FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Saku Moilanen

DENOISING COMPUTED TOMOGRAPHY IMAGES WITH 3D-CONVOLUTION BASED NEURAL NETWORKS

Master's Thesis
Degree Programme in Biomedical Engineering
June 2021

Moilanen S. (2021) Denoising Computed Tomography Images with 3D-Convolution Based Neural Networks. University of Oulu, Degree Programme in Biomedical Engineering, 44 p.

ABSTRACT

Low-dose computed tomography (CT) is an imaging technique used in imaging cross-sectional images of the body that minimizes the radiation dose of the patient. Low-dose CT results in larger amounts of noise in the image and therefore in loss of information. Different denoising methods are used to try to reduce the noise corrupting the images.

The aim of this thesis is to research if the temporal correlation of noise between the slices of the computed tomography volumes could be utilized in the denoising of the scans. A convolutional neural network with three-dimensional convolutional layers is trained using publicly available CT images. The images were injected with artificial noise simulating low-dose CT scans. Another network using two-dimensional convolutional layers was also trained for comparison. Different metrics were measured from results of a test dataset to determine the effect of denoising.

The results indicate that utilizing the temporal information of the slices by three-dimensional convolutional layers is especially good in denoising of extremely low-dose CT scans. The denoising results between the different methods were closer to each other when the noise level was lower.

Keywords: low-dose computed tomography, sinogram, deep learning, U-Net

Moilanen S. (2021) Tietokonetomografiakuvien kohinanpoisto käyttäen 3D-konvoluutiopohjaisia neuroverkkoja. Oulun yliopisto, Lääketieteen tekniikan tutkinto-ohjelma, 44 s.

TIIVISTELMÄ

Matalan säteilyannoksen tietokonetomografia on kuvantamismenetelmä, jolla saadaan kuvattua läpileikkauskuvia kehosta samalla minimoiden potilaan säteilyannosta. Matalan annoksen tietokonetomografia johtaa suurempaan kohinaan kuvassa ja täten informaation katoamiseen. Erilaisia kohinanpoistometodeja käytetään pyrkiessä pienentämään kuvien kohinaa.

Tämän työn tarkoituksena oli tutkia, voitaisiinko tietokonetomografiakuvien viipaleiden kohinan välistä temporaalista informaatiota käyttää skannauksien kohinanpoistossa. Kolmiulotteisia konvoluutiotasoja käyttävä konvoluutioneuroverkko koulutettiin julkisesti saatavilla olevilla tietokonetomografiakuvilla. Kuviin oli asetettu keinotekoista kohinaa, simuloidakseen matalan annoksen tietokonetomografiakuvia. Toinen neuroverkko, jossa oli kaksiulotteisia konvoluutiotasoja, koulutettiin vertailua varten. Kohinanpoiston arvioimiseen mitattiin erilaisia metriikoita testidatasetistä saaduista tuloksista.

Tulokset osoittavat, että viipaleiden välisen temporaalisen informaation käyttäminen kolmiulotteisten konvoluutiotasojen avulla on erityisen hyvä todella matalan annoksen tietokonetomografiakuvien kohinanpoistossa. Eri metodeilla saatujen kohinanpoiston tulosten väliset erot olivat pienempiä, kun kohinan taso oli matalampi.

Avainsanat: matalan annoksen tietokonetomografia, sinogrammi, syväoppiminen, U-Net

TABLE OF CONTENTS

FOREWORD

This thesis is the culmination of my studies in artificial intelligence, but just the start of my journey in working professionally with machine learning. Visidon OY allowed me to get into the field and ignite the spark in me to work on these topics. My work at Visidon allowed me to expand my knowledge also on medical imaging by making the writing of this thesis possible.

I would like to thank Matteo Pedone for being my thesis supervisor. Matteo teached me a lot about research and the reality of it, during this period of writing this thesis. In addition, special thanks to Juho-Petteri Lesonen for not only being my technical instructor during the thesis project, but also for being my mentor during my first two years at Visidon. Lots of my practical skills of working with machine learning is thanks to him. Special thanks to the guys in room 4, for all the support, ideas, and conversations about neural networks and everything else.

Lastly, a huge thanks to my wife Pinja for all of the love, support, and faith in myself she has given me during my studies and the process of writing this thesis.

Oulu, June 10th, 2021

Saku Moilanen

LIST OF ABBREVIATIONS AND SYMBOLS

CT	computed tomography
3D	three-dimensional
2D	two-dimensional
HU	Hounsfield unit
FBP	filtered back-projection
mAs	X-ray tube time-current product
kVp	peak kilo voltage
Sv	sievert
Gy	gray
NLM	non-local means
BM3D	block-matching and 3D filtering
CNN	convolutional neural network
ANN	artifical neural network
GPU	graphical processing unit
CPU	ventral processing unit
FC	fully connected
ReLU	rectified linear unit
ResNet	residual network
SGD	stochastic gradient descent
API	application programming interface
DnCNN	denoising convolutional neural network
CNN DAE	convolutional denoising autoencoder
DICOM	Digital Imaging and Communications in Medicine
MSE	mean squared error
PSNR	peak signal-to-noise ratio
SSIM	structural similarity index

μ	linear attenuation coefficient
L_1	mean absolute error, L1-loss
L_2	mean squared error, L2-loss
L_P	perceptual loss
ϕ	loss network
σ	standard deviation
$\cup$	union
$\cap$	intersection

1. INTRODUCTION

Computed tomography (CT) is a medical imaging modality widely used for diagnostic purposes. Like traditional radiography, CT is also based on X-rays and their attenuation on different tissues. Traditional X-ray radiography provides only planar information of the subject's anatomy while CT provides also sectional information. This is achieved by rotating the X-ray tube and the detector around the subject and taking subsequent scans. The scans are computationally reconstructed into planar slices of the subject. The sectional information of the CT can be used in diagnosing and monitoring tumors in the body. Along with diagnosing and monitoring cancer, CT is also essential in planning of radiotherapy and its dose management. CT scan can also be used in diagnosing pneumonia from lung scans. Figure 1 visualizes a simplified CT image acquisition pipeline from scanning the patient with the X-ray tube and detector, to acquiring the grayscale CT image.

X-rays are a form of ionizing radiation. Ionizing radiation is capable of ionizing atoms or molecules of tissues, causing negative effects and hence the exposure of the radiation for the patient needs to be kept as low as possible. The energy of the X-rays dictates the effects of the radiation on the body. A higher energy results in higher radiation dose. The energy of the X-rays is dependant on the voltage and current of the X-ray tube used in creation of the X-rays. In the worst case scenario, the damage to DNA caused by ionizing radiation may lead to the induction of cancer.

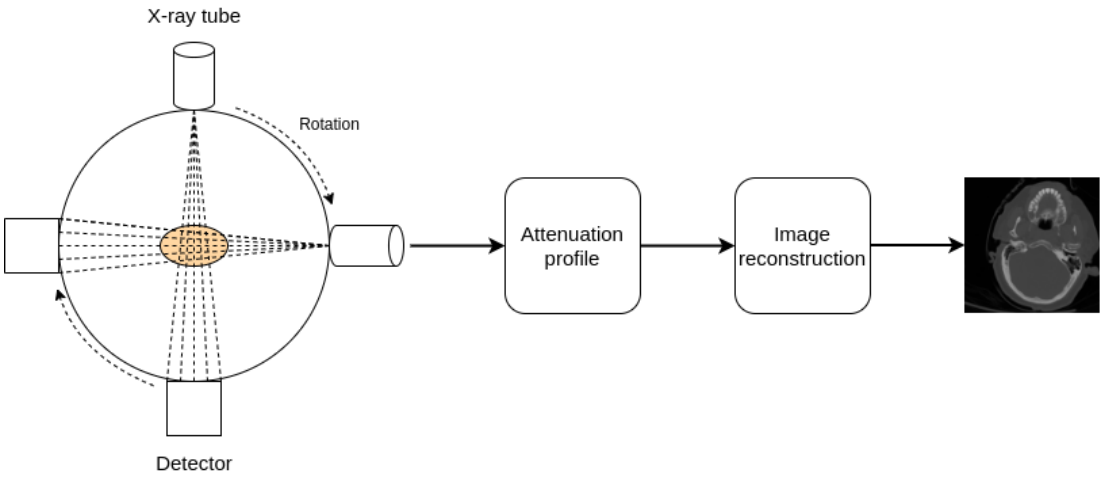


Figure 1. A simplified CT image acquisition pipeline.

To reduce the radiation dose of the patient, low-dose CT scans, where the X-ray tube current and voltage are set to lower values, are used. Using low-dose CT scans is especially critical for patients at higher risk of developing cancer. Low-dose CT scans suffer from increased noise. Image noise is unwanted random variation of pixel values in the image. The noise in CT scans has two sources. Electronic noise is caused by the electronic circuits of the machine itself. The variation of photons radiated also causes noise as the image quality is dependant on the number of photons arriving at the detector. Low-dose CT scans radiate lower number of photons than normal-dose CT and hence low-dose CT scan contains a larger amount of noise. Increased noise can make it more difficult for the physician to read the image and use it for diagnosis

as some of the information is lost. Denoising can be done on CT scans to reduce the noise and improve the image quality.

Several denoising algorithms exist, ranging from using smoothing filters to computing non-local averages of similar pixels in the image. The algorithm used depends on the acceptable trade-off between image detail and noise level. The details are important in medical imaging as they may contain information critical for diagnostic purposes. CT scan denoising can be done in three different phases: before, during, or after the reconstruction of the image.

Artificial intelligence has been a popular topic of research for decades now, while the increasing amount of computational power has made different applications possible. When the computer learns patterns from data, it is called machine learning. Machine learning applications can range from email spam filters to predictions of real estate values. Deep learning is a part of machine learning, where the deeper depth of the model allows for learning of more complex features. Deep learning can be used in the denoising of images, including CT scans.

This thesis is structured in the following way: chapter 2 shortly describes the use of deep learning in CT denoising and the motivation to this thesis. In chapter 3 we take a closer look into CT image acquisition, the noise in them, and the methods used to try and remove the noise. Chapter 4 describes the mechanisms and the structure of convolutional neural networks and the training process. In chapter 5 the denoising network of this thesis is introduced, along with the data used and the training process. In chapter 6 the results are presented and further discussed in chapter 7.

2. MOTIVATION

Deep learning is used in numerous different image processing tasks ranging from object detection to medical image segmentation. Deep neural networks are designed to follow a similar structure as natural neural connections in the human brain. The units of neural networks are artificial neurons. The neurons are connected to each other and transmit information between them. Each neuron computes their activation value which is the output of the neuron. The activations are calculated with an activation function.

Neurons are usually in a layered structure. The outputs of one layer are the inputs of the second layer. The neurons have varying states and weights which are adjusted during the training process of the network.

Neural networks can be used in the denoising of CT images. When image processing tasks are done, the neural networks utilize a mathematical operation called convolution. The non-linearity of the networks allows for learning complex noise models and producing noise-free estimates of noisy input images. The traditional denoising methods are not as capable of denoising CT scans, where the noise distribution is not uniform. Highly optimized neural networks should be capable of producing reduced noise estimates of CT scans faster than traditional denoising methods.

The temporal information of three-dimensional (3D) CT scans can be utilized with the use of 3D layers in the network. The theory is that the consecutive slices of the volume contain similar kind of noise information and utilizing this information could lead to a better denoising result. The tissues in CT scans are usually several slices thick, meaning there are structural similarities in neighbouring slices. It can be assumed that the characteristics of noise in tissue are similar in all slices containing the tissue. This information is not utilized at all when just the individual slices are denoised with 2D denoising methods. Training of denoising neural networks are done on noisy input- and noise-free output image pairs.

The denoising of CT images by deep learning has already received attention in the research field. Chen *et al.* [1] was one of the first researches using deep convolutional networks to denoise reconstructed CT images. Yi *et al.* [2] used a generative adversarial network, originally introduced by Goodfellow *et al.* in [3], where two networks compete. A generator network generates samples and tries to fool the discriminator network. Yang *et al.* [4] and You *et al.* [5] also used generative adversarial networks with a combination of per-pixel loss and perceptual loss. Perceptual loss is used to compute the difference of images using a separate pretrained feed-forward network. The latter research utilized the 3D volumetric information. Ghani *et al.* [6] researched if CT scan could be denoised in the sinogram domain before the reconstruction of the scan.

