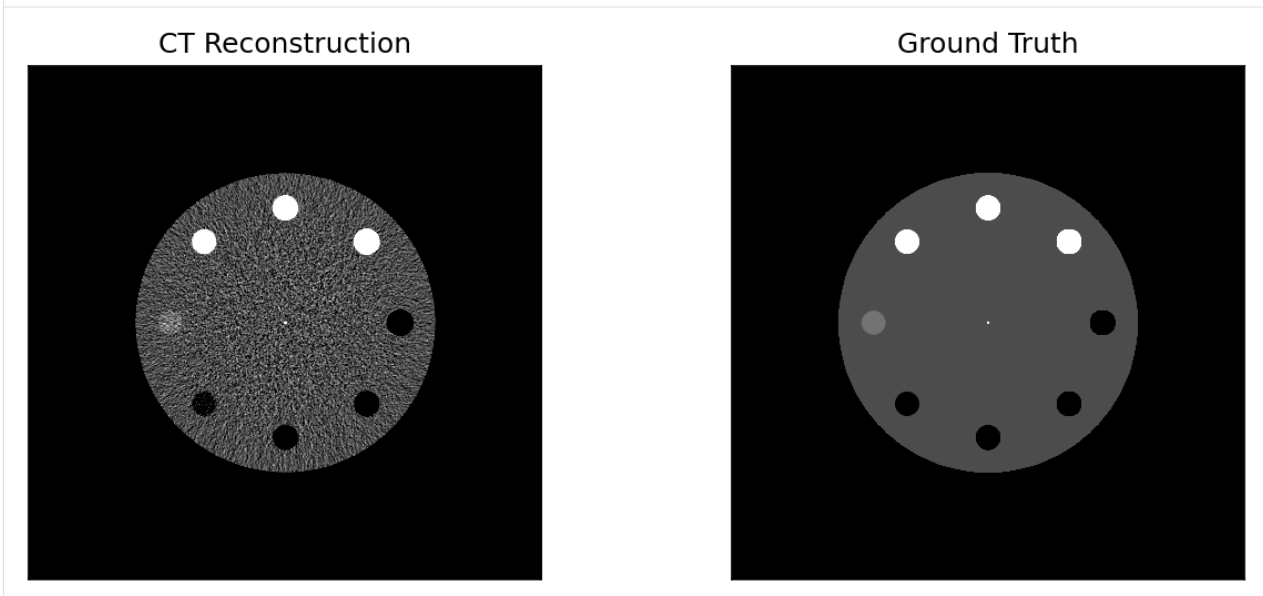
```
[7]: import matplotlib.pyplot as plt

from utils import ctshow
```

The `CTobj` [object](https://pediatriciqphantoms.readthedocs.io/en/latest/function_reference.html#pediatricIQphantoms.CTobj) also holds the `groundtruth` attribute following the reconstruction

```
[8]: plt.figure(figsize=(10,4), dpi=150)
plt.subplot(1,2,1)
win, lvl = 100, 20
display_settings = (win, lvl)
ctshow(ct.recon[0], display_settings)
plt.title('CT Reconstruction')
plt.subplot(1,2,2)
ctshow(ct.groundtruth, display_settings)
plt.title('Ground Truth')
```

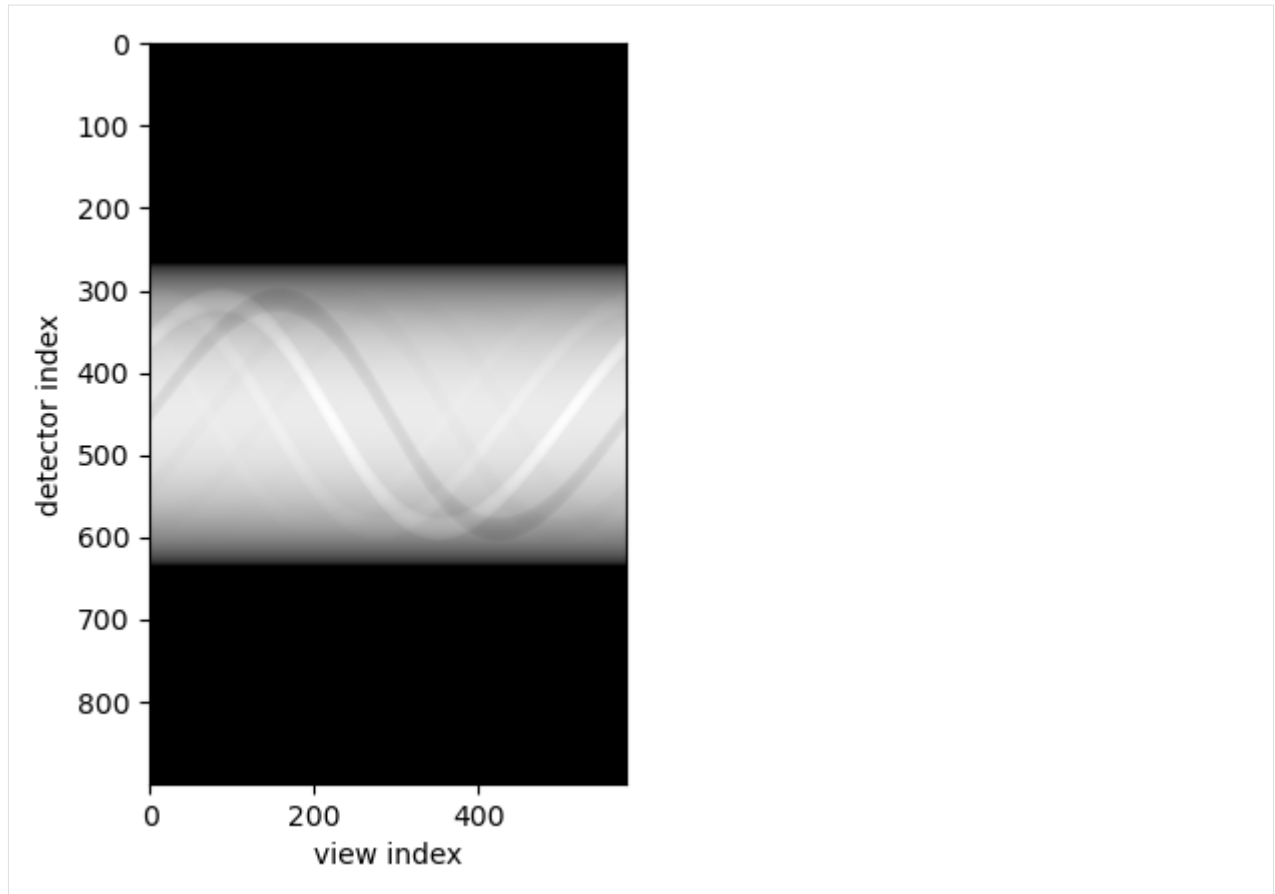
```
[8]: Text(0.5, 1.0, 'Ground Truth')
```



The `CTobj` object also holds the projections

```
[9]: import numpy as np
plt.imshow(-np.log(ct.projections), cmap='gray')
plt.xlabel('view index')
plt.ylabel('detector index')
```

```
[9]: Text(0, 0.5, 'detector index')
```



Different Phantoms

Here we demonstrate the different phantoms available by defining three CTobj objects each with different phantoms

```
[10]: ctp404 = CTobj('CTP404').run()
      mital_lcd = CTobj('MITA-LCD', I0=3e7).run() # <-- higher dose to show low contrast_
      ↪ lesions
      uniform = CTobj('uniform').run()
```

(continues on next page)

(continued from previous page)

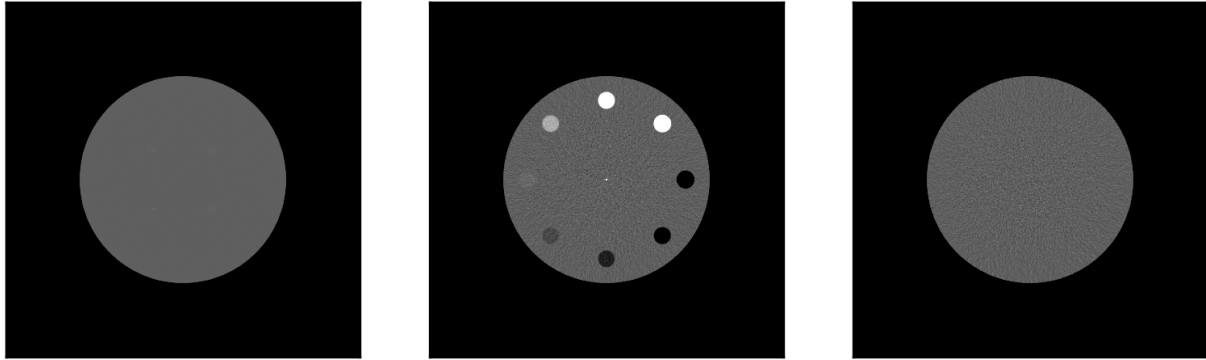
```
[11]: plt.figure(dpi=300)

display_settings = 'soft tissues'

plt.subplot(1,3,1)
ctshow(mital_lcd.recon, display_settings)

plt.subplot(1,3,2)
ctshow(ctp404.recon)

plt.subplot(1,3,3)
ctshow(uniform.recon, display_settings)
plt.show()
```



Different Dose Levels

Next let's run a simulation using the `Uniform` phantom but scanned at different dose levels to see how noise trends compare with increasing dose level.

