Assignment 1 - Medical Image Analysis

2020/2021

Francesco Corti & Julie Mazzella

12002944 12040967

Graz, May 16, 2021

Contact: Francesco Corti & Julie Mazzella, corti@student.tugraz.at julie.mazzella@student.tugraz.at

1 Introduction

CT scanners are devices that exploit the attenuation of X-rays through a body of interest. X-rays are produced through both collisional and radiative transfer: in the first one, part of the electron's kinetic energy is transferred to another electron in the target medium with which it collides and excites it. When the electron returns to its original state, it emits infra-red radiation and produces heat.

In the second transfer mechanism, we identify two phenomena: the characteristic radiation and the Bremsstrahlung effect. In the former, the incident electron collides with another electron, exciting or ionizing the atom and leaving a hole in the shell. This hole is filled by an electron from a lower shell (with lower energy). Due to this loss of energy, an electromagnetic photon is created, leading to a characteristic X-ray. In the latter phenomenon, a Bremsstrahlung radiation is created after the interaction of an energetic electron with the nucleus of an atom: the positive charge of the nucleus attracts the electron, decelerating it. In this way, the electron loses energy in the form of an electromagnetic photon.

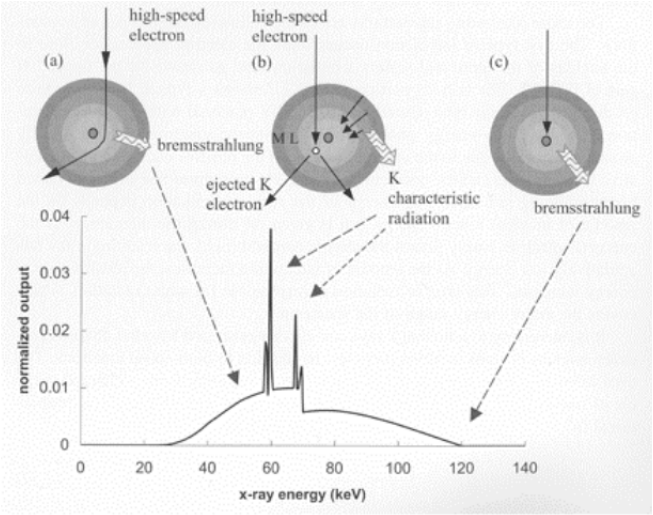

Figure 1: X-ray energy spectrum.

These are the physic principles with which X-ray tubes work: they are based on the acceleration of electrons across a voltage potential in a vacuum. When an electron strikes the tungsten anode, it loses energy by both collisional and radiative transfer. As a consequence, both heat, characteristic radiation and Bremsstrahlung X-rays are produced.

Moreover, the anode is rotating to produce a cooling effect and a collimator is used to focus the beam in a specific direction. Digital detectors are used to transform the radiation outside of the visible spectrum into visible light by using a scintillator and a photomultiplier tube to enhance the result.

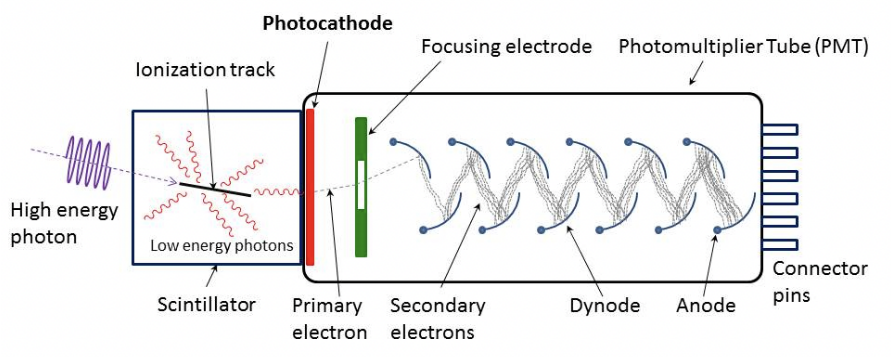

Figure 2: X-ray detector.

To provide a signal, an electron beam creates X-ray photons, which pass through the body of interest and get attenuated depending on the tissue. On the other side of the material, there is a detector that measures the incident X-ray energy.