In this thesis, I introduce a 3D convolutional neural network capable of producing noise reduced estimates from volumes of low-dose CT images. The introduced network has several dense connections between the layers. The output of the network combines feature maps from several different stages of the network by averaging them. The goal is to have a neural network model capable of producing noise-free estimates of low-dose CT volumes with complex noise distribution while preserving the details of the image.

3. NOISE AND DENOISING

3.1. Computed Tomography

CT was first used in clinical use in the early 1970s. Nowadays it is widely used in radiology [7]. CT image is done by rotating X-ray tube and detector. The combination of the X-ray tube and the detector is called a gantry. The X-ray beams are narrow and the measurements are done in several different directions by rotating the gantry within a desired image plane around the patient. In some of the newer CT systems, only the source rotates around the subject, but this technique is not often used because of the higher system costs.

The principle of CT is based on the attenuation of X-rays in the object. The intensity of the X-rays is reduced as photons traverse the object. The photons can be either absorbed into the matter or be deflected by scattering. A quantitative measurement of the beams interaction with the subject can be acquired by measuring the X-ray beam intensity after it has passed through the object. The detector records the attenuated X-ray beams. The amount of attenuation is dependant on the attenuating material and its thickness. Each material has its own linear attenuation coefficient μ which describes the fraction of the beam that is absorbed or scattered per unit thickness of the subject.

Intensity I_x of the X-ray beam at depth x after interacting with the subject follows exponential decay and is related to the material according to Beer-Lambert's law. I_x can be calculated with

$$I_x = I_0 e^{-\mu x} \tag{1}$$

where I_0 is the original intensity of the X-ray beam. From Equation (1) we get the equation for calculating μ:

$$\mu = \frac{-\ln \frac{I_x}{I_0}}{x} \tag{2}$$

Usually the thickness is measured in cm so the unit of μ is cm^{-1}.

The acquired attenuation signal S is the sum of μ of different tissues. Summing of different attenuation coefficients is visualized in Figure 2. Attenuation coefficients μ are acquired by mathematical reconstruction.

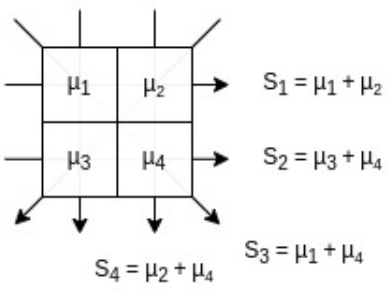
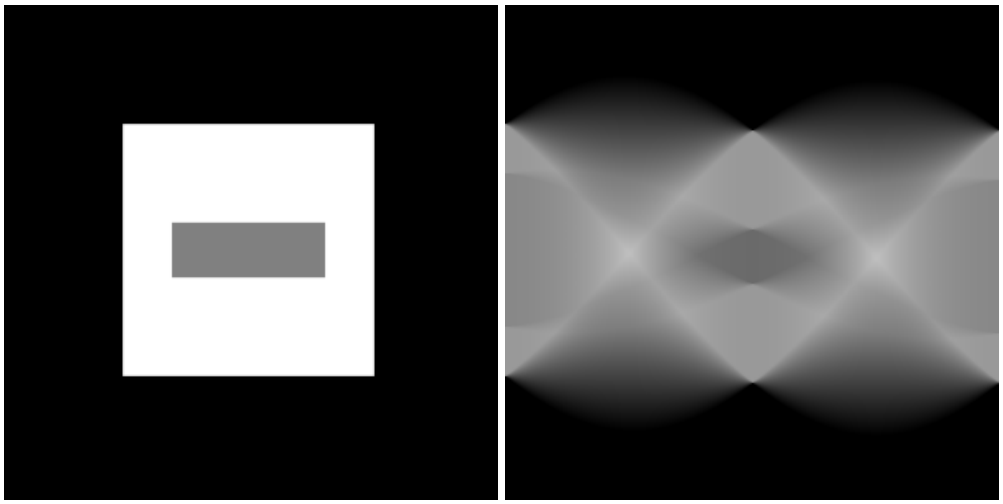


Figure 2. The attenuation signals $S_{i,j}$ acquired by the detector are the sum of attenuation coefficients $\mu_{i,j}$ of different tissues.

The logarithm of the attenuated intensity along a beam path is called the line-integral of the patient's attenuation coefficients. Multiple line integrals are acquired by rotating the gantry around the patient. Combination of line integrals fully around the object within a single angular position is a single projection. The combination is also referred to as the sinogram. The sinogram is also the result of the Radon transform of the of the object. Figure 3 visualizes a simple image and the sinogram acquired from it with the Radon transform. The transform is named after Johann Radon who introduced the transform in 1917 [8]. The sinogram can be transformed back into image space with the inverse Radon transform [9]. Modern CT scanners are able to take about 1000 projections in a full 360° rotation around the patient. Each projection comprises of up to 900 individual line integrals. [7]



(a) (b)

Figure 3. A simple box shape (a) and the sinogram acquired by Radon transform (b).

The CT images are usually viewed on the Hounsfield scale. On the Hounsfield scale, water has a value of 0 Hounsfield units (HU). Tissues denser than water have positive values and tissues less dense than water have negative values. The HU values for tissues are calculated with

$$HU = 1000 \times \frac{(\mu_{tissue} - \mu_{H_2O})}{\mu_{H_2O}} \tag{3}$$

where μ is the linear attenuation coefficient of the tissue. The HUs for different tissues are listed in Table 1.

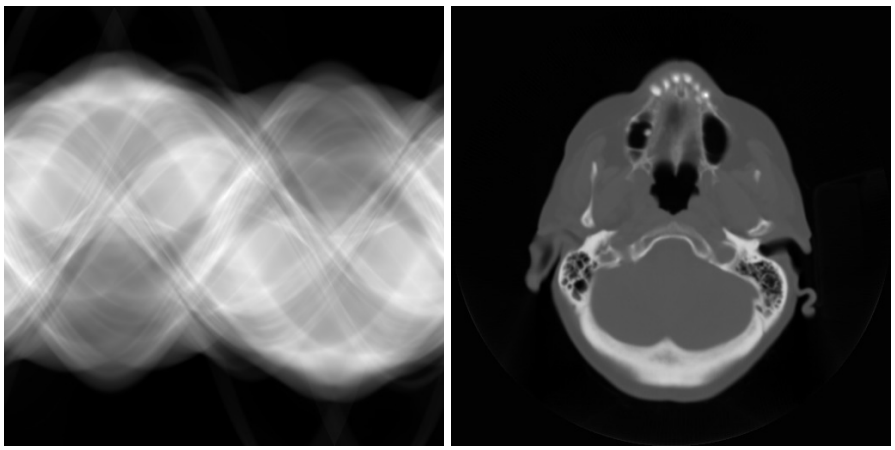
When CT image is shown on a computer monitor, the HUs correspond to the grayscale values of individual pixels. The HUs may be converted to 10-bit or 8-bit grayscale values, depending on the software and hardware used to visualize the images. The greyscale components of the CT image can be modified by using windowing. Window width dictates what range of values are visible in the image. If an examination with many different tissues of interest is done, a wider window is used. A narrower window width is utilized when the examination area consists of tissues with similar attenuation.

Table 1. Hounsfield units (HU) for different tissues.

Tissue	**HU**
Air	-1000
Lung	-500
Fat	-100 to -50
Water	0
Kidney	30
Muscle	10 to 40
Grey matter	37 to 45
White matter	20 to 30
Liver	40 to 60
Soft tissue	100 to 300
Cancellous bone	1000
Dense bone	3000

3.2. CT Reconstruction

The established analytical CT image reconstruction algorithm is the filtered back-projection (FBP) [10]. FBP is a robust and fast algorithm that is capable of generating CT studies of adequate image quality. The principle of reconstruction with FBP is the inversion of the line-integrals. The intensity profiles acquired by the detector are preprocessed into projection data. Projection data is low-pass filtered to compensate for the blur resulting from the different number of projections that pass the center and the edges of the subject. The filtering kernel can be chosen to modify the image properties. The choice of the filtering kernel depends on the region of interest. Apart from low-pass filtering, edge enhancing and smoothing filtering is done. The resulting image noise is dependant on the image sharpness: The sharper the image, the noisier the result. A sinogram and its reconstruction is visualized in Figure 4.



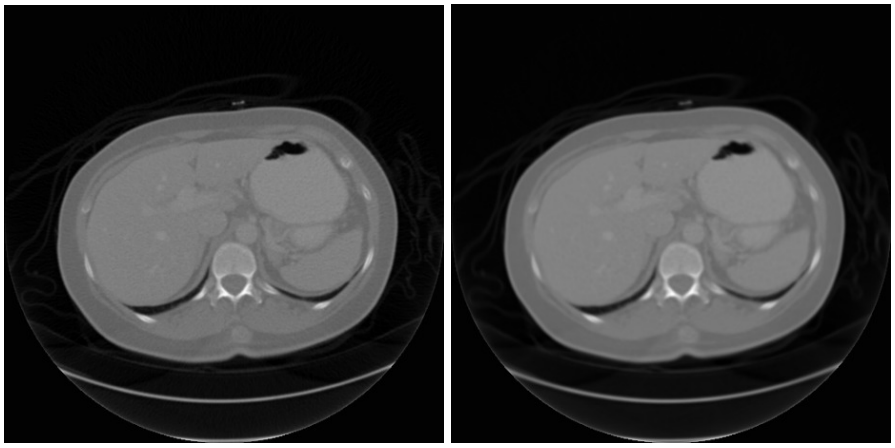
(a) (b)

Figure 4. Sinogram (a) and the CT reconstructed (b) by using back propagation.

FBP is unable to account for the noise that results from variations of photon numbers across the image plane. Using smoothing filtering results in lower noise but also tends to blurs the edges and details, which are important in CT imaging and need to be preserved. Figure 5 illustrates the result of using a smoothing filter on sinogram.

Iterative image reconstruction algorithms are a more sophisticated CT image reconstruction method. An image example is generated from the measured projections. The first example is compared with an image generated by simulated projection data. The image estimate and the projection data are updated if the results are different. The process is repeated until a predefined condition is satisfied. Iterative image algorithms are computationally demanding and are reported to not have sufficient results for diagnostic purposes [11, 12]. The images acquired with iterative reconstruction tend to be blurry and too smooth. Older generations of iterative reconstruction methods are also shown to result in reduced spatial resolution when compared to FBP methods [13], but newer more sophisticated methods show better results [14].

Deep learning based reconstruction methods are also used in CT image reconstruction. The deep learning based methods use convolutional neural networks and can achieve better diagnostic image quality than iterative reconstruction methods [15, 16, 17]. The problem with deep learning based methods is the amount of data available to train the models. [18]



(a) (b)

Figure 5. The result of processing the sinogram with a normalized box filter. The resulting image (b) has less noise but also a poorer spatial resolution than unprocessed image (a).

3.3. Low-Dose CT

Low-dose CT scan is especially crucial for patients at higher risk of developing cancer. National Lung Screening Trial Research team demonstrate in their work [19], that using low-dose CT scans on high-risk patients lowers lung cancer mortality. Normal dose CT scan tube parameters depend on the body part of interest and the size of the patient. The patient dosage is proportional on tube current-time product (mAs), tube peak kilo voltage (kVp), and patient centering. Usual full dose chest scan parameters

are around 100 to 120 kVp and 300 to 330 mAs, but can vary. It is stated in [20] that the radiation dose changes with the square of kVp. For example, a reduction of kVp from 120 to 100 lowers the patient dose by 33 %. Low-dose CT scans can have the tube current set at as low as 20 mAs. There is no clear consensus on which dose level is considered as low-dose [21].