Due to the relationship of $noise \propto \sqrt{1/N}$, thus we expect to see **1/2 the noise at 4x dose**

```
[12]: import numpy as np
max_dose = 1e5 # photons per detector see MIRT https://github.com/JeffFessler/mirt
dose_levels = np.linspace(1, 4, 3)
print(dose_levels)
doses = dose_levels*max_dose
res_list = [CTobj('Uniform', I0=I0, nsims=2).run() for I0 in doses]

[1.  2.5  4. ]
```

(continues on next page)

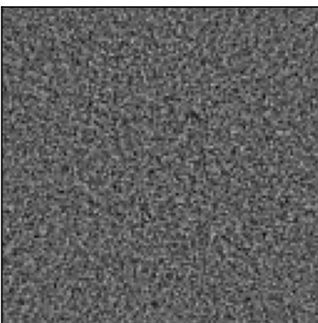
(continued from previous page)

[13]: res_list

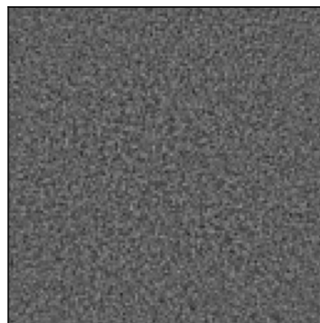
```
[13]: [<class 'pediatricIQphantoms.make_phantoms.CTobj'> 20.0 cm Uniform I0: 100000.0
Recon: (2, 512, 512) 34.0 cm FOV
Projections: (900, 580),
<class 'pediatricIQphantoms.make_phantoms.CTobj'> 20.0 cm Uniform I0: 250000.0
Recon: (2, 512, 512) 34.0 cm FOV
Projections: (900, 580),
<class 'pediatricIQphantoms.make_phantoms.CTobj'> 20.0 cm Uniform I0: 400000.0
Recon: (2, 512, 512) 34.0 cm FOV
Projections: (900, 580)]
```

```
[14]: nres = 3
std_noise = []
plt.figure(figsize=(nres*4, nres), dpi=150)
for idx, res in enumerate(res_list):
    nz, nx, ny = res.recon.shape
    if idx//3 == 0:
        plt.subplot(1, nres, idx+1)
        ctshow(res.recon[0, nx//2-nx//8:nx//2+nx//8, nx//2-nx//8:nx//2+nx//8], 'soft_
↪tissues')
        nz, nx, ny = res.recon.shape
        std_noise.append(res.recon[0, nx//2-nx//8:nx//2+nx//8, nx//2-nx//8:nx//2+nx//8].
↪std())
    plt.title(F'Photons/pixel {doses[idx]:.1g}\nstd: {std_noise[idx]:.2f} HU')
```

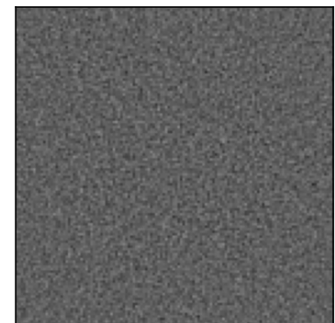
Photons/pixel 1e+05
std: 39.63 HU



Photons/pixel 2e+05
std: 25.26 HU

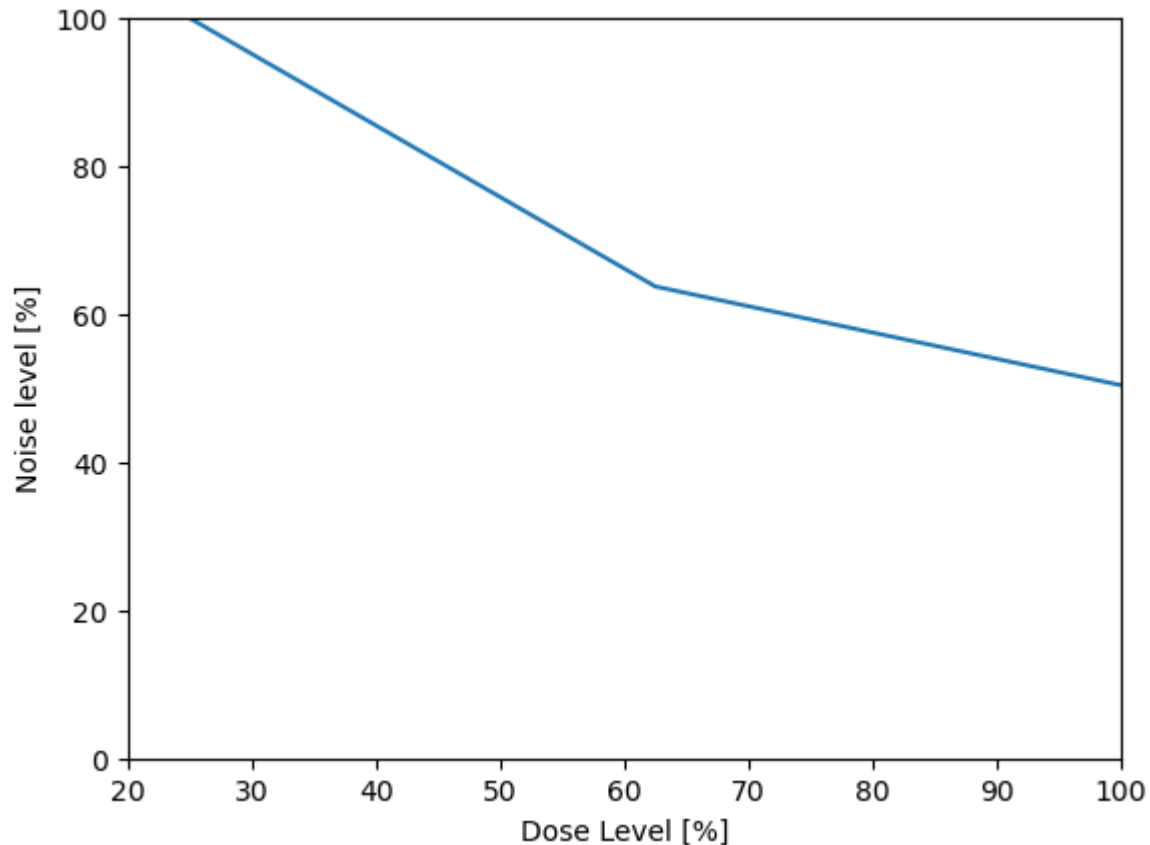


Photons/pixel 4e+05
std: 19.98 HU



```
[15]: rel_doses = doses/doses.max()*100
std_noise = np.array(std_noise)
rel_noise = std_noise/std_noise.max()*100
plt.plot(rel_doses, rel_noise)
plt.xlim([20, 100])
plt.ylim([0, 100])
plt.ylabel('Noise level [%]')
plt.xlabel('Dose Level [%]')
```

```
[15]: Text(0.5, 0, 'Dose Level [%]')
```



These figures and plots confirm that the [MIRT simulations](#) are performing as expected in terms of dose-noise relationships.

Below shows how we can write out to DICOM and then the next example explores changing recon kernels.

```
[16]: from pathlib import Path
writedir=Path('data/test')
fnames=[]
for res in res_list:
    fname = res.write_to_dicom(writedir / f'{res.seriesname}.dcm')
    fnames += fname
len(fnames)
```

[16]: 6

Recon Type

In MIRT convolution kernels are specified as <window function>, <cutoff frequency mm> for example a typical sharp kernel can be defined as 'hanning,2.05' where a **Hann window** with a cutoff (MTF frequency of 2.05 mm. The table below provides a few Siemen's equivalent kernels based upon matching 50% and 10% MTF cutoffs (Zeng et al 2022).

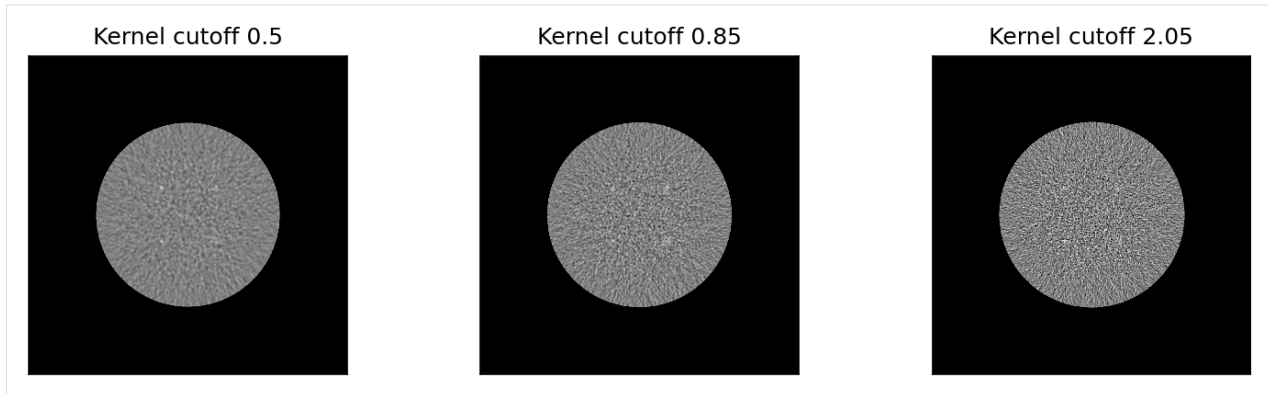
Description	MIRT	Siemen's Equivalent
Sharp	hanning,2.05	D45
Smooth	hanning,0.85	B30

The following experiment runs simulations on increasingly higher cutoff frequency kernels ranging from smooth to sharp.
|

```
[17]: cutoff_freq = [0.5, 0.85, 2.05]

res_list = [CTobj('MITA-LCD', I0=1e6, fbp_kernel=f'hanning,{freq:1.2f}').run() for
↪freq in cutoff_freq]
```

```
[18]: nres = len(res_list)
plt.figure(figsize=(nres*4, nres), dpi=150)
for idx, res in enumerate(res_list):
    plt.subplot(1, nres, idx+1)
    ctshow(res.recon, (40, 0))
    plt.title(f'Kernel cutoff {cutoff_freq[idx]}')
```