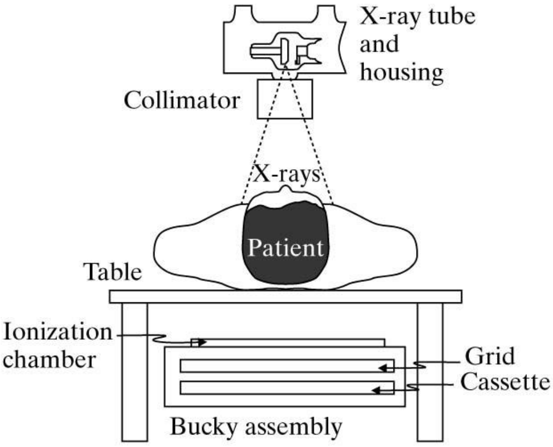

Figure 3: X-ray Based Imaging.

The measurement of the X-rays attenuation along a line between the source and the detector is the base of Computed Tomography. Using a collection of such measurements along all lines within a cross-section, a 2-D image can be mathematically reconstructed by solving an inverse problem.

To obtain the attenuation coefficients, it is necessary to compute the inverse Radon Transform, which is the 1-D projection of a slice, captured by rotating around the body of interest and computing the line integrals. X-ray intensity at any detector is given by this simplified equation:

$$I_d = I_0(\bar{E}) \cdot e^{-\int_0^d \mu(s;\bar{E})ds} \tag{1}$$

where the effective energy E is considered. Using those measurements, it is possible to compute the basic projection given as:

$$g_d = -\ln(\frac{I_d}{I_0}) = \int_0^d \mu(s;\bar{E})ds \tag{2}$$

Therefore, the basic measurement of a CT scanner is a line integral of the linear attenuation coefficient μ at the effective energy of the scanner.

A line in the plane is given as:

$$L(l, \theta) = \{(x, y)|xcos(\theta) + ysin(\theta) = l\} \tag{3}$$

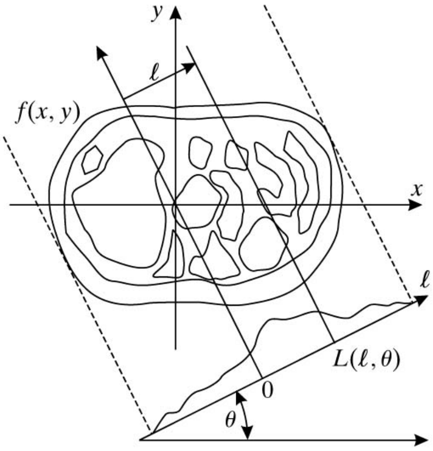

Figure 4: Projection line.

where l is the lateral position of the line and θ is the angle of a unit normal to the line. The line integral of the function $f(x, y)$ is given as:

$$g(l, \theta) = \int_{-\infty}^{+\infty} f(x(s), y(s))ds \tag{4}$$

where

$$x(s) = lcos(\theta) - ssin(\theta) \tag{5}$$

$$y(s) = lsin(\theta) + scos(\theta) \tag{6}$$

By fixing θ, a projection is obtained, while by varying both l and θ the 2-D Radon transform of $f(x, y)$ is computed.

An image of $g(l, \theta)$ with l and θ as coordinates is called a sinogram: it represents the data that is needed to reconstruct $f(x, y)$. To obtain the image back from the sinogram a backward computation step needs to be performed.

Let $g(l, \theta)$ be the 2-D Radon transform of $f(x, y)$ and consider the projection at $\theta = \theta 0$. It is not possible to uniquely determine $f(x, y)$ from one projection, because a single projection maps to an infinite number of $f(x, y)$.

Therefore, by assigning to every point $L(l0, \theta 0)$ on the line the value $g(l0, \theta 0)$, the resulting function is called back-projection image and is given by: $b\theta(x, y) = g(xcos(\theta) + ysin(\theta), \theta)$
By adding up the back-projection images at different angles, we get the back-projection summation image given as:

$$f_b(x, y) = \int_0^{\pi} b_\theta(x, y) d(\theta) \tag{7}$$

In the projection-slice theorem, the relationship between the 1-D Fourier transform of a projection and the 2-D Fourier transform of the object is considered. The 1-D Fourier transform of a projection with respect to l is written as:

$$G(\rho, \theta) = F_{1D}\{g(l, \theta)\} = \int_{-\infty}^{+\infty} g(l, \theta) \cdot e^{-j2\pi\rho l} dl \tag{8}$$

where ρ is the spatial frequency.
The final expression of $G(\rho, \theta)$ recalls the 2-D Fourier transform of $f(x, y)$, defined as:

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \cdot e^{-j2\pi(xu+yv)} dxdy \tag{9}$$

where the variables u and v are the frequency variables in the x and y directions. By introducing:

$$u = \rho cos(\theta) \tag{10}$$

$$v = \rho sin(\theta) \tag{11}$$

we get the relationship:

$$G(\rho, \theta) = F(\rho cos(\theta), \rho sin(\theta)) \tag{12}$$

The equation is known as the projection-slice theorem and it is the basis of two reconstruction methods: the Fourier method and Filtered Back-projection.
The theorem states that the 1-D Fourier transform of a projection is a line passing through the origin of the 2-D Fourier transform of the object, at the projection angle. This can be used to reconstruct the image.
It is possible to take the 1-D Fourier transform of each projection, insert it with the corresponding correct angular orientation into the correct slice of the 2-D Fourier plane. Then, the inverse of the result is taken:

$$f(x, y) = F^{-1}\{G(\rho, \theta)\} \tag{13}$$

The inverse Fourier transform of $F(u, v)$ can be rewritten in polar coordinates as:

$$f(x, y) = \int_0^{2\pi} \int_0^{+\infty} F(\rho cos(\theta), \rho sin(\theta)) \cdot e^{-j2\pi\rho(xcos(\theta)+ysin(\theta))} \rho d\rho d\theta \tag{14}$$

Using the projection-slice theorem, we get:

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{+\infty} |\rho| G(\rho, \theta) \cdot e^{-j2\pi\rho(xcos(\theta)+ysin(\theta))} d\rho d\theta \tag{15}$$

$|\rho|$ acts as a ramp filter in the frequency domain as a high-pass that emphasizes high frequencies and zeroes background values.

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{+\infty} |\rho| G(\rho, \theta) \cdot e^{j2\pi\rho k} d\rho \right]_{k=xcos(\theta)+ysin(\theta)} \tag{16}$$

2 Methods

The assignment aimed to implement different methods for image reconstruction, such as the back-projection, the filtered back-projection, and the projection slice theorem with interpolation. To do that a Python code was implemented, using Numpy [1] and Scipy [4] as scientific computing libraries. The image of interest is shown in Fig 5:

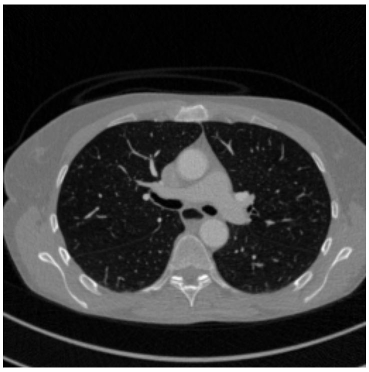

Figure 5: Original image.

After loading the image of size 263x263, a padding was introduced to get a 363x363 image. The padded image was then used to obtain the sinogram, by rotating for 180 angles and summing them up.

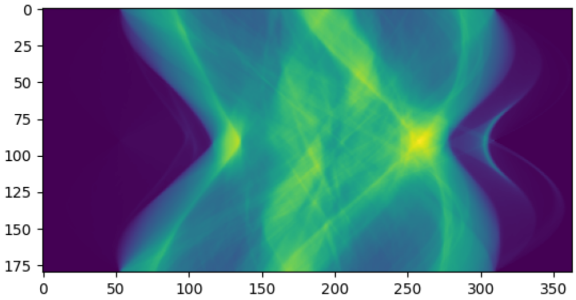

Figure 6: Obtained sinogram.

Afterward, the reconstruction of the original image was performed through the back-projection method, by using the Numpy broadcasting operation.

Moreover, a filtered version of the sinogram was fed to the back-projection algorithm. The filtering was done using a ramp filter shown in Fig 7.