The radiation dose is calculated in sieverts (Sv). Tissues are considered to have different tissue weighting factors w_T as some tissues are more sensitive to radiation than others. Weighting factors are used when calculating the effective dose E in tissues T by radiation R by using the equation

$$E = \sum_T w_T \sum_R w_R D_{T,R} \tag{4}$$

where w_R is the radiation weighting factor of R and $D_{T,R}$ is the absorbed dose in grays (Gy) in T by R. The International Commission on Radiological Protection [22] sets the tissue weighting factors as listed in Table 2. The approximate effective dose of a normal chest CT scan is 8 mSv. Low-dose chest CT scan effective dose ranges from 0.2 to 3.5 mSv [19, 23, 24, 25, 26].

Table 2. Tissue weighting factors w_T according to [22 p. 65].

Tissue	w_T
Stomach, breast, lung, colon, red bone-marrow, remainder tissues[1]	0.12
Gonads	0.08
Bladder, liver, thyroid, oesophagus	0.04
Bone surface, skin, brain, salivary glands	0.01

The major downside of low-dose CT scans is the introduced noise in the images. The number of photons vary across the image plane. This variation is Poisson statistical variation. The variation of photon emission from the X-ray source is a Poisson process. The probability P_k of emitting photons k in an time interval of the process in an time interval can be calculated by

$$P_k(N_0) = \frac{(N_0^k e^{-N_0})}{k!} \tag{5}$$

where N_0 is the average number of photons emitted during the said interval [27 p.21]. Reduction in the radiation dose leads to a lower number of photons being radiated. This leads to Poisson noise. CT images also contain electronic noise originating from the analogical circuits of the system itself. Electronic noise is independent of the number of photons radiated or detected. Electronic noise can be neglected in full-dose CT scans, as its effect is only minor. However, on low-dose CT scans the impact of electronic noise becomes larger and cannot be neglected anymore.

[1]Remainder tissues consists of: Adrenals, gall bladder, heart, kidneys, muscle, lymphatic nodes, prostate, uterus, spleen, small intestine, thymus, extrathoracic region, oral mucosa

3.4. Denoising

The denoising of CT can be classified into three categories based on which part of the process and how the denoising is achieved [28]: sinogram pre-processing, post-processing of reconstructed images, and iterative reconstruction methods.

Sinogram pre-processing methods perform the denoising on the sinograms before the image reconstruction process. Some common methods are sinogram smoothing, penalized weighted least-squares, local average filtering and convolutional masks. In sinogram smoothing, a likelihood method is derived to smoothen the sinogram [29]. The smoothened sinogram contains less noise. Penalized weighted least-squares is a cost function used in modeling the noise properties in either the sinogram space or image domain [30]. Minimizing this cost function while doing image reconstruction can lead to reduced noise in the final CT. Local average filtering is replacing a pixel value with the average value of the neighbour pixels. Denoising with convolutional masks is a similar technique, but uses different kinds of convolutional kernels. These methods blur the image details and edges.

Iterative reconstruction techniques use prior information on the noisy images to achieve the denoising task. Non-local means (NLM) [31] is one method requiring prior information of the noise standard deviation before it can be applied. NLM differs from local averaging by calculating the mean of similar pixels in the image. The newer generations of iterative image reconstruction algorithms are capable of denoising, reducing artifacts, and improving spatial resolution [14].

Post-processing techniques apply straight to the reconstructed CT images. The advantage of working on the image domain is the availability of images. Sinogram data is not as widely available as the reconstructed CT data. Iterative reconstruction methods' requirement of prior information can also be challenging to have access to. Neural network -based methods are popular and require large amounts of data. Sagheer and George state in their work [28] that different slices of CT image have correlations and that the frames' intensities are almost similar to the neighbouring frames. Adjacent slices contain structural information and are not independent from each other. The 3D information is also crucial for the radiologist to utilize when examining CT images.

NLM, block-matching and 3D filtering (BM3D) [32], and median filtering are some of the most popular denoising methods with BM3D and NLM considered to be state-of-the-art in image denoising. NLM measures the similarity of pixels in the image and computes the mean of the similar pixels to achieve denoising. NLM results in lesser loss of image detail than local mean filtering. BM3D groups similar 2D blocks of the image into clusters in 3D data arrays by block-matching. Collaborative filtering is done on fragment groups. Collaborative filtering reveals fine details that are shared by the grouped blocks. Transform-domain shrinkage is followed by a linear transform to reproduce all fragments which are returned to their original positions to achieve the denoised image. Median filtering computes the median of the neighbouring pixels to calculate a new value for the for the pixel.

Convolutional neural networks (CNN) are used in image denoising problems in both sinogram space and image domain. [33]

4. CONVOLUTIONAL NEURAL NETWORKS

CNNs are a type of artificial neural networks (ANN). ANNs are a part of deep learning. Deep learning is a subclass of machine learning which in turn is a part of artificial intelligence. Any technique that is designed to mimic human behaviour can be considered artificial intelligence. Machine learning is defined by Mitchell in his work Machine Learning (1997) as following: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [34 p.2].

CNNs are commonly used in many tasks ranging from image recognition to video super-resolution. In healthcare, CNNs can be used for example in automatic osteoarthritis evaluation [35], gray matter age prediction [36], or bone age prediction [37]. This chapter takes a closer look into the different building blocks of CNNs, the training of a CNN model, and the data needed for it.

Neural networks

ANNs are inspired by biological neurons and their vast networks present in a human brain. The output of artificial neurons is based on the input of the neuron. Biological neurons have the same kind of behaviour. Neural network learns the features of the input and returns an output based on these features. The neurons are in a layered structure, where the neurons between different layers are connected to each other. Each layer has hundreds to millions of learnable parameters. In a layered CNN format, the outputs of the previous layer are fed as the input to the next layer. The connections can also skip between layers and be fed into layers further in the network. The layers between the input and output layers are called hidden layers. Mathematically, the principle of neural networks can be defined as the learning of the function f when the input x and output y are known:

$$f(x) = y \tag{6}$$

CNNs utilize convolutions in the network architecture. Convolution is a mathematical operation in which each element of the image is added to its local neighbors. The values are weighted by the kernel. The convolutional kernel's parameters are learnable. By using convolutions, the CNNs are capable of processing data that is array-like. The breakthrough for CNNs came in 2004 when graphical processing units (GPU) were utilized in the computation of CNNs [38]. GPUs are able to process multiple computations simultaneously as they have a large number of cores. GPUs have a larger memory bandwidth than central processing units (CPU) meaning that huge amounts of data are processed more efficiently. GPUs are optimized for training deep learning models.

The training procedure is done with known input-output pairs. The inputs can be images, word vectors, or numerical vectors. The output of the network can be a label in a classifying task or a predicted value in a regression task. During the training of the network, the parameters of the network are trained according to the input fed to

the network and the output of that. With help of an objective function, the output is evaluated and the parameters of network's layers are adjusted accordingly.

4.1. Structure

Basic building blocks of CNNs are the convolutional layers, activation units, and pooling layers. Figure 6 illustrates the usual use of these layers. The first layers of the CNN see the higher level features of the input. They are larger shapes such as the head or torso when anatomical images are considered. More abstract features are obtained when the input propagates towards the deeper layers of the CNN. Activation units are used to scale the linear activations of convolutional layers into nonlinear activation maps. Pooling layers are used to downsample the input of the layer by usually taking the maximum or average value of a certain window and setting it as the new value for the patch. CNNs are often used in image recognition tasks where the input is classified into some predefined class.

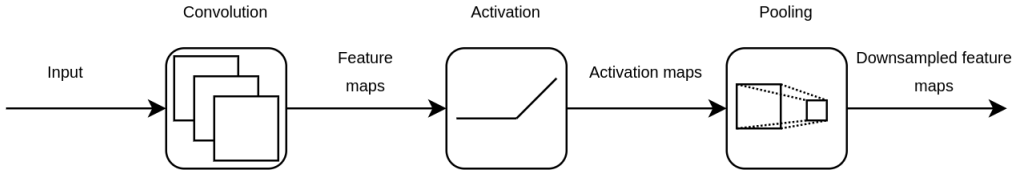


Figure 6. The basic building block for convolutional neural network. Linear feature maps of input are calculated with convolutional layers and scaled into nonlinear activation maps with activation layer. Pooling operation is used to downsample the activation maps.

A transposed convolution layer can be used to perform upsampling. Upsampling could also be achieved by performing interpolation, where the upsampling method is predefined. By using transposed convolution layers, the neural network is allowed to learn the upsampling transformation by itself.

Other popular CNN layers are the fully connected (FC) layer and the softmax layer. In classifying networks, a FC layer followed by softmax activation layer is used to return the decimal probabilities of the input belonging to predefined classes. FC or softmax layers are not used in the networks used in this thesis so they will not be presented closer here.

Before a closer look into different layers, let us define some terminology. $Kernel$ is a single matrix with learnable weights that is used in the convolution operation. The dimensions of the kernel define the name for the convolution. If a 2D kernel is used in the convolution operation, the operation is referred to as $2D-convolution$. $Filter$ is a concatenation of multiple kernels. $Stride$ defines how a convolution or pooling moves around the input. The value of stride defines how many units the window shifts at a time. $Encoder$ is the analysis part while $decoder$ is the synthesis part of an encoder-decoder network [39]. The $gradient$ is a measure of how much the output of the network would change for an update of the parameter in question.

4.1.1. Convolutional Layer

Convolution is the result of sliding a kernel across the input signal and computing the dot product between the kernel and the signal. In 2D-CNNs, the input is an image. The input is a 3D-volume in this thesis' work. The convolutional layers in CNNs have several kernels that extend the full depth of the input. The kernels all have learnable parameters (or weights and biases).

The usual neural network implementation of a convolutional layer is actually cross-correlation [40 p.329]. In convolution operation, the kernel is flipped relative to the input. This flip is done to achieve commutative property in the operation. The kernel is not flipped in neural network implementations as is usually done in convolution. The flipping is not needed as the kernel weights are learned during the training phase of the network. From now on, the term *convolution* refers to the neural network implementation of the convolutional layer. The equation for the convolution S between 2D-image I and kernel K is presented in Equation (7) [40 p.329] and visualized in Figure 7. In Equation (7), i and j correspond to rows and columns of the image I. Convolutional layers can be set to use different strides for the convolution. The size of the output feature maps can be made smaller by making the convolutional kernel shift more.

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{7}$$

The convolution operation can also be visualized in a matrix multiplication form, where the convolution kernel is represented as a Toeplitz matrix. A Toeplitz matrix has diagonal elements as constants. Let us define an example. A regular 3×3 convolutional kernel w is presented as a Toeplitz matrix $\boldsymbol{C}$. The nonzero elements are the $w_{i,j}$ of the kernel with i and j being the row and column of the kernel, respectively:

$$\boldsymbol{C} = \begin{pmatrix} w_{0,1} & w_{0,2} & w_{0,3} & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,1} & w_{0,2} & w_{0,3} & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & w_{0,1} & w_{0,2} & w_{0,3} & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{0,1} & w_{0,2} & w_{0,3} & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & 0 \end{pmatrix}$$

Define the input image $\boldsymbol{I}$ to be of shape 4×4 and flattened into a vector $\boldsymbol{X}$ with a shape of 16×1:

$\mathbf{X} = \big(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}\big)^{\mathrm{T}}$

When doing the matrix multiplication of the kernel $\boldsymbol{C}$ and image $\boldsymbol{X}$ we get

$$\mathbf{CX} = \mathbf{Y} \tag{8}$$

where $\boldsymbol{Y}$ is the output vector with a shape of 4×1: $\mathbf{Y} = \big(y_0, y_1, y_2, y_3,\big)^{\mathrm{T}}$ which can be resized into an image with the shape of 2×2.

The kernel sizes are relatively small. The sizes are usually 3×3 or 5×5 pixels. Having smaller kernels than the input leads to the networks having sparse interactions (sparse weights). This is opposed to traditional ANNs where FC layers are used. FC layers' every output unit is connected to every input unit of the next layer. Sparse weights allow for the model to need fewer parameters, leading to reduced memory requirements. Statistical efficiency is also increased. The input images could have

millions of pixels. Smaller features can be detected with kernels occupying just tens or hundreds of pixels. [40 p.330]

The output of the convolution operation is smaller than the input array, as visualized in Figure 7. This is because the operation is done only on valid pixels. If padding is introduced to the input array, the output of the convolution would have the same dimensions as the input. With zero padding the image is extended by a layer of zeros.