Phantom Size and FOV

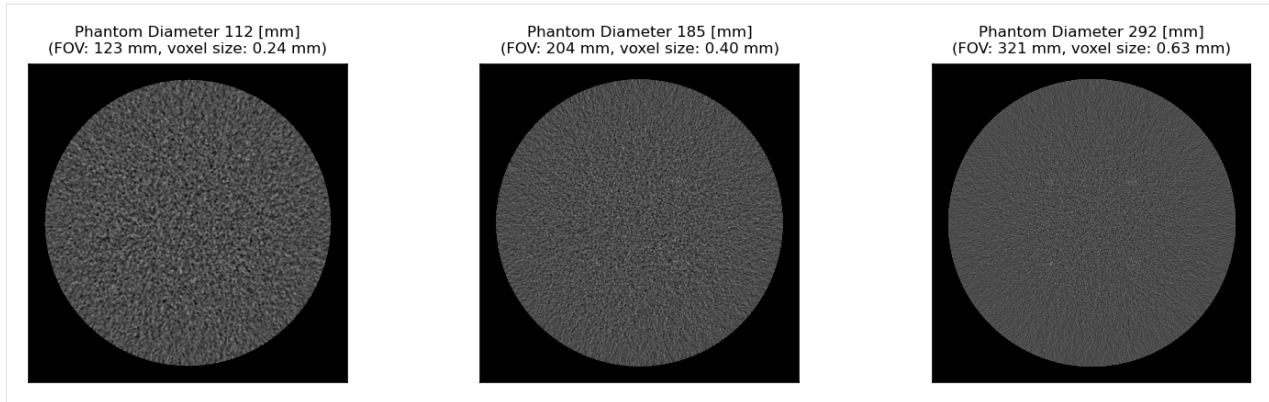
Finally let's see how noise texture changes as a result of imaging different sized phantoms with body fitting field of views (FOV)s.

```
[19]: phantom_diameters = [112, 185, 292]

res_list = [CTobj('MITA-LCD', I0=1e6, patient_diameter=d).run() for d in
            phantom_diameters]
```

```
[20]: fov = np.array(phantom_diameters)*1.1 # in mm
       dx = fov/res.recon.shape[-1]

nres = len(res_list)
plt.figure(figsize=(nres*4, nres), dpi=150)
for idx, res in enumerate(res_list):
    plt.subplot(1, nres, idx+1)
    ctshow(res.recon, 'liver')
    plt.title(f'Phantom Diameter {phantom_diameters[idx]} [mm] \n (FOV: {fov[idx]:2.0f}
            mm, voxel size: {dx[idx]:2.2f} mm)', fontsize=8)
```



By default when FOV is not specified, it defaults to a body fitting 1.1x the phantom diameter. Note how FOV changes the relative pixel size

3.2.3 Batch Mode Simulations

The following introduces how to interactively construct batch simulations in a python environment using `run_batch_sim`

```
[21]: from pediatricIQphantoms import run_batch_sim
```

Batch mode can be run by either entering in parameter lists for `model`, `diameter`, or `dose_level`

```
[22]: metadata = run_batch_sim(image_directory='data/interactive_batch', dose_level=[0.1, ↵  
↵0.25, 0.5, 1], verbose=False);
```

```
/  
↵home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/notebooks/data/interactive_ba  
MITA-LCD Simulation series 0/1
```

(continues on next page)

(continued from previous page)

[23]: `metadata.head()`

```

[23]:      Name Patient ID \
0  20.0 cm MITA-LCD      000
1  20.0 cm MITA-LCD      000
2  20.0 cm MITA-LCD      000
3  20.0 cm MITA-LCD      000
4  20.0 cm MITA-LCD      000

      Study Name Study ID      series \
0  30000.0 photons 20.0 cm MITA-LCD 10% dose fbp D45      0 simulation
1  75000.0 photons 20.0 cm MITA-LCD 25% dose fbp D45      1 simulation
2  150000.0 photons 20.0 cm MITA-LCD 50% dose fbp...      2 simulation
3  300000.0 photons 20.0 cm MITA-LCD 100% dose fb...      3 simulation
4  300000.0 photons 20.0 cm MITA-LCD 100% dose fb...      3 noise free

      effective diameter [cm] age [year] pediatric subgroup phantom \
0      20.0      2      infant MITA-LCD
1      20.0      2      infant MITA-LCD
2      20.0      2      infant MITA-LCD
3      20.0      2      infant MITA-LCD
4      20.0      2      infant MITA-LCD

      scanner Dose [%] recon kernel FOV [cm] \
0 Siemens Definition AS+ (simulated)      10 fbp fbp D45      34.0
1 Siemens Definition AS+ (simulated)      25 fbp fbp D45      34.0
2 Siemens Definition AS+ (simulated)      50 fbp fbp D45      34.0
3 Siemens Definition AS+ (simulated)      100 fbp fbp D45      34.0
4 Siemens Definition AS+ (simulated)      None fbp fbp D45      34.0

      repeat      file
0      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
1      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
2      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
3      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
4      0 /home/brandon.nelson/Dev/Regulatory_Science_To...

```

Alternatively, you can load in a config file and change it as needed and then input to `run_batch_sim` using the ``**` operator <<https://realpython.com/python-kwargs-and-args/>>`__

```

[24]: import tomli
with open('sample_config_file.toml', 'rb') as f:
    config_file = tomli.load(f)
config_file

```

```
[24]: {'simulation': [{'image_directory': 'results/test',
  'model': ['MITA-LCD', 'CTP404'],
  'diameter': [112, 292],
  'reference_diameter': 200,
  'framework': 'MIRT',
  'nsims': 2,
  'aec_on': True,
  'add_noise': True,
  'full_dose': 300000.0,
  'dose_level': [0.25, 1.0],
  'sid': 595,
  'sdd': 1085.6,
  'ndetectors': 880,
  'nangles': 100,
  'detector_size': 1,
  'detector_offset': 1.25,
  'fov': 340,
  'matrix_size': 256,
  'fbp_kernel': 'hanning,2.05'}]}
```

```
[25]: config_file.keys(), len(config_file['simulation'])
```

```
[25]: (dict_keys(['simulation']), 1)
```

Note: the toml config files are structured to allow multiple simulations with updating select parameters in each simulation, see [Documentation/Usage](#) for more details.

This for this example, we want to take the first simulation since `run_batch_sim` expects only flat dictionaries, str, float, and int arguments.

```
[26]: config = config_file['simulation'][0]
config
```

```
[26]: {'image_directory': 'results/test',
  'model': ['MITA-LCD', 'CTP404'],
  'diameter': [112, 292],
  'reference_diameter': 200,
  'framework': 'MIRT',
  'nsims': 2,
  'aec_on': True,
  'add_noise': True,
  'full_dose': 300000.0,
  'dose_level': [0.25, 1.0],
  'sid': 595,
  'sdd': 1085.6,
  'ndetectors': 880,
  'nangles': 100,
  'detector_size': 1,
  'detector_offset': 1.25,
  'fov': 340,
  'matrix_size': 256,
  'fbp_kernel': 'hanning,2.05'}
```

```
[27]: config['image_directory'] = 'data/interactive_batch'
      config['dose_level'] = [0.1, 0.25, 0.5, 1]
      metadata = run_batch_sim(verbose=False, **config)

/
↳home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/notebooks/data/interactive_batch
MITA-LCD Simulation series 0/2
```

(continues on next page)

(continued from previous page)

CTP404 Simulation series 1/2

(continues on next page)

(continued from previous page)



[28]: metadata.head()

```
[28]:      Name Patient ID \
0  11.2 cm MITA-LCD      000
1  11.2 cm MITA-LCD      000
2  11.2 cm MITA-LCD      000
3  11.2 cm MITA-LCD      000
4  11.2 cm MITA-LCD      000

      Study Name Study ID      series \
0  30000.0 photons 11.2 cm MITA-LCD 10% dose fbp D45      0 simulation
1  30000.0 photons 11.2 cm MITA-LCD 10% dose fbp D45      0 simulation
2  75000.0 photons 11.2 cm MITA-LCD 25% dose fbp D45      1 simulation
3  75000.0 photons 11.2 cm MITA-LCD 25% dose fbp D45      1 simulation
4  150000.0 photons 11.2 cm MITA-LCD 50% dose fbp...      2 simulation

      effective diameter [cm] age [year] pediatric subgroup phantom \
0      11.2      2      infant MITA-LCD
1      11.2      2      infant MITA-LCD
2      11.2      2      infant MITA-LCD
3      11.2      2      infant MITA-LCD
4      11.2      2      infant MITA-LCD

      scanner Dose [%] recon kernel FOV [cm] \
0 Siemens Definition AS+ (simulated)      10 fbp fbp D45      34.0
1 Siemens Definition AS+ (simulated)      10 fbp fbp D45      34.0
2 Siemens Definition AS+ (simulated)      25 fbp fbp D45      34.0
3 Siemens Definition AS+ (simulated)      25 fbp fbp D45      34.0
```

(continues on next page)

(continued from previous page)

```

4 Siemens Definition AS+ (simulated)      50 fbp fbp D45      34.0

  repeat                                     file
0      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
1      1 /home/brandon.nelson/Dev/Regulatory_Science_To...
2      0 /home/brandon.nelson/Dev/Regulatory_Science_To...
3      1 /home/brandon.nelson/Dev/Regulatory_Science_To...
4      0 /home/brandon.nelson/Dev/Regulatory_Science_To...