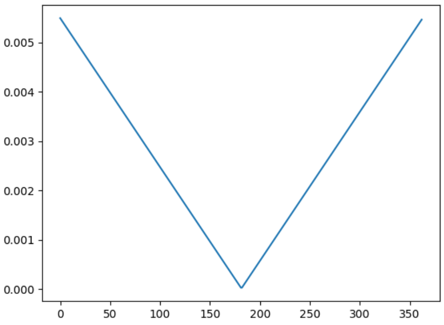

Figure 7: Ramp filter.

Furthermore, a linear interpolation of the 2-D Fourier transform of the sinogram was conducted, and then the original image was obtained by applying the inverse operator on the result.
At last, the root means squared error (RMSE) for the three methods was computed by using the Scikit-learn library [3], to evaluate their trend along with the considered angles.

3 Results

The reconstructed images along the considered angles, obtained through the implemented methods, are shown in the following images:

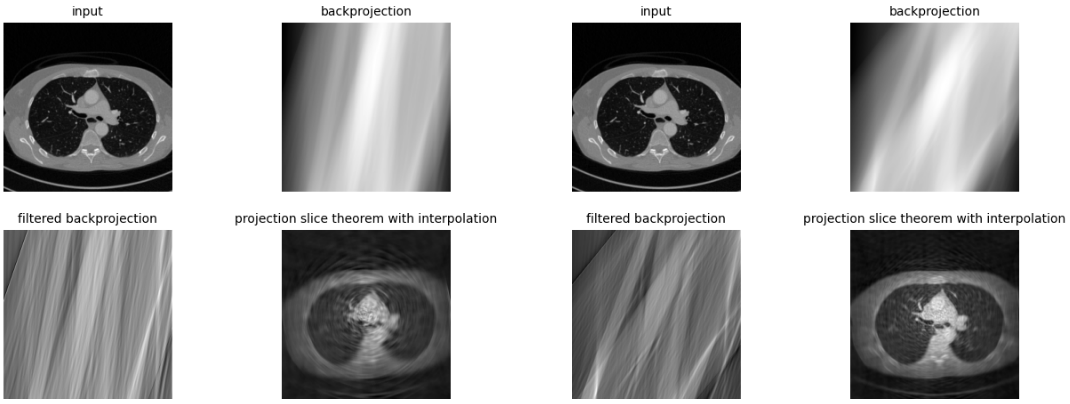
input
backprojection
input
backprojection
filtered backprojection
projection slice theorem with interpolation
filtered backprojection
projection slice theorem with interpolation

Figure 8: Images with 20 and 40 projection lines.

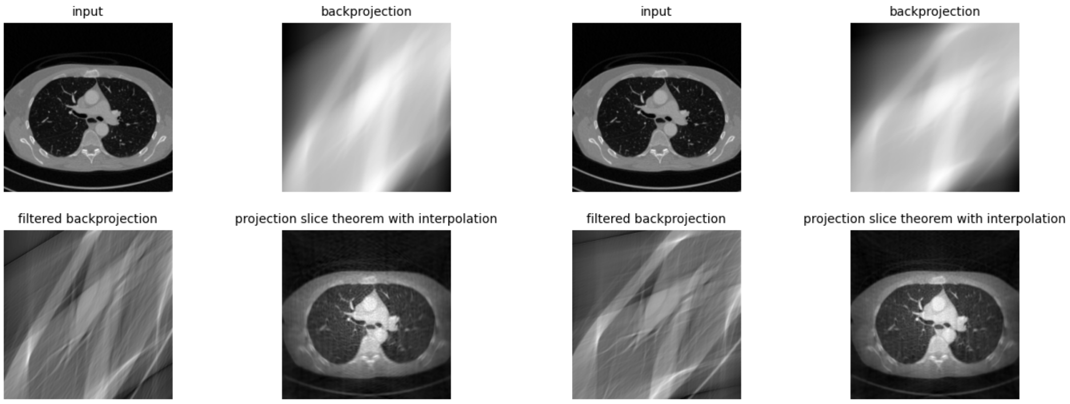
input
backprojection
input
backprojection
filtered backprojection
projection slice theorem with interpolation
filtered backprojection
projection slice theorem with interpolation

Figure 9: Images with 60 and 80 projection lines.