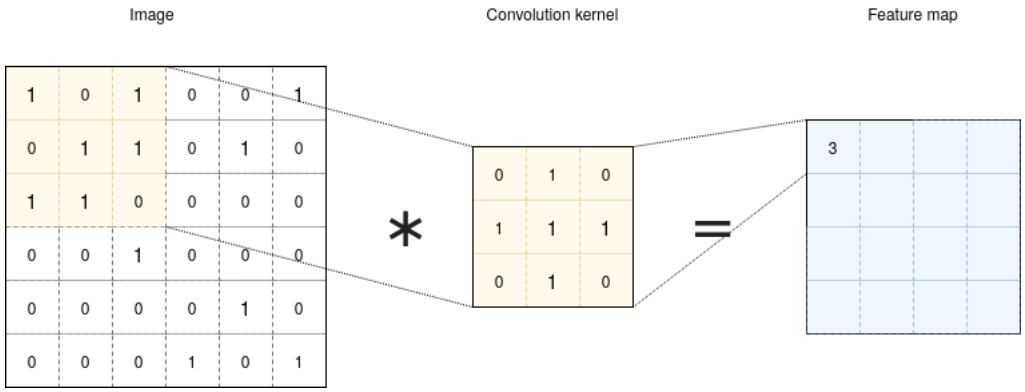


Figure 7. The convolution operation in 2D array. The array has dimensions of 6×6 while the kernel has 3×3. The convolution kernel is set to have a stride of 1. The output feature map has dimensions of 4×4.

3D-convolution layers utilize 3D-kernels. The kernel is able to move in height, width, and depth. The depth dimension contains the slices of the CT scan. The 3D convolution is visualized in Figure 8.

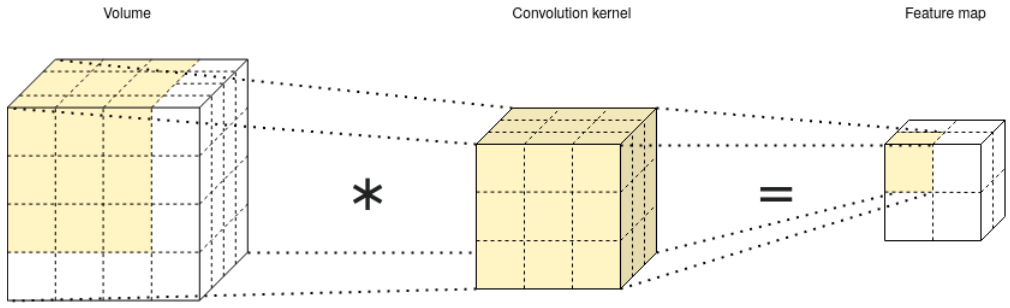


Figure 8. The convolution operation in 3D array.

4.1.2. Transposed Convolutional Layer

The transposed convolutional layer is often used when a transformation in the opposite direction of the normal convolution is needed. Transposed convolutional layer is often referred to as $deconvolution$. Transposed convolutional layer enables decoders for recovering the input images from feature maps. [41, 42]

Let us define the transposed convolutional kernel as the transpose of $\boldsymbol{C}$ used in the example in 4.1.1. Output $\boldsymbol{Y}$ of the said example is the input for the transposed

convolution. $\boldsymbol{C}^{\mathrm{T}}$ has a shape of 16×4 and the $\boldsymbol{Y}$ a shape of 4×1. Now Equation (8) takes the form of

$$\boldsymbol{C}^{\mathrm{T}} \cdot \boldsymbol{Y} = \boldsymbol{X}_2 \tag{9}$$

where the result is $\boldsymbol{X}_2$ which is a vector with a shape of 16×1. $\boldsymbol{X}_2$ can be then transformed into an image $\boldsymbol{I}_2$ with the shape of 4×4. Figure 9 visualizes the operation on an zero padded input.

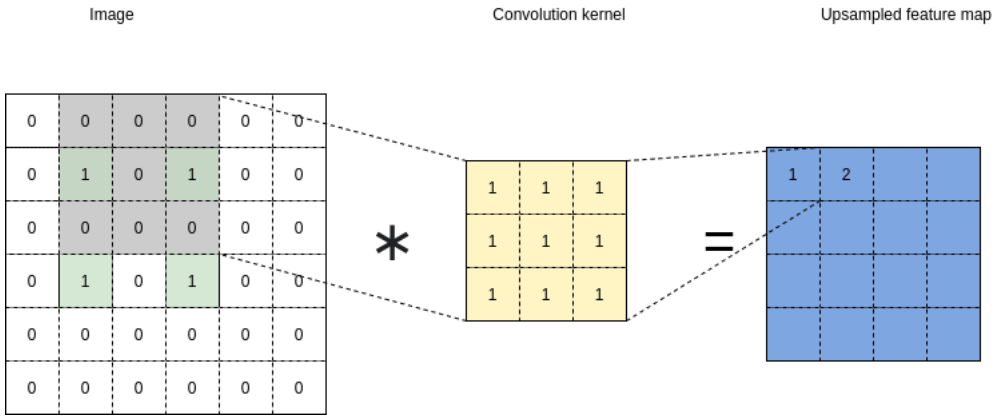


Figure 9. Transposed convolution of 2D image. The 2×2 image is zero-padded to size of 6×6 with a stride of 2. The padding is asymmetrical. A transposed convolution kernel of size 3×3 is used. The output feature maps of the transposed convolution has a size of 4×4. Notice that the first pixel of the output feature map has already been calculated to have a value of 1.

4.1.3. Activation Layer

Activation layers in CNNs are used to scale the linear feature maps of the convolutional layers into nonlinear activation maps. Without activation layers, the linear neural network would not be able to perform on more complex cases. Some of the most common activation functions are *sigmoid*, *tanh*, and Rectified Linear Unit (*ReLU*). The graphs of *sigmoid*, *tanh*, and *ReLU* are visualized in Figure 10. *ReLU* maps the feature maps of the convolution following Equation (10) :

$$f(x) = max(0, x) \tag{10}$$

ReLU has taken its place as the most popular activation unit. The function of *ReLU* (Equation (10)) has a simple definition where the negative values of feature maps are set to zero. The positive values have constant gradient. *sigmoid* and *tanh* suffer from a vanishing gradient problem. This happens as the networks get deeper and gradients get extremely small. *ReLU* does not suffer from this because of the constant positive gradient. [43]

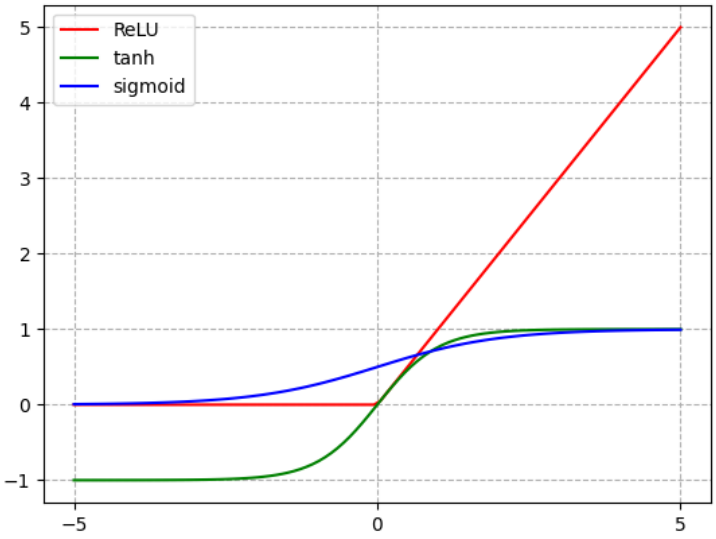


Figure 10. Different activation functions and their graphs. Figure modified from [43].

4.1.4. Pooling Layer

Pooling layers are usually implemented after activation layers. Pooling layers compute a single value from a specific window from the neighboring pixels of the layer input. This acts as a summary statistic of the nearby values. This operation allows the network to be more robust to small translations of the input. The values of the pooled outputs do not change even if the input of the network is slightly translated. Pooling layers also downscale the activation maps. This makes the following computations of the network less computationally demanding.

Max-pooling layer [44] selects the maximum value of the pooling area and passes this value forward. Figure 11 visualizes using max-pooling to pool an activation map of size 4×4 with an pooling window of 2×2 and stride of 2. The pooled output is of size 2×2. Average pooling is also a common pooling layer. In average pooling, the average of the pooling window is passed forward.

4.1.5. Skip Connections

Skip connections are sometimes utilized in deeper network architectures. Skip connections are used to feed the gradients to the deeper layers of the network. The gradients are also feeded as an input to the next layer. Skip connections tackle the problem of vanishing gradient. The skip connections also allow for layers deeper in the network to learn the simpler features learned in the earlier layers. The gradients are either added or concatenated together before fed into the next layer of the network. Skip connections are most notably used in Residual Networks (ResNet) [45] and in U-Nets [46].

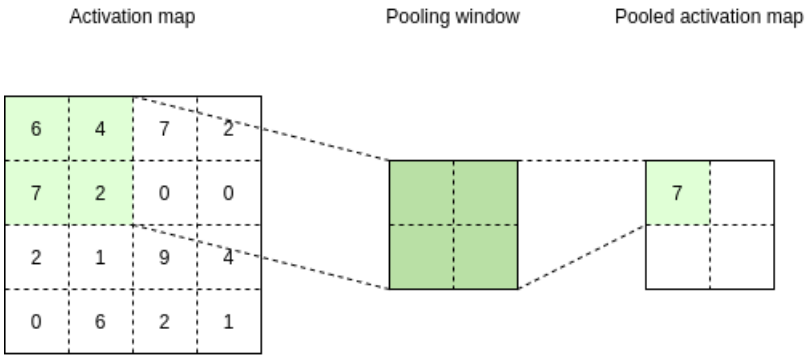


Figure 11. Max-pooling of activation map with a pooling window of size 2×2 and stride 2.

Utilizing skip connections helps the network converge faster when training [47]. Faster converge allows for the use of deeper networks.

The network architecture introduced in this thesis utilizes the skip connections the same way as in [46]. The U-Net architecture is visualized[2] in Figure 12. The U-Net is a variation of an encoder-decoder network, with added skip connections [49]. In an U-Net -based architecture, the gradients are concatenated.

4.2. Training

Process of training CNNs is based on the feedforward and back propagation phases. In the feedforward phase the input of the network is fed through every layer of the network and an output is received. An objective function is used to determine how well the network has approximated the function f as according to Equation (6). The parameters of the network model are then updated to minimize the result of the objective function. Updating parameters are done via back propagation [50]. The back propagation algorithm computes an error gradient for each of the parameters and updates them accordingly. Parameters are updated on the direction of negative gradient. The gradients are calculated from the output of the network all way to the input. The back propagation is done using the chain rule. The gradients of the later layers are used to calculate the gradients in earlier layers. The goal is to find the parameters where the error is at the global minimum.

CNNs utilize different optimization algorithms for gradient descent. Stochastic gradient descent (SGD) has been the core optimization function for neural networks. SGD updates the weights based on a set learning rate, which dictates on how much the parameters can be changed during a single back propagation. Having too small of a learning rate leads to the network converging a lot slower. Too high of a learning rate allows the parameter updates to overshoot the minimum. Adam [51] has become a popular option for optimization and is shown to outperform several other optimization algorithms [52]. Adam utilizes individual adaptive learning rates for each of the network parameters.