```

3.2.4 Command Line Interface Tool: `make_phantoms`

Finally, datasets can be made in batch mode using the command line tool `make_phantoms` <https://github.com/DIDSR/pediatricIQphantoms/blob/main/src/pediatricIQphantoms/make_phantoms.py#L390> that should be available globally in your terminal after pip installing the `pediatricIQphantoms` package.

```

[29]: !make_phantoms -help

usage: make_phantoms [-h] [config]

Make Pediatric IQ Phantoms: command line interface

positional arguments:
  config      input is a configuration .toml file containing simulation
              parameters (see configs/defaults.toml as an example)

options:
  -h, --help  show this help message and exit

```

For this example we will use `sample_config_file.toml` as quick example

```

[30]: !cat sample_config_file.toml

# test.toml
[[simulation]]

# directories
image_directory = "results/test"
# phantoms
model = ['MITA-LCD', 'CTP404'] # <-- current options include ['CTP404', 'MITA-LCD',
↳ 'UNIFORM']
diameter = [112, 292] # <-- units in mm
reference_diameter = 200 # <-- diameter in mm of the real phantom for comparison

# acquisition
framework = "MIRT" # CT simulation framework options include ['MIRT']
↳ <https://github.com/JeffFessler/mirt>
nsims = 2 # <-- number of simulations to perform with different noise instantiations
aec_on = true # (aec built in to ped xcat) <-- 'aec' = automatic exposure control,
↳ when `true`, it ensures constant noise levels for all `patient_diameters` (see
↳ `reference_dose_level` for more info)
add_noise = true # <-- if true adds Poisson noise, noise magnitude set by
↳ `reference_dose_level`, noise texture set by reconstructed field of view (currectly
↳ fov = 110# patient_diameter)

```

(continues on next page)

(continued from previous page)

```

full_dose = 3e5 # <-- units of photons per pixel
dose_level = [0.25, 1.00] # <-- units of photons, this expression is evaluated by
↳matlab, so keep in this format '[xx, yy, zz]'

# acquisition geometry # CT geometry (the following parameter values simulate Siemens
↳Force)
sid = 595          # (mm) source-to-isocenter distance (value based on AAPM LDCT data
↳dicom header)
sdd = 1085.6      # source-to-detector distance
# isocenter-to-detector distance dod = sdd - sid

ndetectors = 880      # number of detector columns (set it to be large enough to
↳cover the projected FOV to avoid truncation)
nangles = 100        # number of views in a rotation
                    # (na=1160 based on ZengEtAl2015-IEEE-NuclearScience-v62n5:"A
↳Simple Low-Dose X-Ray CT Simulation From High-Dose Scan")

detector_size = 1     # detector column size
detector_offset = 1.25 # lateral shift of detector

# reconstruction

fov = 340 # <-- FOV in mm of adult protocol used in scanning real physical phantom
↳for comparison
matrix_size = 256 # <-- reconstructed matrix size in pixels (square, equal on both
↳sides)
fbp_kernel = 'hanning,2.05' # 'hanning,xxx', xxx = the cutoff frequency, see
↳fbp2_window.m in MIRT for details.
                    #'hanning,2.05' approximate a sharp kernel D45 in Siemens
↳Force.
                    #'hanning,0.85' approximate a smooth kernel B30 in
                    #Siemens Force.

```

```
[31]: !make_phantoms sample_config_file.toml
```

```

/
↳home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/notebooks/results/test
MITA-LCD Simulation series 0/2

```

The variable "irtmdir" is not set, so trying default, assuming that you launched matlab from the irt install directory.

You may need to edit setup.m or adjust your path otherwise.

Assuming you installed irt in directory "/"

```
↳home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-
```

Path setup for irt appears to have succeeded.

Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?

Warn: sino_geom 156: called from "ct_sim : 26"

```
lesion_diameter =
```

(continues on next page)

(continued from previous page)

```
0.015000 0.025000 0.035000 0.050000

ct_sim, simulation: [1/2]
ct_sim, simulation: [2/2]

The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

0.015000 0.025000 0.035000 0.050000

ct_sim, simulation: [1/2]
ct_sim, simulation: [2/2]

The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

0.015000 0.025000 0.035000 0.050000

ct_sim, simulation: [1/1]

The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
```

(continues on next page)

(continued from previous page)

```

Assuming you installed irt in directory "/"
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

    0.015000    0.025000    0.035000    0.050000

ct_sim, simulation: [1/2]
ct_sim, simulation: [2/2]

```

The variable "irtdir" is not set, so trying default, assuming that you launched matlab from the irt install directory. You may need to edit setup.m or adjust your path otherwise.

```

Assuming you installed irt in directory "/"
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

    0.015000    0.025000    0.035000    0.050000

ct_sim, simulation: [1/2]
ct_sim, simulation: [2/2]

```

The variable "irtdir" is not set, so trying default, assuming that you launched matlab from the irt install directory. You may need to edit setup.m or adjust your path otherwise.

```

Assuming you installed irt in directory "/"
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

    0.015000    0.025000    0.035000    0.050000

ct_sim, simulation: [1/1]
CTP404 Simulation series 1/2

```

(continues on next page)

(continued from previous page)

```
The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
```

```
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
```

```
Path setup for irt appears to have succeeded.
```

```
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
```

```
Warn: sino_geom 156: called from "ct_sim : 26"
```

```
lesion_diameter =
```

```
    0.015000    0.025000    0.035000    0.050000
```

```
ct_sim, simulation: [1/2]
```

```
ct_sim, simulation: [2/2]
```

```
The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
```

```
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
```

```
Path setup for irt appears to have succeeded.
```

```
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
```

```
Warn: sino_geom 156: called from "ct_sim : 26"
```

```
lesion_diameter =
```

```
    0.015000    0.025000    0.035000    0.050000
```

```
ct_sim, simulation: [1/2]
```

```
ct_sim, simulation: [2/2]
```

```
The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
```

```
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
```

```
Path setup for irt appears to have succeeded.
```

```
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
```

(continues on next page)

(continued from previous page)

```
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =
```

```
    0.015000    0.025000    0.035000    0.050000
```

```
ct_sim, simulation: [1/1]
```

```
The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
```

```
You may need to edit setup.m or adjust your path otherwise.
```

```
Assuming you installed irt in directory "/"
```

```
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
```

```
Path setup for irt appears to have succeeded.
```

```
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
```

```
Warn: sino_geom 156: called from "ct_sim : 26"
```

```
lesion_diameter =
```

```
    0.015000    0.025000    0.035000    0.050000
```

```
ct_sim, simulation: [1/2]
```

```
ct_sim, simulation: [2/2]
```

```
The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
```

```
You may need to edit setup.m or adjust your path otherwise.
```

```
Assuming you installed irt in directory "/"
```

```
↪home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
```

```
Path setup for irt appears to have succeeded.
```

```
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
```

```
Warn: sino_geom 156: called from "ct_sim : 26"
```

```
lesion_diameter =
```

```
    0.015000    0.025000    0.035000    0.050000
```

```
ct_sim, simulation: [1/2]
```

```
ct_sim, simulation: [2/2]
```

(continues on next page)

(continued from previous page)

```

The variable "irtmdir" is not set, so trying default, assuming
that you launched matlab from the irt install directory.
You may need to edit setup.m or adjust your path otherwise.
Assuming you installed irt in directory "/"
→home/brandon.nelson/Dev/Regulatory_Science_Tools/pediatricIQphantoms/src/pediatricIQphantoms/mirt-r
Path setup for irt appears to have succeeded.
Warn: sino_geom 154: strip_width=0 is rarely realistic! are you sure?
Warn: sino_geom 156: called from "ct_sim : 26"
lesion_diameter =

    0.015000    0.025000    0.035000    0.050000

ct_sim, simulation: [1/1]

```

3.2.5 Conclusions

In this notebook we walked through how to run *individual simulations* and *batched simulations* interactively as well as how to run and modify configuration files to be run noninteractively using the command line interface.

Next see the notebook on [viewing the simulation images](#) or use the simulations to *evaluate pediatric denoising generalizability*.

3.3 Evaluating Denoising Efficiency

3.3.1 Introduction

This notebook demonstrates how the simulated image quality phantoms generated in *the first demo notebook* can be used to evaluate pediatric generalizability of CT denoising models. Pediatric generalizability refers to how well a model performs across pediatric subgroups, please see [Nelson et al., 2023](#) for more.

Subgroup	Age Range	Waist Diameter Range
Newborn	≤ 1 mo	≤ 11.5 cm
Infant	> 1 mo & < 2 yrs	> 11.5 cm & ≤ 16.8 cm
Child	> 2 yrs & ≤ 12 yrs	> 16.8 cm & ≤ 23.2 cm
Adolescent	> 12 yrs & < 21 yrs	> 23.2 cm & < 34 cm
Adult	≥ 22 yrs	≥ 34 cm

Pediatric subgroups investigated defined by age range (Source: [Guidance on Premarket Assessment of Pediatric Medical Devices](#)) with representative effective waist diameters ([AAPM Task Group 220](#)). level

3.3.2 Noise Reduction in Uniform Phantoms

This notebook focuses on how to use the uniform water phantom (similar to the CTP486 module of the [Catphan 600 phantom](#)) to evaluate noise reduction efficiency.

Here noise is defined as local pixel standard deviation σ in a uniform region:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x'_i - \mu)^2}, \text{ where } \mu = \frac{1}{N} \sum_{i=1}^N x'_i \text{ for each pixel } x_i$$

We can thus define noise reduction relative to the input FBP image as:

$$\text{Noise Reduction} = 100\% \times \frac{(\text{FBP} - \text{denoised})}{\text{FBP}}$$

For the purposes of demonstration, this notebook compares a pretrained deep learning denoising model based on a RED-CNN architecture to compare against a baseline FBP image series. To do this a few more libraries need to be installed to load and run the model ([Torchvision](#)) and to visualize the results ([Matplotlib](#) and [Seaborn](#))

```
[1]: !pip install torchvision -q
      !pip install seaborn -q
      !pip install matplotlib -q
```

The following cell loads the necessary libraries and downloads the pregenerated dataset from [Zenodo](#). Note, you can replace the value of `base_dir` with the path of your simulation results from [the first demo notebook](#) or from running the [command line function](#), so long as the path contains the `metadata.csv` file which containing the file paths and file metadata summaries.

```
[2]: from pathlib import Path

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import pydicom
from torchvision.datasets.utils import download_and_extract_archive

base_dir = Path('data')

if not base_dir.exists():
    url = 'https://zenodo.org/records/11267694/files/pediatricIQphantoms.zip'
    download_and_extract_archive(url, download_root=base_dir)
base_dir = base_dir / 'pediatricIQphantoms'
```