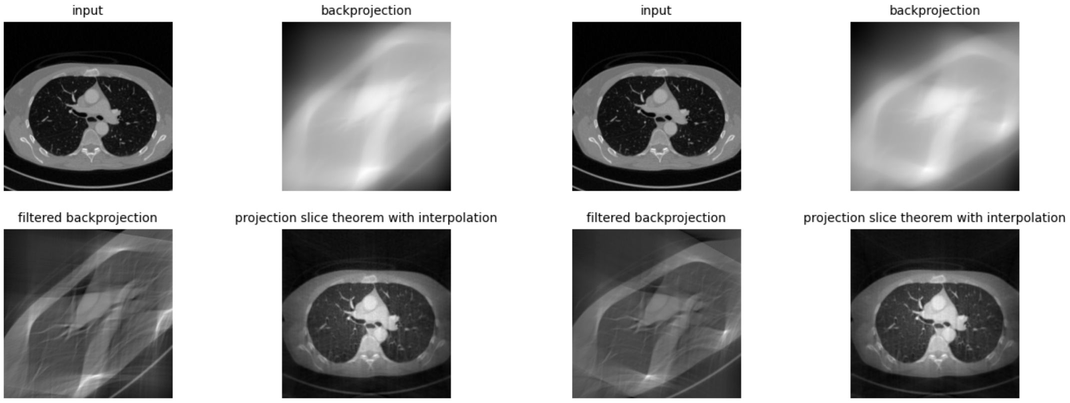
input
backprojection
input
backprojection
filtered backprojection
projection slice theorem with interpolation
filtered backprojection
projection slice theorem with interpolation

Figure 10: Images with 100 and 120 projection lines.

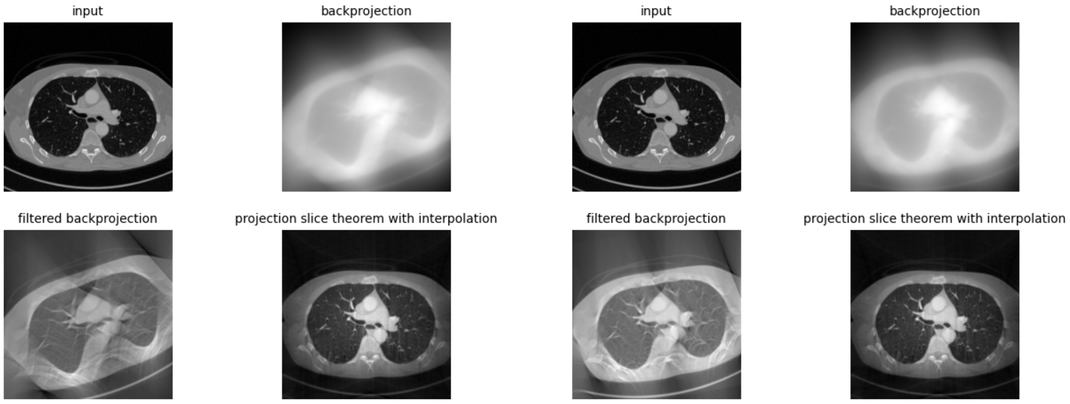
input
backprojection
input
backprojection
filtered backprojection
projection slice theorem with interpolation
filtered backprojection
projection slice theorem with interpolation

Figure 11: Images with 140 and 160 projection lines.

The final images reconstructed with 180 projection lines are shown in Figure 13:

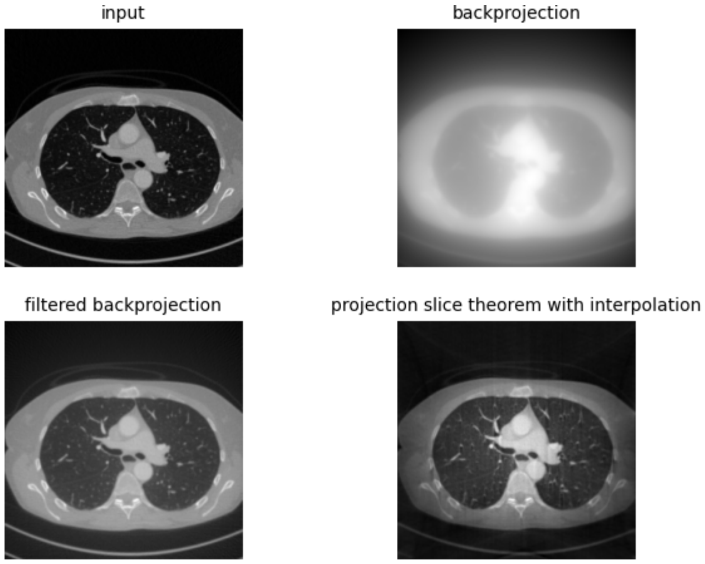

Figure 12: Images with 180 projection lines.