[2]The graph visualization is done with PlotNeuralNet [48]

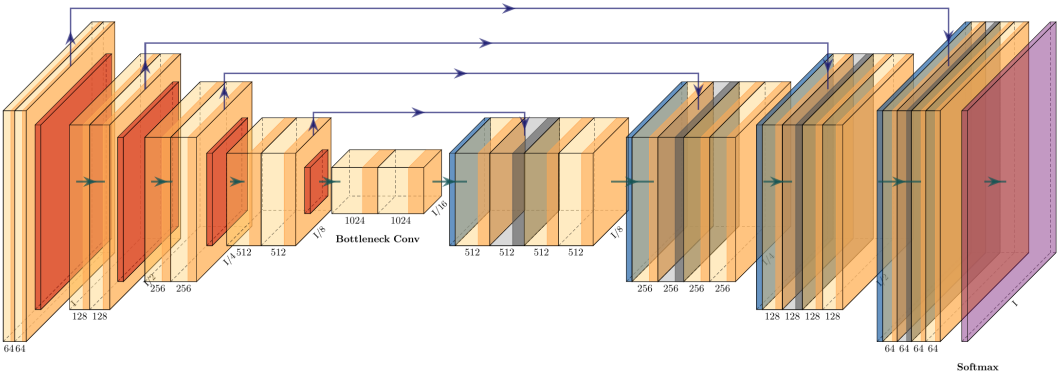


Figure 12. The architecture of the U-Net [46]. The skip connections are depicted as dark blue lines between the layers in the encoder (left) and decoder phases (right). This architecture has convolutional layers (light yellow) using *ReLU* activations (dark yellow), max-pooling layers (orange), and transposed convolution layers (light blue). The last layer of the network is a softmax layer (violet). I denotes the input and output image shape.

Usually the inputs of the network are normalized before fed to the network. The normalization is done to ensure that all inputs are in the same range and have a similar distribution. Normalization allows for the network to converge faster during training. Typical normalizations are ranges from 0 to 1 or -1 to 1.

Objective function is a function the optimization algorithm is trying to either minimize or maximize. The optimizer is searching for network parameters which have either the lower or highest score of the objective function. In CNNs, the objective function is typically an error function and usually referred to as a loss function. The optimizer is trying to minimize the loss function. The type of object function used in the training depends on the problem type. In classifying problems, cross-entropy object function is a popular choice. Mean absolute error (L_1) and mean squared error (L_2) are the usual choices in regression problems. L_1 loss is the average of the sum of absolute differences between the prediction y and the ground truth $\hat{y}$:

$$L_1 = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{11}$$

where n is the number of pixels when images or voxels when volumes are used. L_1 is more robust to outliers than L_2 loss. L_2 loss is the average of squared difference between y and $\hat{y}$ and calculated with

$$L_2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{12}$$

Both L_1 and L_2 losses are per-pixel-losses. They only compare each pixel of the output with the corresponding pixel of the ground truth. They are not able to capture any $perceptual$ differences between the images. Using only a per-pixel loss can lead to blurred outputs when doing a denoising task with the network model.

Perceptual loss was originally used in a style transfer network by Johnson, Alahi and Fei-Fei [53]. The aim of the perceptual loss function is to different high-level

features between the images. The features are extracted using a pretrained CNN. The loss network ϕ is a pretrained classification network trained to encode semantic and perceptual information. This information can be used in the perceptual loss function. In [53] the loss network was a pretrained VGG-16 network [54] which was pretrained on the ImageNet dataset [55]. Two perceptual loss functions were introduced: the feature reconstruction loss and the style transfer loss. The activations $\phi_j(x)$ of the jth layer of ϕ are used in the calculation of the loss rather than the output of the loss network to calculate the feature reconstruction loss. Let y be the target image and $\hat{y}$ be the output image. The convolutional layer $\phi_j(x)$ having a feature map of shape $C_j \times H_j \times W_j$. The feature reconstruction loss $l^{\phi,j}_{feat}$ is calculated with

$$l^{\phi,j}_{feat}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2 \tag{13}$$

When compared with Equation (12) it can be seen that L_2 loss is calculated from the extracted features.

Using perceptual loss leads to increased computational need in the training phase of the model but the run-time is real-time.

4.3. Data

Training a CNN requires a vast amount of data for the network to be generalizing. The model should be able to do the task with data the model has not seen earlier. The network model is overfitting if the model achieves near perfect results on the training data but poor results on some test data. The available data should be divided into three subsets: training, validation, and test data. An example ratio for the subsets could be 70 %, 20 %, and 10 %, respectively. The model is trained with the training data. The performance of the model is evaluated by using the evaluation data after each iteration (epoch) of the training data. The models final performance is tested on the test data only after training the model.

4.4. Denoising with Convolutional Neural Networks

CNNs can be used in image denoising problems. Zhang *et al.* [56] first introduced the denoising convolutional neural network (DnCNN). DnCNN has a deep architecture consisting of convolutional layers, *ReLU* -layers, and batch normalization [57] layers. Batch normalization layers normalize the input by re-centering and re-scaling it. Batch normalization allows for the faster training of the network. DnCNN predicts the difference of the noisy and ground truth image. This is different than the model proposed in this thesis which outputs the denoised image. The first convolutional layer followed by *ReLU* maps the input to 64 feature maps. After the initial layers, 18 blocks of convolutional layer followed by batch normalization layer and *ReLU* layer are used. A convolutional layer with a single filter is used to reconstruct the output. All convolutional layers in this network have 3×3 kernel size.

Gondara [58] introduced a rather simple convolutional denoising autoencoder (CNN DAE) for denoising of medical images. The architecture consisted of five convolutional layers. Max-pooling followed the first two convolutional layers, downscaling the feature maps. The feature maps of the following two convolutional layers were upsampled. The output of the network was the output of the last convolutional layer with a single filter. The original network was trained on dental X-rays and mammograms.

The work of this thesis aims to use the temporal information of CT images in the denoising task by using a 3D convolutional neural network.

5. METHODOLOGY

5.1. Denoising Network

The CNN model proposed in this thesis is based on the image segmentation network proposed on [49]. The architecture is visualized in Figure 13. The network architecture is more advanced version of [46], but with dense skip pathways. The use of dense connections allows for the latter layers in the network to utilize the feature maps acquired from earlier layers. The idea is that the network's optimizer would have an easier task of learning when the feature maps of the encoder and decoder are similar. The finer details of the CT scans that are essential for diagnostic purposes should also be preserved better. Apart from the skip connections, also the average of the layers in the first level of the network is computed and fed as the input to the output layer. All feature maps in the first level of the network are full resolution, so averaging the feature maps even further utilizes the dense connections. By using batch normalization layers in the blocks, the training of the network should be faster and more stable. The denoising network could be seen as a combination of four U-Net networks with different depths. The denoising model had a total of 7.7 million trainable parameters.

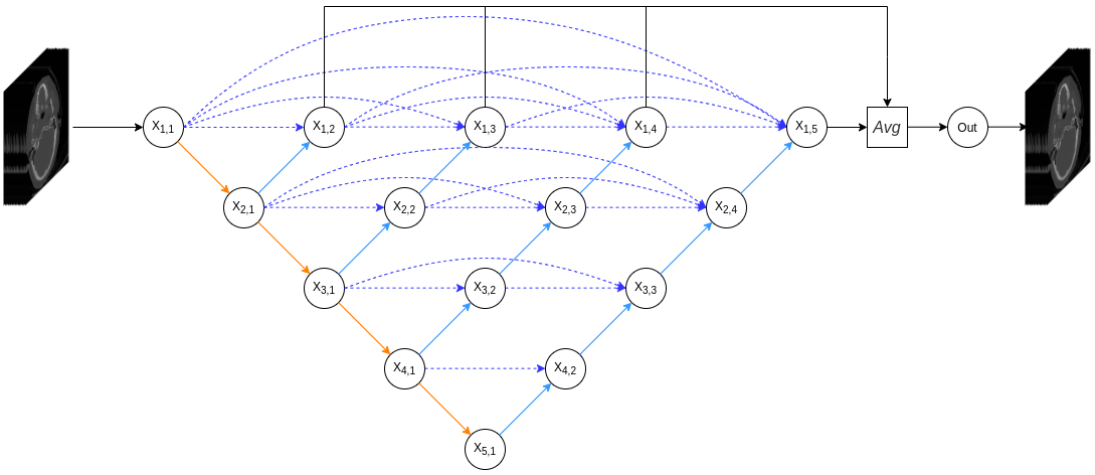


Figure 13. The architecture for the denoising convolutional neural network. The input is a volume of gray-scale CT scans with dimensions of $512 \times 512 \times 16$. The circles depict two blocks of 3D convolutional-, batch normalization-, and $ReLU$ -layers. The model has a depth of 5. The convolutional layers in depth 1 have 16 filters. Layers on subsequent depths have double the filters as compared to layers on previous depth. Convolutional layers in block $X_{1,1}$ and $X_{5,1}$ have 16 and 128 filters, respectively. Orange arrows depict 3D max-pooling layers, while light blue arrows depict 3D transposed convolutional layers. Dotted dark blue lines depict skip connections between different blocks of layers. All the incoming arrows to blocks of layers are concatenated. The average of layers $X_{1,i}$, where $i = \{2, 3, 4, 5\}$, is computed. The average is fed to the output convolutional layer with 1 filter and kernel size of 1×1. The output is of the same shape as the input of the model.

The denoising network uses 3D layers instead of 2D layers. The use of 3D layers makes it possible to use volumes of CT scans as the input for the denoising network

instead of just individual 2D slices. By using 3D volumes as the input, the network can use the structular information and the connection between neighbouring slices in the denoising process. The temporal information should remove the complex distribution noise better and retain the fine details of the inputs.

Secondary model with similar architecture as in Figure 13 was also trained. This model has 2D-convolutional and pooling layers instead of 3D versions of those and takes a single CT slice as an input. This model was used to compare if a similar denoising result is achievable without the temporal information.

5.2. Training the Model

T TensorFlow framework was chosen [59] for the development of the CNN. TensorFlow is an open source end-to-end framework for development of machine learning models. It provides high-level application programming interfaces (API) such as the neural network library Keras [60]. TensorFlow was developed by Google and has been available since 2015. TensorFlow version 2.3.0 was used.

Training of the model was done on the dataset provided by [61]. The dataset provides both image-domain and projection-domain (sinograms) CT examinations acquired from 299 CT exams. The dataset includes head scans, chest scans, and abdomen scans. All of the scans were performed at routine levels for the anatomical region of interest. The dataset provides both full-dose scans but also simulated low-dose scans where Poisson noise was inserted to the sinograms right before the reconstruction of the images. The head and abdomen scans are simulated to 25 % of the routine dose while the chest scans are simulated to 10 % of the routine dose. Only the examinations in the image domain were used in this thesis. The reconstructed images were in the Digital Imaging and Communications in Medicine (DICOM) standard medical imaging format [62].

The scans were converted from DICOM -format into 16-bit images. Each pixel corresponds to Hounsfield units. The images with HU values I_{HU} were acquired with

$$I_{HU} = \frac{SV}{m} + b \tag{14}$$

where SV are the stored pixel values of the image acquired from the DICOM -file. Rescale slope m and rescale intercept b are also provided in the metadata of the said DICOM -file.

The volumes were created from individual slices. Each volume consisted of 16 slices. The dimensions of the volumes were $512 \times 512 \times 16$. The dataset was split into training, validation, and test data. The training data contained a total of 760 volumes. The validation and test data had 157 and 35 volumes, respectively. Random crops of size 128×128 were taken from the volumes during training. Cropping of the images results in lower memory consumption. The images were randomly flipped vertically with a probability of 50 %. Both random crop and random flip are forms of data augmentation to reduce overfitting of the model. The images were normalized into

the range of −1 to 1 prior to feeding them to the model. Normalized images I_N were acquired from I_{HU} using the equation

$$I_N = 2 \times \frac{I_{HU} - HU_{\text{min}}}{HU_{\text{max}} - HU_{\text{min}}} - 1 \tag{15}$$

where $HU_{\text{min}} = -1024$ and $HU_{\text{max}} = 3071$. These are the minimum and maximum possible values on the Hounsfield scale. The outputs of the model are denormalized back to HUs with the inverse of Equation (15).

TensorFlow is able to utilize datasets in a TFRecord format. In TFRecord format the data is serialized in a binary format. TFRecord is especially useful for datasets that are too large to be stored fully in memory, as only the data that is required at a time is loaded from disk. The dataset can easily be sharded into multiple files. Sharding enables for parallelizing of the data reading.