One important detail when loading `metadata.csv` is that file paths under the file column are *relative* which means the dataset can be moved without breaking file paths. However, some functions require absolute file paths, these can be added using the `apply` method on the dataframe column (3rd line in the cell below and copied here for reference):

```
meta.file = meta.file.apply(lambda o: base_dir.absolute() / o)
```

```
[3]: meta = pd.read_csv(base_dir / 'metadata.csv')
      meta = meta[meta.phantom == 'uniform']
      meta.file = meta.file.apply(lambda o: base_dir.absolute() / o)
      meta.head()
```

```
[3]:
```

	Name	effective diameter [cm]	age [year]	pediatric subgroup	\	
3544	11.2 cm uniform	11.2	0.083333	newborn		
3545	11.2 cm uniform	11.2	0.083333	newborn		
3546	11.2 cm uniform	11.2	0.083333	newborn		
3547	11.2 cm uniform	11.2	0.083333	newborn		
3548	11.2 cm uniform	11.2	0.083333	newborn		
	phantom	scanner	Dose [%]	recon kernel	FOV [cm]	\
3544	uniform	Siemens Somatom Definition	25.0	fbp	D45	12.0
3545	uniform	Siemens Somatom Definition	25.0	fbp	D45	12.0

(continues on next page)

(continued from previous page)

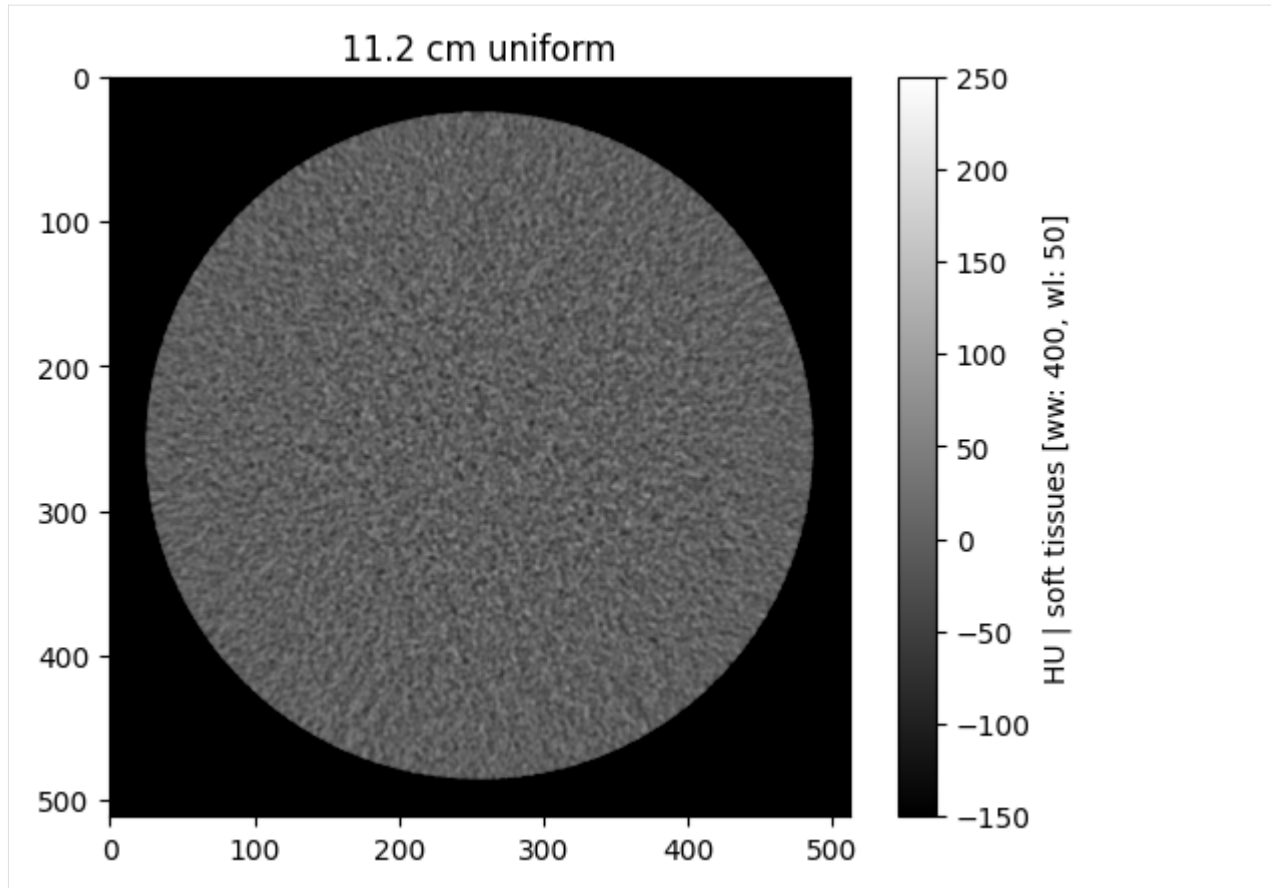
3546	uniform	Siemens	Somatom	Definition	25.0	fbp	D45	12.0
3547	uniform	Siemens	Somatom	Definition	25.0	fbp	D45	12.0
3548	uniform	Siemens	Somatom	Definition	25.0	fbp	D45	12.0
						file	patientid	studyid \
3544	/home/brandon.nelson/Dev/Regulatory_Science_To...						16.0	64
3545	/home/brandon.nelson/Dev/Regulatory_Science_To...						16.0	64
3546	/home/brandon.nelson/Dev/Regulatory_Science_To...						16.0	64
3547	/home/brandon.nelson/Dev/Regulatory_Science_To...						16.0	64
3548	/home/brandon.nelson/Dev/Regulatory_Science_To...						16.0	64
		series	repeat					
3544	simulation		0					
3545	simulation		1					
3546	simulation		2					
3547	simulation		3					
3548	simulation		4					

```
[4]: sorted(meta['FOV [cm]'].unique())
```

```
[4]: [12.0, 14.0, 17.0, 20.0, 24.0, 32.0, 34.0, 39.0]
```

The notebook `utils` folder contains a couple of useful visualization functions, `browse_functions` for static image viewing and `study_viewer` provides an interactive widget

```
[5]: from utils import browse_studies, study_viewer
browse_studies(meta, phantom='uniform', fov=12, dose=100, recon='fbp')
```



```
[6]: study_viewer(meta)

interactive(children=(Dropdown(description='phantom', options=('uniform',), value=
↪ 'uniform'), Dropdown(descrip...
```

3.3.3 Apply a denoiser

The following loads a pretrained RED-CNN implementation and uses it to process the phantom test series and compare denoising performance to the FBP inputs.

```
[7]: import os
import torch
from denoising_networks import RED_CNN

def read_dicom(dcm_file):
    dcm = pydicom.read_file(dcm_file)
    return dcm.pixel_array + float(dcm.RescaleIntercept)

def load_model(save_path, iter_=13000, device='cpu'):
    REDCNN = RED_CNN()
    f = os.path.join(save_path, 'REDCNN_{iter}.ckpt'.format(iter_))
    REDCNN.load_state_dict(torch.load(f, map_location=device))
    return REDCNN
```

(continues on next page)


```
[12]:
      Name effective diameter [cm] age [year] pediatric subgroup \
0 11.2 cm uniform          11.2  0.083333          newborn
1 11.2 cm uniform          11.2  0.083333          newborn
2 11.2 cm uniform          11.2  0.083333          newborn
3 11.2 cm uniform          11.2  0.083333          newborn
4 11.2 cm uniform          11.2  0.083333          newborn

      phantom                scanner Dose [%]   recon kernel FOV [cm] \
0 uniform Siemens Somatom Definition  25.0    fbp    D45    12.0
1 uniform Siemens Somatom Definition  25.0  RED-CNN  D45    12.0
2 uniform Siemens Somatom Definition  25.0    fbp    D45    12.0
3 uniform Siemens Somatom Definition  25.0  RED-CNN  D45    12.0
4 uniform Siemens Somatom Definition  25.0    fbp    D45    12.0

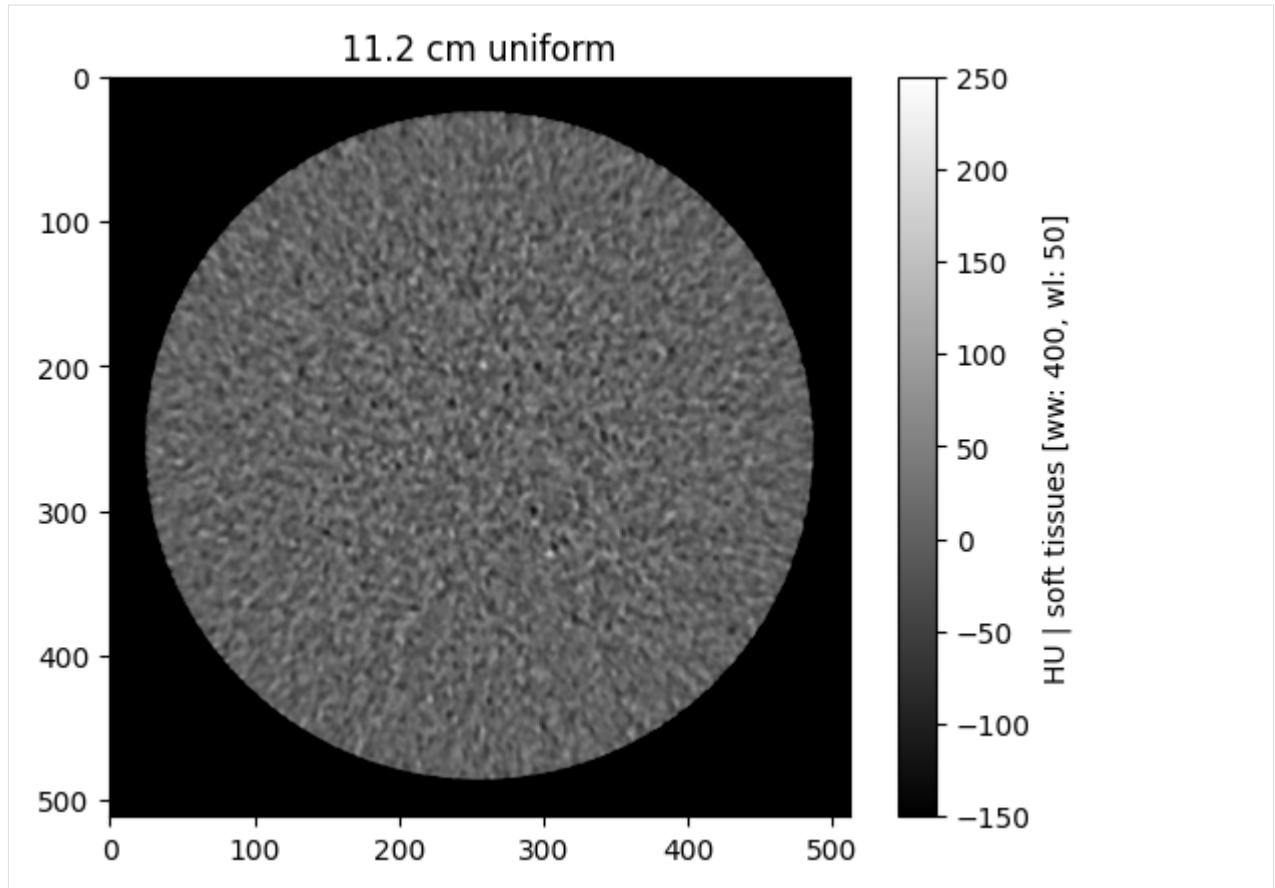
      file patientid studyid \
0 /home/brandon.nelson/Dev/Regulatory_Science_To...  16.0    64
1 /home/brandon.nelson/Dev/Regulatory_Science_To...  16.0    64
2 /home/brandon.nelson/Dev/Regulatory_Science_To...  16.0    64
3 /home/brandon.nelson/Dev/Regulatory_Science_To...  16.0    64
4 /home/brandon.nelson/Dev/Regulatory_Science_To...  16.0    64

      series repeat
0 simulation      0
1 simulation      0
2 simulation      1
3 simulation      1
4 simulation      2
```