As for the calculation of the RMSE for the three methods, the values along the 180 angles are shown in the table 1:

Angles	Backprojection	Filtered Backprojection	PST with Interpolation
20	0.44404	0.22665	0.24732
40	0.44933	0.20408	0.24195
60	0.44307	0.20120	0.23846
80	0.46912	0.20301	0.23449
100	0.45182	0.16729	0.23903
120	0.44425	0.15943	0.23546
140	0.46007	0.16327	0.21627
160	0.47868	0.16955	0.18479
180	0.47965	0.17587	0.12183

Table 1: RMSE values obtained for different angles.

The trend of the RMSE is also shown in the plot 13. The values were obtained starting from a number of angles equal to 2.

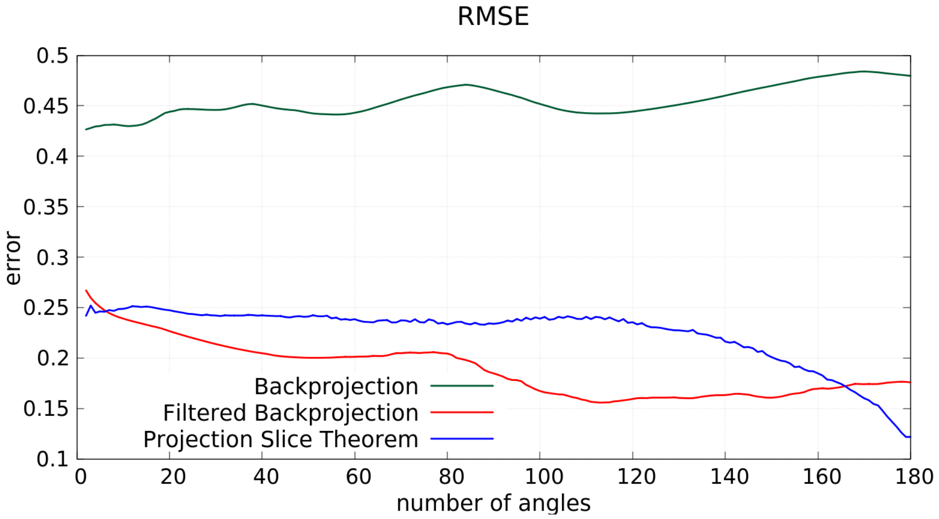

Figure 13: RMSE along the 180 angles.

4 Discussion

As we can observe in Fig 8, Fig 9, Fig 10 and Fig 11 as the number of the considered projection lines increases, the quality of the reconstructed images visually improves.
In the last reconstructed images (Fig 13), it can be observed how the projection slice theorem with interpolation leads to better results compared to the other two reconstruction methods. In particular, the back-projection method is significantly less efficient than the others.
These differences between the methods can be observed also in the RMSE values displayed in table 1. In fact, the values for the back-projection method are significantly higher than the others, for all the different angles.
Concerning the other two methods, the behavior of the RMSE depends on the number of angles considered. In particular, for the Filtered Back-projection, the minimum value of the error is reached with the number of angles equal to 112, after which it starts increasing again. For the Projection Slice Theorem, on the other hand, the behavior is very different: the value of

the error is stable until the number of angles equal to 124. After that, the value of the error starts decreasing significantly, reaching a minimum at the 180th number of angles. This result is confirmed in the fact that, as previously mentioned, at the 180th number of angles, the Slice Projection Theorem provides the best visual result.

5 Conclusion

The aim of the assignment was to learn how to implement reconstruction methods for CT image analysis. During the development phase, we understood how the implementation of the algorithms is strictly related to the theoretical background [2] of the CT techniques. Also, we understood how scientific programming libraries like [4], [3] and [1] can simplify a mathematical domain, such as the one used for this assignment.

References

[1] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[2] Jerry L.Prince and Jonathan M.Lins. *Medical Imaging Signal and Systems.* 2006.

[3] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[4] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.