The loss function used in the training was a combination of L_1 and the perceptual loss (L_P). L_P utilized a VGG-19 classification network [54] pretrained on ImageNet dataset. ImageNet is an image dataset of more than 14 million images and the pretrained VGG-19 is trained using about 1.2 million images. Keras library has an implementation of the VGG-19 network along with the pretrained weights. The 14th convolutional layer of VGG-19 with 512 filters was chosen as the layer to be used in the loss network ϕ. ϕ has a total of 15.3 million parameters. The loss network is visualized in Figure 14. The total loss L_{total} was the sum of L_1 and L_P:

$$L_{total} = L_1 + L_P \tag{16}$$

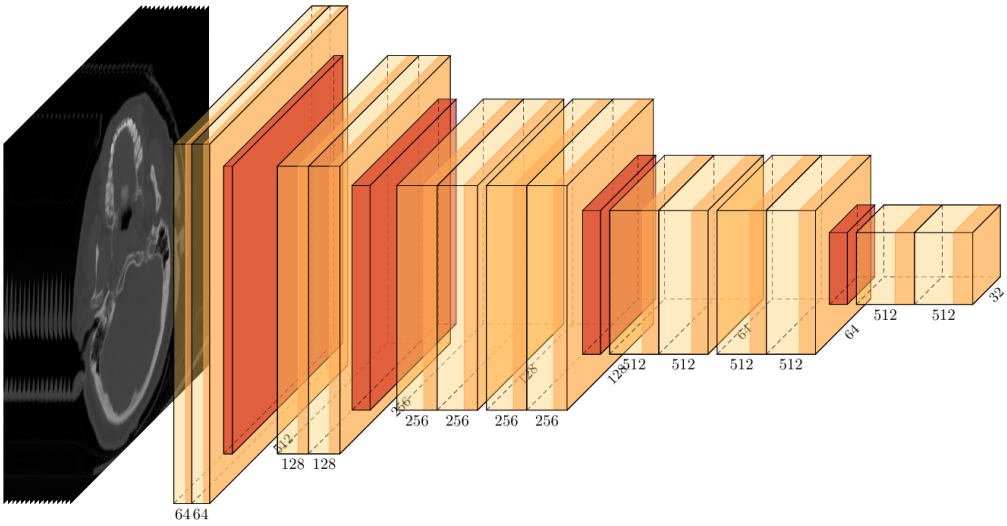


Figure 14. The loss network used in the perceptual loss.

The training of the model was done on two GeForce RTX 2080 Ti GPUs. Adam optimizer was chosen as the optimizer algorithm with a learning rate of 0.0001.

6. EVALUATION

The evaluation of the denoising model was done by comparing its results with the results of traditional denoising methods mentioned in 3.4. *3D* and *2D* in Tables 4 and 5 correspond to the methods introduced in this thesis. Two denoising networks with DnCNN [56] and CNN DAE [58] architectures were also trained from scratch on the same data mentioned in 5.2.

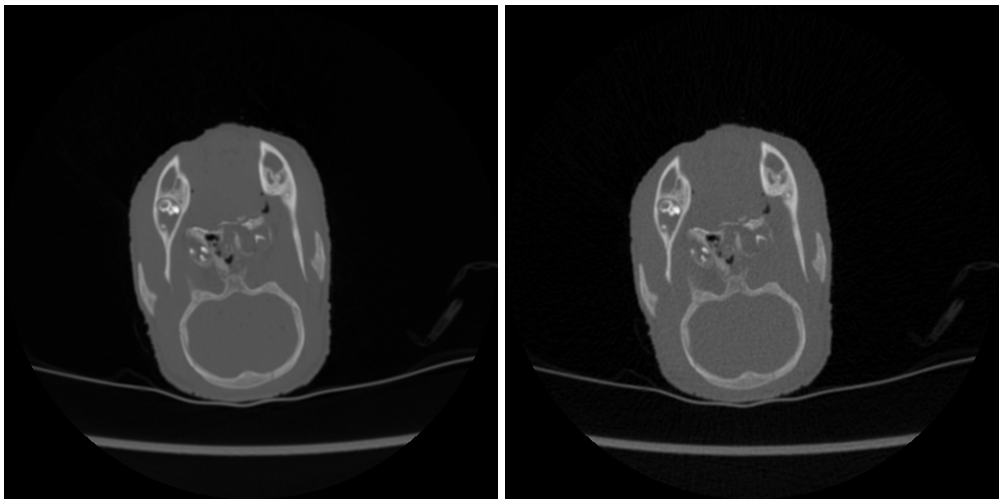
NLM denoising was done by the implementation provided by Scikit-image Python library [63]. First, the noise standard deviation, σ, was estimated by a function provided in the library. A patch size of 5×5 was used along with the search area of 13×13. The function receives the parameter $h = 0{,}8 \times \sigma$ as an argument which controls the decay in patch weights as a function of the distance between patches. The slow NLM algorithm was chosen for the best possible denoising result.

The authors of [32] provide a BM3D Python package. The same estimated σ as above was used in the BM3D algorithm. The full BM3D algorithm was used for the best denoising result.

Median filtering was implemented by OpenCV Python library [64]. The median filter size of 5×5 was selected for the computation.

6.1. Testing

A piglet CT dataset by Yi and Babyn [65] was used for testing. The dataset consisted of real CT scans of a deceased piglet. Scans were taken with 100 kVp X-ray tube voltage and with varying tube currents. 300 mAs scan was the conventional full dose scan. Scans with 50 %, 25 %, 10%, and 5 % dose of full dose were taken. The 5 % scans had a tube current of 15 mAs. The scans were of a size 512×512. 850 slices of each dose were provided. Figure 15 visualizes full-dose and 5 % dose scans. The 5 % scans were used in the testing of the denoising methods.



(a) (b)

Figure 15. Full-dose (a) and 5 % dose (b) CT scans from the piglet CT dataset.

6.1.1. Noise Injection

For testing of the proposed denoising network, simulated very low-dose CT images were needed. The injection of noise to simulate low-dose CT from full-dose CT was implemented based on the following procedure [66, 67]:

1. Compute the Hounsfield unit numbers for the full-dose CT with Equation (14) and acquire $HU_{fulldose}$.
2. Transform the pixel values into linear attenuation coefficients $\mu_{fulldose}$ with Equation (2). Substitute μ_{tissue} in the equation with $\mu_{fulldose}$:

$$\mu_{fulldose} = \frac{\mu_{water}}{1000} HU_{fulldose} + \mu_{water}$$

   The value for μ_{water} depends on the voltage of the X-ray tube used when acquiring the images and is usually also available in the DICOM -file. Some values for the μ_{water} are listed in Table 3.
3. Obtain the projection data $p_{fulldose}$ for the full-dose image by applying a Radon transform on $\mu_{fulldose}$. Multiply the result with voxel size s to eliminate the size factor:

$$p_{fulldose} = radon(\mu_{fulldose}) \times s$$

4. The full-dose transmission data $T_{fulldose}$ is acquired with

$$T_{fulldose} = exp(p_{fulldose})$$

5. The low-dose transmission is generated by injecting $T_{fulldose}$ with Poisson noise:

$$T_{lowdose} = Poisson(I^0_{lowdose} T_{fulldose})$$

   where $I^0_{lowdose}$ is the simulated low-dose scan incident flux.
6. Calculate low-dose projection data $p_{lowdose}$:

$$p_{lowdose} = \ln\left(\frac{I^0_{lowdose}}{T_{lowdose}}\right)$$

7. The projection data of the added noise is

$$p_{noise} = p_{fulldose} - p_{lowdose}$$

8. Now, compute the linear attenuation coefficients for the low-dose CT $\mu_{lowdose}$ by utilizing the inverse Radon transform $iradon$. Divide the result of the inverse transform with the voxel size s and add the noise to $\mu_{fulldose}$:

$$\mu_{lowdose} = \mu_{fulldose} + \frac{iradon(p_{noise})}{s}$$

9. Now the inverse of Equation (2) can be used to transform $\mu_{lowdose}$ into low-dose CT with HU values $HU_{lowdose}$.

$I^0_{lowdose}$ was chosen to have values between 5×10^5 to 1×10^7. Figure 16 visualizes the difference between different $I^0_{lowdose}$ values. A higher $I^0_{lowdose}$ value results in less noise.

Table 3. Water attenuation coefficients μ of water on different X-ray tube kVp.

kVp [keV]	$\mu[\mathrm{cm}^{-1}]$
80	0.431
90	0.341
100	0.276
110	0.228
120	0.192

6.1.2. Lung Segmentation

A secondary test for the denoising network introduced in this thesis is done with a lung segmentation tool. Medical image segmentation is a process of finding the boundaries of a region of interest in the image. The segmentation can be done semi-automatically or fully automatically. Image segmentation can be also done with CNNs. The most famous medical image segmentation CNN is the U-Net [46]. Segmentation can be used for diagnosis purposes and in treatment of diseases. Lung segmentation helps detecting lesions which can be a result of some disease or trauma, or even cancer. The idea is to test if the denoising of low-dose chest CT scans could improve the segmentation result.

The lung segmentation tool used in the testing of the denoising network was introduced by Hofmanninger *et al.* [68]. The authors provide a Python tool capable of automated lung segmentation from chest CT-scans. The tool uses a standard U-Net CNN trained on a large and diverse dataset. The tool is capable of extracting right and left lungs separately, including tumors, effusions, and airpockets. The segmentation result will be evaluated with Jaccard index -metric.

For testing the lung segmentation, the dataset introduced in [69, 70] was used. The dataset consisted of 120 CT series. The dataset also provided ground truth lung segmentation masks created manually by experts. Noise was injected into the images as mentioned in 6.1.1 and the segmentation result was compared with the ground truth segmentations.

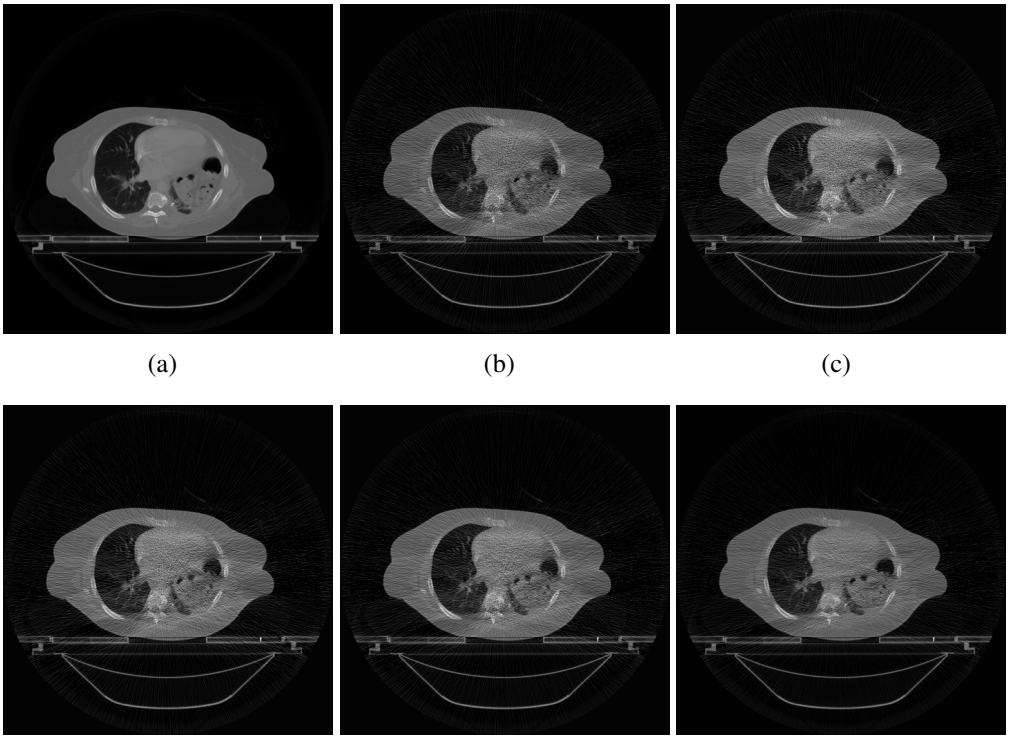
6.1.3. Metrics

Mean squared error (MSE) is a cumulative error. MSE is the squared error between the ground truth I_1 and the noisy image I_2.