3.3.4 Visualizing the Results

We can use the `results.csv` similar to `meta` to visualize the new image series which include the RED-CNN processed results

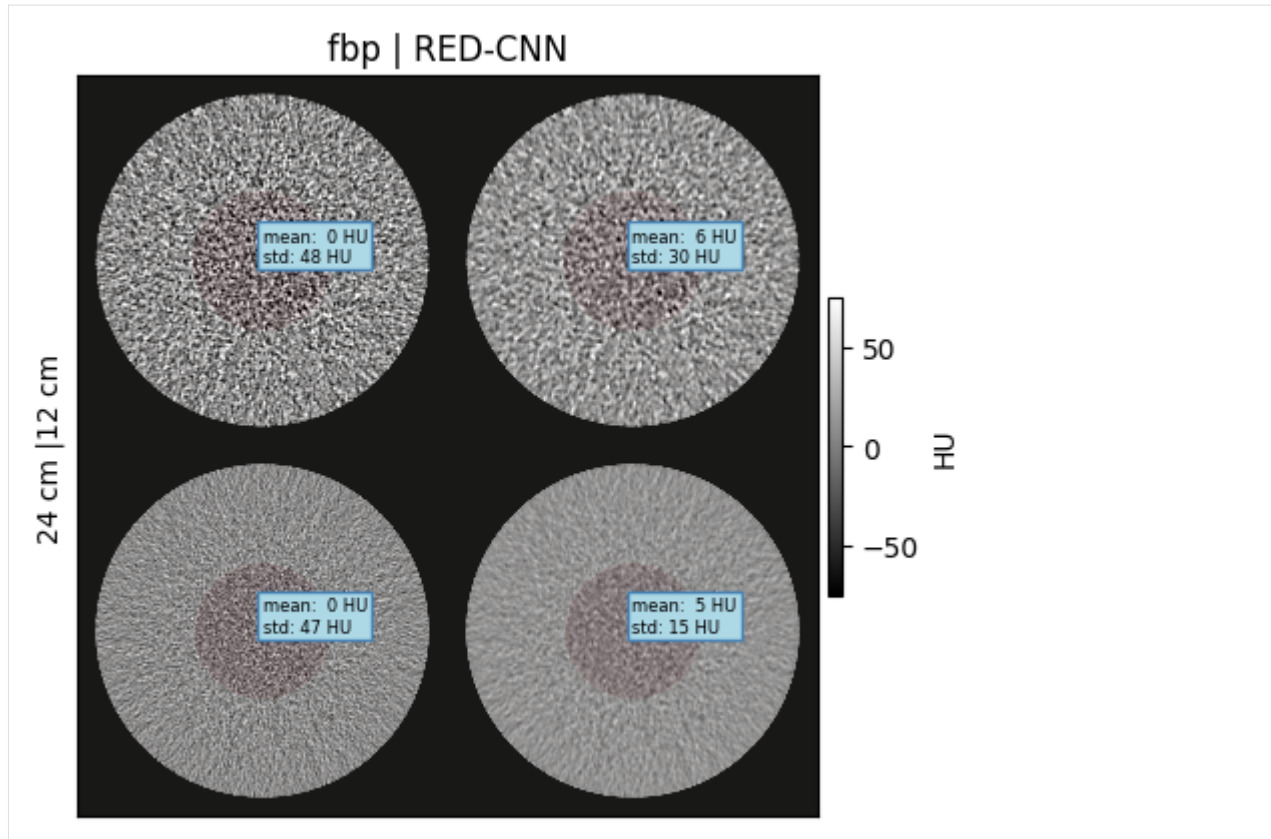
```
[13]: browse_studies(results, phantom='uniform', fov=12, dose=25, recon='RED-CNN')
```



```
[14]: study_viewer(results)

interactive(children=(Dropdown(description='phantom', options=('uniform',)), value=
↪ 'uniform'), Dropdown(descrip...
```

```
[15]: from utils import make_montage
make_montage(results, roi_diameter=0.4, fofs=[12, 24], wwvl=(150, 0))
```



3.3.5 Assessing Noise Reduction

```
[16]: from utils import measure_roi_std
      %timeit measure_roi_std(results.file.iloc[1])
```

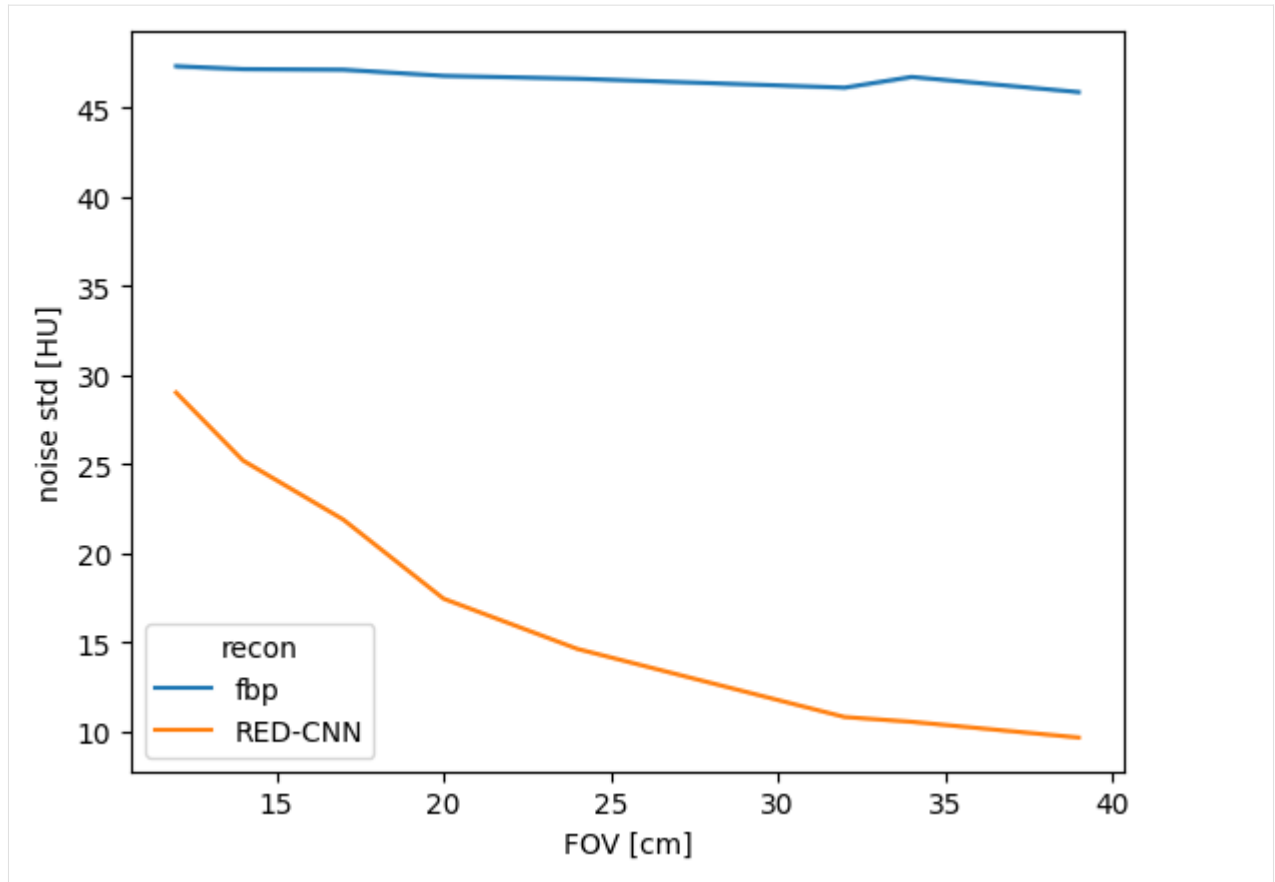
31.1 ms ± 155 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)