The MSE is calculated using the equation

$$\mathrm{MSE}(I_1, I_2) = \frac{\sum_{M,N,K}[I_1(m,n,k) - I_2(m,n,k)]^2}{M \times N \times K} \tag{17}$$

where M, N, and K denotes rows, columns, and depth, respectively.



(a) (b) (c)

(d) (e) (f)

Figure 16. Manual noise injection following procedure explained in 6.1.1. Full-dose scan (a) and noise injected (b-f) with various $I^0_{lowdose}$ values ranging from 6×10^6 to 2×10^7.

The peak signal-to-noise ratio (PSNR) expresses the ratio between the maximum possible value of an image and the power of the noise that affects the said image. The PSNR is expressed in an logarithmic scale.

The equation for PSNR is

$$\mathrm{PSNR} = 10 \log_{10}(\frac{R^2}{\mathrm{MSE}}) \tag{18}$$

where R is the maximum value possible in the image. $R = 3071$, as the CT images are represented in the Hounsfield scale.

Structural-similarity index (SSIM) [71] is used in comparing the quality of digital images or videos. The equation for SSIM between two images is

$$\mathrm{SSIM}(I_1, I_2) = \frac{(2\mu_{I_1}\mu_{I_2} + c_1)(2\sigma_{I_1 I_2} + c_2)}{(\mu^2_{I_1} + \mu^2_{I_2}) + c_1)(\sigma^2_{I_1} + \sigma^2_{I_2} + c_2)} \tag{19}$$

where μ_{I_1} and μ_{I_2} are the averages of images I_1 and I_2, respectively. $\sigma^2_{I_1}$ and $\sigma^2_{I_2}$ are variances for both images. $\sigma_{I_1 I_2}$ is the covariance of I_1 and I_2. c_1 and c_2 are variables used in stabilizing the division. More specifically, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$. L is the bit depth of the image pixels. The default values for k_1 and k_2 are 0.01 and 0.03, respectively.

The Jaccard index is a statistical method used in calculating the accuracy of a segmentation. It is also known as the intersection-over-union between the ground truth segmentation y and predicted segmentation y' and is defined with

$$J(y, y') = \frac{|y' \cap y|}{|y'| + |y| + |y' \cup y|} \tag{20}$$

where $\cup$ denotes union and $\cap$ denotes intersection.

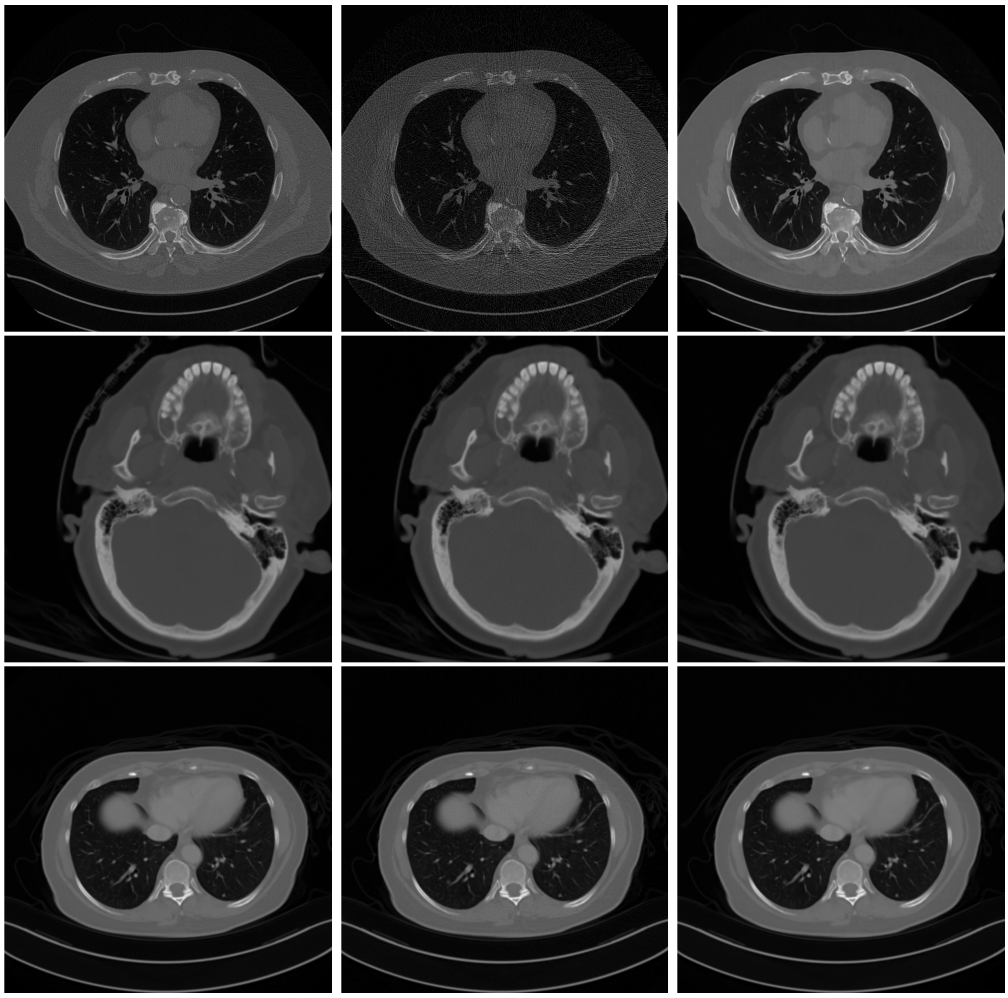
6.2. Results

The denoising results for 35 volumes of simulated low-dose test data are listed in Table 4. Equation (17), Equation (18) and Equation (19) were used to compute metrics for each denoising method. The test data contained CT scans of the chest, head, and abdomen. The chest scans contained more noise than head and abdomen scans, respectively. Figure 17 visualizes example slices of all three anatomical locations, along with the denoised slice acquired by using the 3D network introduced in this thesis. Figure 19 illustrates how the CNN based networks work much better than traditional methods in removing streak artefacts from extremely low-dose CT scans.

Table 5 lists the metrics computed from denoising CT images from the real low-dose dataset [65]. The 2D method outperforms the other methods on real low-dose CT scans. MSE is the lowest of the methods, while PSNR is the highest. 2D network and median filtering have identical SSIM values. The 3D network has the second highest PSNR and the third lowest MSE, after median filtering. SSIM of the 3D network is the third lowest, with only non-local means and BM3D resulting in lower values. Interestingly, CNN DAE outperforms DnCNN on both MSE and PSNR while DnCNN was better on the artificial low-dose dataset.

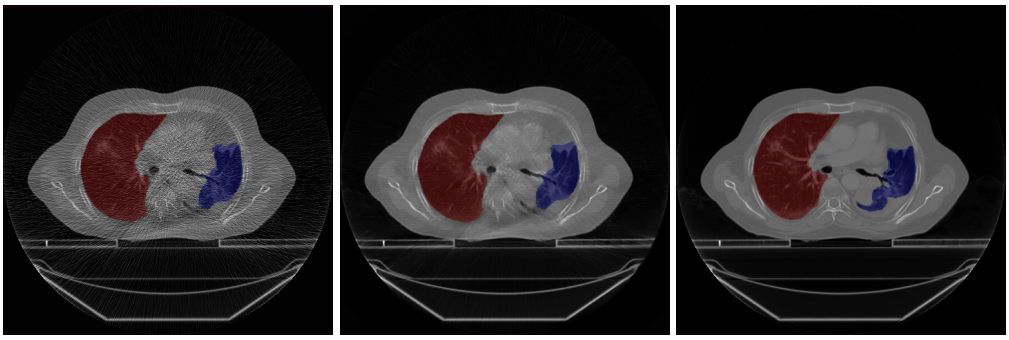
Lung segmentation

Lung segmentation of CT scans denoised with different methods were the secondary test for the denoising results. The results were evaluated using the Jaccard index between the ground truth and segmentation. Example segmentation result is visualized in Figure 18. Result metrics are listed in Table 6. The denoising results do not differ from each other that much, with similar kinds of Jaccard indices achieved with each of the methods. Neural network-based denoising methods were all within 0.0036 of each other, with also the standard deviation being very close to each other. The mean of median filtering, being the best of the traditional denoising methods, also differed just 0.0047 units from the 3D denoising network. The Jaccard indices of all denoising methods were also close to that of the low-dose scan. No significant enhancement to the segmentation was achieved by denoising of the noisy CT scan.



(a) (b) (c)

Figure 17. Full-dose CT (a), simulated low-dose CT (b) and denoised CT (c) for scans from [61]. From top to bottom: chest, head and abdomen.



(a) (b) (c)

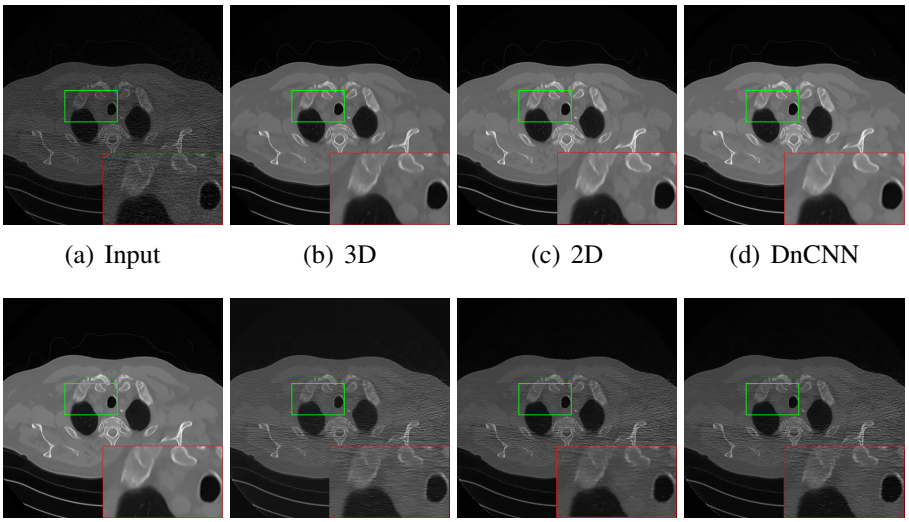
Figure 18. The result of low-dose (a) and denoised (b) lung segmentation with ground truth (c).

Table 4. Mean metrics for different methods on test dataset from [61]. The first row contains the metrics for the simulated low-dose CT images before denoising. Example slices visualized in Figure 17. *3D* and *2D* denote the 3D and 2D networks introduced in this thesis.