This takes about 1-2 min to make all of the noise measurements across 3200 images

```
[17]: results['noise std [HU]'] = results.file.apply(measure_roi_std)
```

```
[18]: import seaborn as sns
      sns.lineplot(data=results, x='FOV [cm]', y='noise std [HU]', hue='recon')
```

```
[18]: <Axes: xlabel='FOV [cm]', ylabel='noise std [HU]'>
```



The RED-CNN model we evaluated was trained on an adult dataset of large fields of view (FOV) thus the above results show the lowest noise levels in the larger FOVs most similar to the training set. We can calculate this as a noise reduction as described previously

$$\text{Noise Reduction} = 100\% \times \frac{(\text{FBP} - \text{denoised})}{\text{FBP}}$$

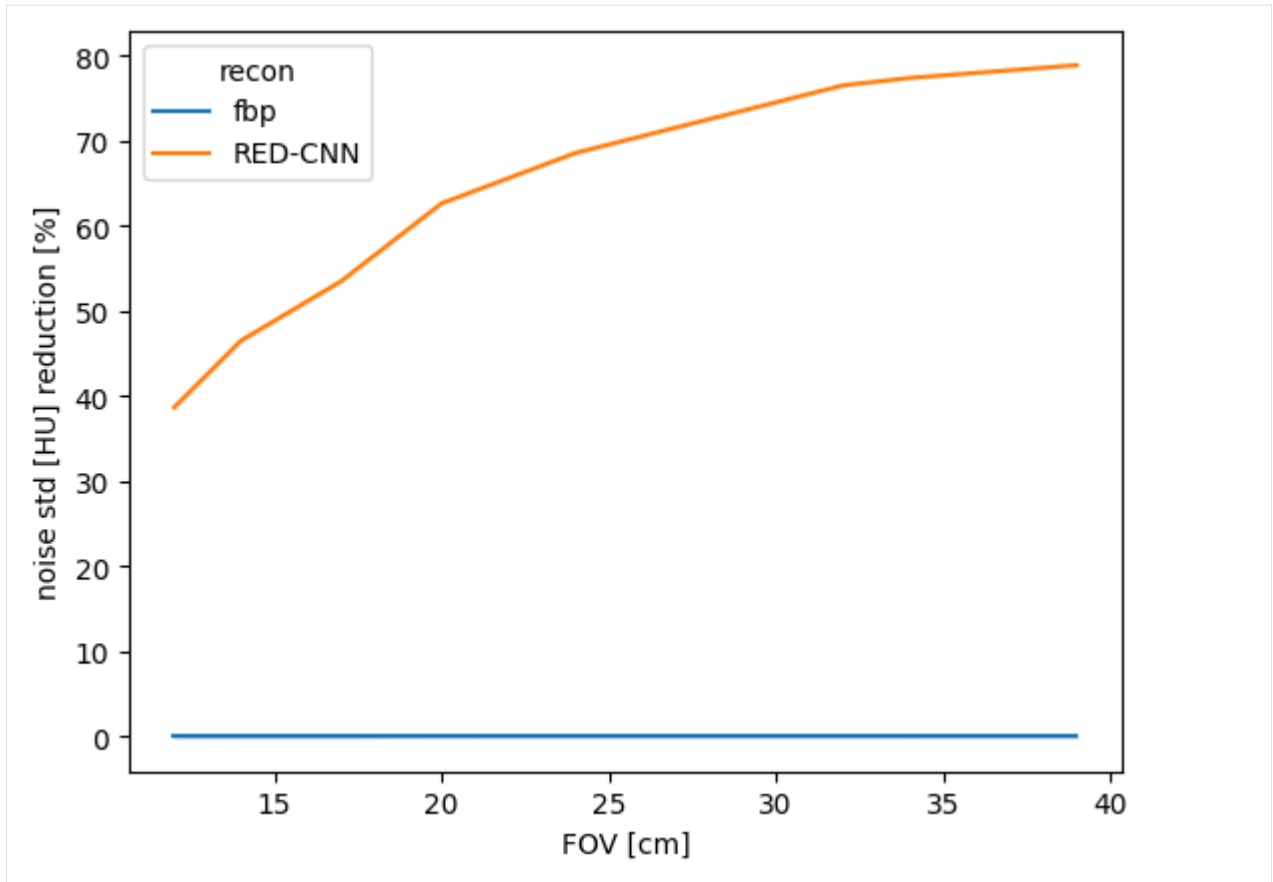
To get the following curve of noise reduction performance as a function of FOV.

```
[19]: from utils import calculate_noise_reduction

[20]: results = calculate_noise_reduction(results)

[21]: sns.lineplot(data=results, x='FOV [cm]', y='noise std [HU] reduction [%]', hue='recon
→')

[21]: <Axes: xlabel='FOV [cm]', ylabel='noise std [HU] reduction [%]'>
```

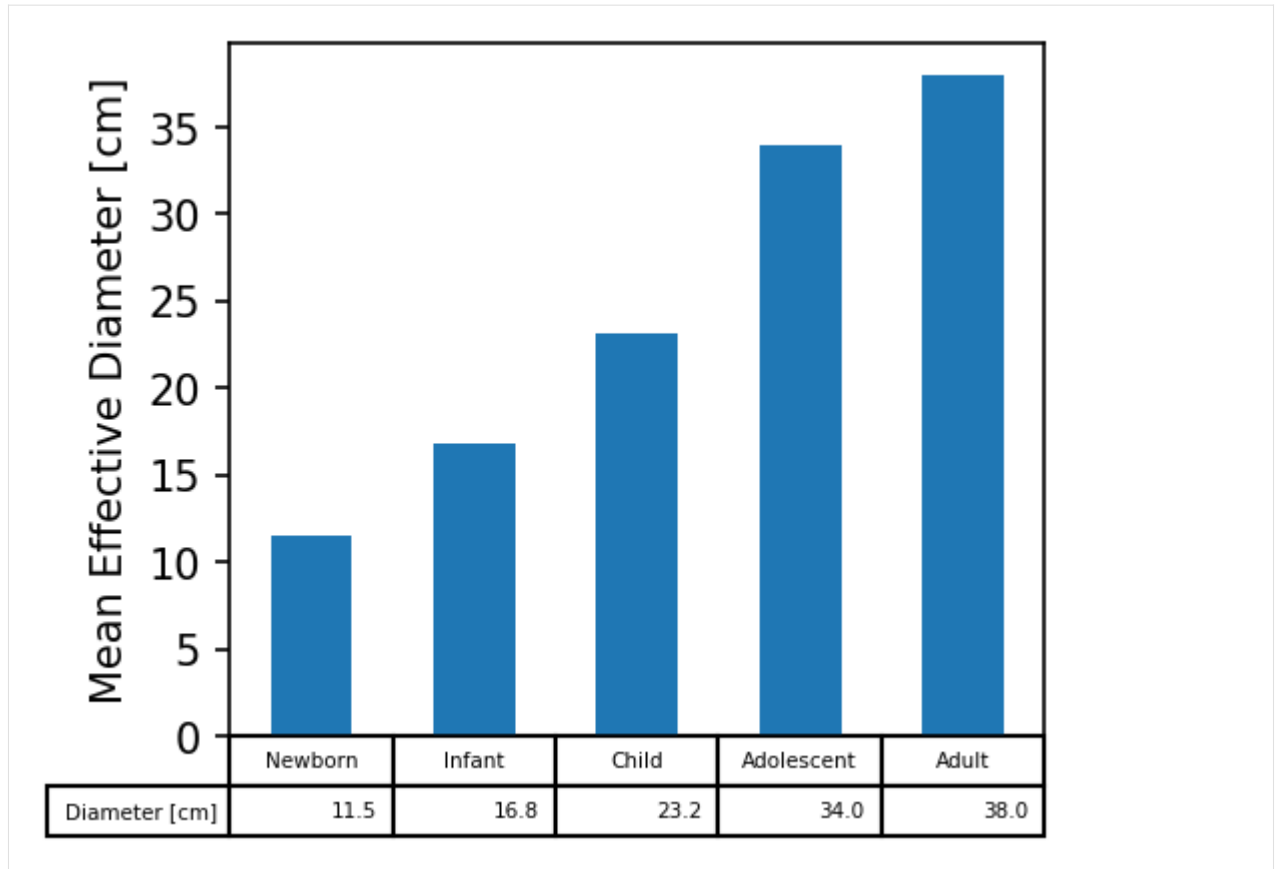


Keep in mind that abdominal imaging protocols typically use body fitting FOVs, thus for smaller patients including pediatric patients, this poorer performance in small FOVs translates to poorer performance in pediatric patients.

To this end we can rebin our results based on the pediatric subgroups define previously, here they are again plotted for comparison:

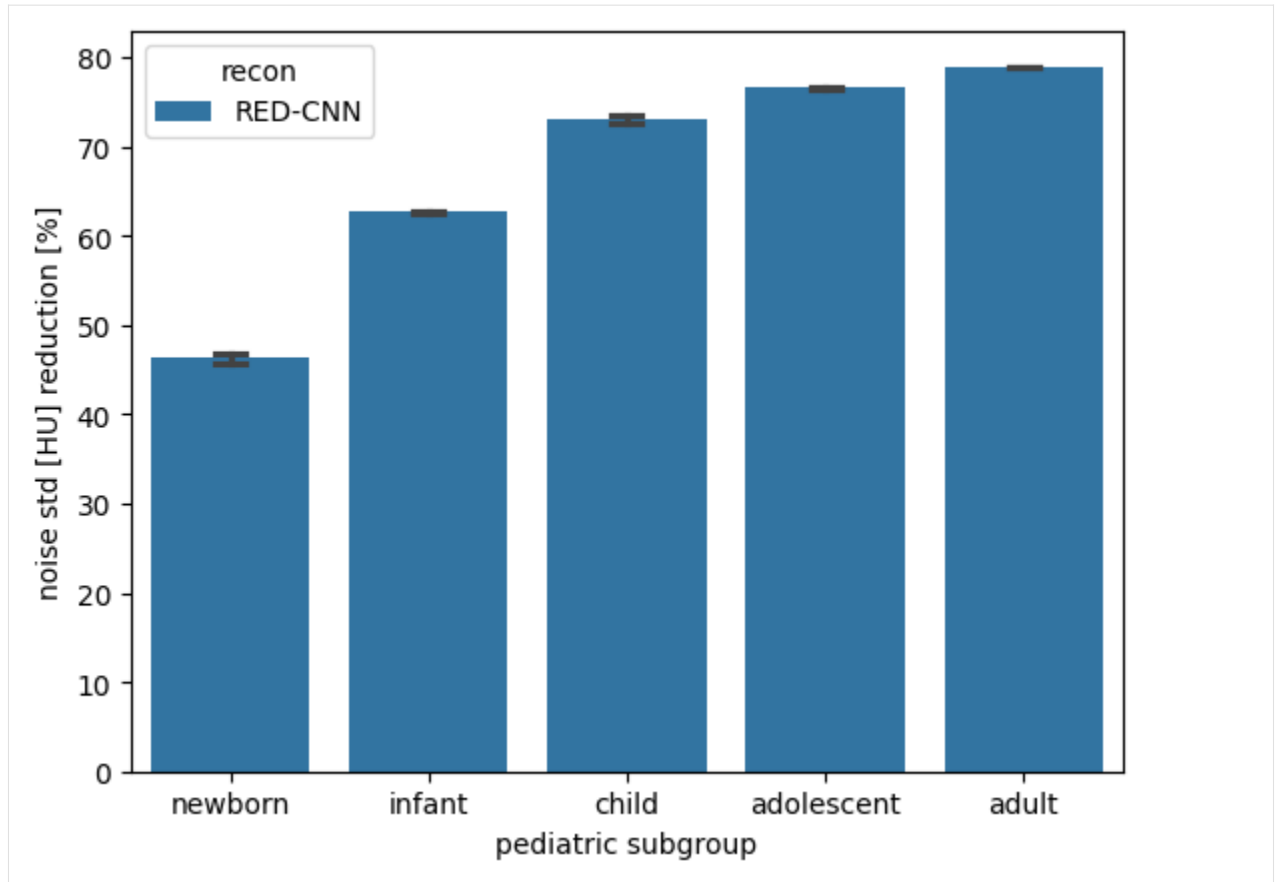
```
[22]: df = pd.DataFrame({'Subgroup': ['Newborn', 'Infant', 'Child', 'Adolescent', 'Adult'],
                        'Diameter [cm]': [11.5, 16.8, 23.2, 34, 38],
                        'Age [yrs]': [0, 2, 12, 21, 38]})
f, ax=plt.subplots(figsize=(3.5,3), dpi=150)
df.plot.bar(ax=ax, x='Subgroup',y='Diameter [cm]', table=True, legend=False)
ax.get_xaxis().set_visible(False)
ax.set_ylabel('Mean Effective Diameter [cm]')

[22]: Text(0, 0.5, 'Mean Effective Diameter [cm]')
```



Below we can compare noise reduction performance as a function of pediatric subgroup following the binning of patient size into these subgroups and see that performance is best in adults and adolescents which better represent the adult training dataset, the [Mayo Clinic Low Dose Grand Challenge Dataset](#)

```
[23]: f, ax = plt.subplots()
sns.barplot(data=results[results.recon != 'fbp'], x='pediatric subgroup', y='noise_
→std [HU] reduction [%]', hue='recon', capsize=0.15, ax=ax)
f.savefig('../pediatric_subgroup_performance.png', dpi=300)
```



3.3.6 Conclusions

Noise reduction is but one of several factors to consider when assessing image quality of CT reconstructions and denoising algorithms, for more information on image quality assessments to justify dose reduction please see [Vaishnav 2018](#). For a review of unique considerations for dose reduction with AI reconstruction and denoising please see [Brady 2023](#).

This concludes our worked examples. Next you can refer to the [API guide](#) for more detailed function documentation or [consider contributing the package](#).

Further Steps

These notebooks have provided simple examples for generating CT datasets representative of pediatric-sized patients with flexibility to simulate a range of x-ray dose and system parameters. Next you are encouraged to explore other tools for assessing task-based image quality using the [Low Contrast Detectability for CT Toolbox](#) which is compatible with the MITA-LCD images made available in [pediatricIQphantoms](#)

3.4 Function Reference

Reference page Functions and classes for interactively running CT simulations

This module provides classes and functions to interactively run CT simulations and make phantom datasets

```
pediatricIQphantoms.run_batch_sim(image_directory: str, model=['MITA-LCD'], diameter=[200],
                                  full_dose=300000.0, dose_level=[1.0], fbp_kernel='fbp', verbose=True,
                                  **kwargs) → DataFrame
```

Running simulations in batch mode

`run_batch_sim` takes lists of parameters (phantoms, diameters, and dose levels) and iterates through all combinations, returns a pandas DataFrame summarizing file metadata including filename

Parameters

- **image_directory** (*str*) – Directory to save simulated outputs
- **model** (*list[str]*) – Optional list, list of phantom models to simulate (can be length 1) options include ['CTP404', 'Uniform', 'MITA-LCD']
- **diameter** – Optional list, list of simulated phantom diameters in mm (can be length 1)
- **full_dose** – Optional int, units of photons per pixel, multiplied by *dose_level* to determine quantum noise level in projections
- **dose_level** – Optional list, units of photons
- **verbose** – Optional bool, whether to print update information

Returns

pd.DataFrame

See *CTobj* for remaining keyword arguments

```
class pediatricIQphantoms.CTobj (phantom='CCT189', patient_diameter=200, reference_diameter=200,
reference_fov=340, I0=300000.0, ndetectors=900, nangles=580,
detector_size=1, sid=595, sdd=1085.6, detector_offset=1.25,
down_sampling=1, has_bowtie=False, add_noise=True, aec_on=True,
matrix_size=512, fov=340, fbp_kernel='hanning,2.05', nsims=1,
lesion_diameter=False, age=0, patientname="", patientid=0, studyname="",
studyid=0, seriesname="", seriesid=0, framework='MIRT')
```

A class to hold CT simulation data and run simulations

Parameters

- **phantom** – phantom object to be scanned, options include ['CCT189', 'CTP404', 'Uniform']
- **patient_diameter** – Optional, effective diameter in mm. See [AAPM TG220](#) for more details.
- **reference_diameter** – Optional, reference effective diameter in mm for computing automatic exposure control (aec) noise index. For example if a 200 mm reference phantom has a noise level of 24 HU at $I_0=3e5$, smaller phantoms will scale I_0 to match that noise level. Note this only applies if `aec_on=True`.
- **I0** – Optional float, fluence at the detector in the projection data for determining quantum noise default $3e5$ units of photons.
- **ndetectors** – Optional int, number of detector columns
- **nangles** – Optional int, number of views in a rotation (na=1160 based on [Zeng et al 2015](#))
- **detector_size** – Optional float, detector column size in mm
- **sid** – Optional float, source-to-isocenter distance in mm
- **sdd** – Optional float, source-to-detector distance in mm
- **detector_offset** – Optional float, lateral shift of detector (1.25 = quarter pixel offset)
- **has_bowtie** – Optional bool, whether to add a patient fitting bowtie (TODO add different bowtie sizes.)

- **add_noise** – Optional bool, if true adds Poisson noise, noise magnitude set by *reference_dose_level*, noise texture set by reconstructed field of view (currently fov = 110% patient_diameter)
- **aec_on** – Optional bool, ‘aec’ = automatic exposure control, when *true*, it ensures constant noise levels for all *patient_diameters* (see *reference_dose_level* for more info)
- **matrix_size** – Optional int, reconstructed matrix size in pixels (square, equal on both sides)
- **fov** – Optional float, reconstructed field of view (FOV) units mm
- **fbp_kernel** – Optional str, *hanning,xxx*, xxx = the cutoff frequency, see `fbp2_window.m` in [MIRT](#) for details. E.g. ‘hanning,2.05’ approximates a sharp kernel D45 in Siemens Force and ‘hanning, 0.85’ approximates a smooth kernel B30.
- **nsims** – Optional int, number of simulations to perform with different noise instantiations
- **lesion_diameter** – Optional bool | list | float | int, if False lesions scale with phantom size, if model==‘CTP404’ lesion_diameter > 1 interpret as absolute diameter in mm, if lesion_diameter < 1 interpret as scale relative to phantom diameter. If model==‘CCT189’ a list of 4 diameters must be provided for the four inserts in contrast order 14, 7, 5, 3 HU. (only applies for CCCT189 MITA and CTP404 phantoms)
- **framework** – Optional, CT simulation framework options include [‘MIRT’]

Returns

None

run (*verbose=False*)

Runs the CT simulation using the stored parameters.

Parameters**verbose** – optional boolean, if True prints out status updates, if False they are suppressed.**write_to_dicom** (*fname: str | Path, groundtruth=False*) → list[Path]

write ct data to DICOM file, returns list of written dicom file names

Parameters

- **fname** – filename to save image to (preferably with ‘.dcm’ or related extension)
- **groundtruth** – Optional, whether to save the ground truth phantom image (no noise, blurring, or other artifacts). If True, ‘self.groundtruth’ is saved, if False (default) *self.recon* which contains blurring (and noise if ‘add_noise=True’)

Returns

list[Path]

3.5 Contributing Guide

[Issue Tracker](#) | [Source Code](#) |

One of the best ways to contribute is by improving our documentation. This project uses [sphinx](#) as a documentor with the [Matlab Domain](#) extension. More resources on documentation can be found at [writethedocs.org](#).

3.5.1 docstrings

The easiest and most effective way to start contributing to the user manual and documentation of the project is by helping improve our [docstrings](#) docstrings, the comments just below the function signature. See some of the [currently documented functions](#) for examples.

3.6 Frequently Asked Questions

Please let us know and we'll add them here!

DEVELOPERS

If you'd like to contribute to the code or documentation of this project, please check out our [Contributing Guide](#) page.

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

p

`pediatricIQphantoms.make_phantoms`, 46