		MSE	PSNR	SSIM
Chest	Low-dose	$1.24\times10^5 \pm 5.13\times10^4$	19.19 ± 1.89	0.23 ± 0.07
	3D	$1.11\times10^4 \pm 2.88\times10^3$	29.48 ± 1.35	$\mathbf{0.64 \pm 0.09}$
	2D	$1.15\times10^4 \pm 3.05\times10^3$	29.33 ± 1.40	0.63 ± 0.09
	DnCNN	$\mathbf{1.10\times10^4 \pm 2.86\times10^3}$	29.50 ± 1.34	$\mathbf{0.64 \pm 0.09}$
	CNN DAE	$1.32\times10^4 \pm 3.30\times10^3$	28.69 ± 1.25	0.58 ± 0.09
	Median filtering	$1.69\times10^4 \pm 4.07\times10^3$	27.62 ± 1.13	0.53 ± 0.10
	Non-local means	$6.08\times10^4 \pm 1.62\times10^4$	22.09 ± 1.28	0.44 ± 0.08
	BM3D	$5.45\times10^4 \pm 1.43\times10^4$	22.54 ± 1.23	0.45 ± 0.07
Head	Low-dose	$6.87\times10 \pm 4.74\times10$	55.13 ± 12.05	0.99 ± 0.00
	3D	$5.33\times10 \pm 3.49\times10$	54.04 ± 5.04	$\mathbf{1.00 \pm 0.00}$
	2D	$\mathbf{3.74\times10 \pm 2.77\times10}$	55.32 ± 3.61	$\mathbf{1.00 \pm 0.00}$
	DnCNN	$4.55\times10 \pm 3.21\times10$	55.77 ± 8.34	$\mathbf{1.00 \pm 0.00}$
	CNN DAE	$1.32\times10^2 \pm 1.05\times10^2$	50.62 ± 5.10	0.99 ± 0.00
	Median filtering	$2.82\times10^2 \pm 2.28\times10^2$	50.15 ± 13.66	0.99 ± 0.01
	Non-local means	$2.00\times10^2 \pm 8.86\times10^2$	54.60 ± 12.42	0.99 ± 0.00
	BM3D	$6.71\times10 \pm 4.62\times10$	53.43 ± 5.66	0.99 ± 0.00
Abdomen	Low-dose	$3.40\times10^2 \pm 1.01\times10^2$	44.65 ± 1.42	0.97 ± 0.01
	3D	$\mathbf{1.18\times10^2 \pm 1.54\times10}$	$\mathbf{49.08 \pm 0.57}$	$\mathbf{0.99 \pm 0.00}$
	2D	$1.27\times10^2 \pm 2.89\times10$	48.84 ± 1.03	$\mathbf{0.99 \pm 0.00}$
	DnCNN	$1.79\times10^2 \pm 5.75\times10$	47.46 ± 1.48	$\mathbf{0.99 \pm 0.00}$
	CNN DAE	$3.65\times10^2 \pm 4.61\times10$	44.15 ± 0.55	0.98 ± 0.00
	Median filtering	$9.08\times10^2 \pm 9.32\times10$	40.19 ± 0.44	0.98 ± 0.00
	Non-local means	$3.30\times10^2 \pm 1.00\times10^2$	44.79 ± 1.45	0.97 ± 0.00
	BM3D	$3.25\times10^2 \pm 9.89\times10$	44.86 ± 1.46	0.97 ± 0.01

Table 5. Mean metrics for different methods on real low-dose test dataset from [65]. Example slices visualized in Figure 15. *3D* and *2D* corresponds to the 3D and 2D networks introduced in this thesis.

	MSE	PSNR	SSIM
Low-dose	1805.55 ± 889.00	37.99 ± 3.06	0.93 ± 0.04
3D	1631.76 ± 824.68	38.45 ± 3.10	0.94 ± 0.03
2D	1493.1 ± 745.05	38.84 ± 3.11	$\mathbf{0.96 \pm 0.02}$
DnCNN	1725.57 ± 814.21	38.03 ± 2.61	0.95 ± 0.03
CNN DAE	1626.95 ± 791.18	38.38 ± 2.90	0.95 ± 0.03
Median filtering	1634.68 ± 696.68	38.18 ± 2.49	$\mathbf{0.96 \pm 0.02}$
Non-local means	2000.41 ± 1145.98	37.71 ± 3.30	0.93 ± 0.04
BM3D	1801.09 ± 886.19	38.00 ± 3.06	0.93 ± 0.03



(a) Input (b) 3D (c) 2D (d) DnCNN

(e) CNN DAE (f) BM3D (g) NLM (h) Median filtering

Figure 19. Denoising results of a chest CT scan with different methods.

Table 6. Mean Jaccard indices of lung segmentation from dataset mentioned in 6.1.2. Example segmentation results visualized in Figure 18.

	Jaccard index
Full-dose	0.8475 ± 0.1697
Low-dose	0.7968 ± 0.2575
3D	0.8013 ± 0.2492
2D	0.8007 ± 0.2497
DnCNN	0.8043 ± 0.2466
CNN DAE	0.8010 ± 0.2488
Median filtering	0.7966 ± 0.2627
Non-local means	0.7963 ± 0.2584
BM3D	0.7963 ± 0.2585

7. DISCUSSION

A denoising neural network utilizing 3D convolutional layers was created and trained in this thesis. The network was trained on CT scans corrupted with simulated noise. The CT volumes consisted of 16 slices. A network with 2D convolutional layers with otherwise similar architecture was also trained for comparison. The denoising of the 3D network was also compared with two established denoising neural networks, along with three traditional denoising methods. A secondary test, the effect of denoising on lung segmentation results, was also conducted.

The 3D network outperforms the traditional denoising methods with all different noise levels. Mean MSE values were lower while PSNR and SSIM were higher than any of the traditional methods. The most significant difference was with abdomen CT scans, where the mean PSNR of the 3D network was 49.08 ± 0.57 compared to 40.19 ± 0.44 of median filtering. The difference is significant, as PSNR is in the logarithmic scale. Interestingly, NLM denoising produces better PSNR values on head scans than the 3D network.

The differences between different neural-network -based denoising methods are not as large. Both 3D and 2D networks perform quite equally on the chest scans which are the noisiest of the test dataset. MSE, PSNR, and SSIM values are very close to each other. 3D network achieves smaller MSE and higher PSNR on abdomen scans. 2D network has lower MSE and higher PSNR on head scans. SSIM values are equal on head and abdomen scans. The 3D network and DnCNN have almost identical performance on chest scans, with the mean PSNR and standard deviation of those differing only by 0.02 and 0.01, respectively. MSE values are only slightly lower with DnCNN, and SSIM values are identical. With head scans, DnCNN outperforms 3D network the same way the 2D network also did. With slightly noisier abdomen scans the 3D network achieves better metrics than the DnCNN. The CNN DAE performs most poorly out of all neural-network -based denoising methods, with especially the head scans having much lower PSNR than non-local means or BM3D denoising.

The 3D network provided good results on the noisiest of the CT scans. The results indicate that the structural information between slices could help on the denoising of the scans. The information on tissues spanning several slices perhaps allows for the denoising network to better retain the details of the image. When scans containing lower amounts of noise were denoised, simpler networks and methods could achieve a similar or better result with lower computational costs. The results suggest that 3D convolutional neural networks could be utilized in denoising when ultra-low-dose CT examinations are conducted. That being said, the metrics between DnCNN and the 3D and 2D networks implemented in this thesis are quite close to each other. DnCNN achieved the highest mean PSNR on the head CT scans, but both the 3D and 2D networks had significantly lower standard deviations. This suggests that 3D and 2D networks have less variation even though the mean PSNR was lower.

Traditional denoising methods work well with noise having Gaussian distribution, but the complexity of Poisson distribution combined with the fact that the additional reconstruction of the CT image further modifies the noise model does not allow for efficient denoising using any of the traditional methods, especially for ultra-low-dose CT scans. The nonlinearity of the convolutional neural networks allows the denoising of more complex noise.

To further examine the results, inspection conducted by an expert radiologist is needed. High metrics can be calculated from the noise-free estimates of CT scans even though some crucial details could be missing from the images. The question is if the radiologist would rather use a noisy image containing all the information or artificially denoised image where some information could be missing. A human is capable of interpreting some details even from the noisy images that a computer can not.

A continuing problem in training medical image networks is the availability of the data. The collection of datasets containing medical images is difficult as the labeling of the images must be done by a professional. Also the prevalence of some diseases can be extremely low, resulting in unbalanced datasets, especially in classification problems. Gathering ultra-low-dose CT scans from patients for dataset purposes only is problematic because of the ionizing nature of the X-ray radiation. As the noise model of the CT scans is quite complex, the noise simulation is also very difficult. Self-supervised learning could be used in low-dose CT denoising. This method requires, however, more data.

The results of the segmentation tests were rather surprising. It was safe to assume that denoising lung CT scans could also result in better segmentation of the said scan. The segmentation method used could already be rather robust to noise in the scans. It is possible that preprocessing the scans eliminates information crucial for the segmentation from the images. The effect of denoising on segmentation results requires further research.

In this thesis, denoising CT images in sinogram space was not experimented. Sinogram-domain denoising with 3D CNNs would be an interesting research topic. The noise model in sinograms should be less complex, with no information loss caused by the image reconstruction. The data availability for sinogram domain data is not so good as for reconstructed images. However, the same dataset that was used in this thesis also provided simulated low-dose CT scans in the sinogram space. Perhaps 3D CNN based denoising could be combined with iterative image reconstruction methods, to further improve the image quality and minimize the detail loss.

Another potential research topic would be to expand the network introduced also to other medical imaging modalities. Magnetic resonance imaging also produces 3D images, like CT. I believe that a similar kind of network could provide useful also in denoising magnetic resonance images, as they also suffer from image noise.

8. REFERENCES

[1] Chen H., Zhang Y., Zhang W., Liao P., Li K., Zhou J. & Wang G. (2017) Low-dose CT denoising with convolutional neural network. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), IEEE, pp. 143–146.

[2] Yi X. & Babyn P. (2018) Sharpness-aware low-dose CT denoising using conditional generative adversarial network. Journal of digital imaging 31, pp. 655–669.

[3] Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. & Bengio Y. (2014) Generative adversarial networks. arXiv preprint arXiv:1406.2661 .

[4] Yang Q., Yan P., Zhang Y., Yu H., Shi Y., Mou X., Kalra M.K., Zhang Y., Sun L. & Wang G. (2018) Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss. IEEE transactions on medical imaging 37, pp. 1348–1357.

[5] You C., Yang Q., Shan H., Gjesteby L., Li G., Ju S., Zhang Z., Zhao Z., Zhang Y., Cong W. et al. (2018) Structurally-sensitive multi-scale deep neural network for low-dose CT denoising. IEEE Access 6, pp. 41839–41855.

[6] Ghani M.U. & Karl W.C. (2018) CNN based sinogram denoising for low-dose CT. In: Mathematics in Imaging, Optical Society of America, pp. MM2D–5.

[7] Flohr T. (2013) CT systems. Current Radiology Reports 1, pp. 52–63.

[8] Radon J. (1986) On the determination of functions from their integral values along certain manifolds. IEEE Transactions on Medical Imaging 5, pp. 170–176.

[9] Wang J., Lu H., Liang Z., Eremina D., Zhang G., Wang S., Chen J. & Manzione J. (2008) An experimental study on the noise properties of X-ray CT sinogram data in radon space. Physics in Medicine & Biology 53, p. 3327.

[10] Geyer L.L., Schoepf U.J., Meinel F.G., Nance J.W., Bastarrika G., Leipsic J.A., Paul N.S., Rengo M., Laghi A. & De Cecco C.N. (2015) State of the art: Iterative CT reconstruction techniques. Radiology 276, pp. 339–357.

[11] Padole A., Sainani N., Lira D., Khawaja R.D.A., Pourjabbar S., Gullo R.L., Otrakji A. & Kalra M.K. (2016) Assessment of sub-milli-sievert abdominal computed tomography with iterative reconstruction techniques of different vendors. World journal of radiology 8, p. 618.

[12] Park C., Choo K.S., Jung Y., Jeong H.S., Hwang J.Y. & Yun M.S. (2021) Ct iterative vs deep learning reconstruction: comparison of noise and sharpness. European radiology 31, pp. 3156–3164.

[13] McCollough C.H., Yu L., Kofler J.M., Leng S., Zhang Y., Li Z. & Carter R.E. (2015) Degradation of ct low-contrast spatial resolution due to the use of iterative reconstruction and reduced dose levels. Radiology 276, pp. 499–506.

[14] Rozema R., Kruitbosch H.T., van Minnen B., Dorgelo B., Kraeima J. & van Ooijen P.M. (2020) Iterative reconstruction and deep learning algorithms for enabling low dose computed tomography in midfacial trauma. Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology .

[15] Xie S., Zheng X., Chen Y., Xie L., Liu J., Zhang Y., Yan J., Zhu H. & Hu Y. (2018) Artifact removal using improved googlenet for sparse-view ct reconstruction. Scientific reports 8, pp. 1–9.

[16] Zhang Y. & Yu H. (2018) Convolutional neural network based metal artifact reduction in x-ray computed tomography. IEEE transactions on medical imaging 37, pp. 1370–1381.

[17] Tatsugami F., Higaki T., Nakamura Y., Yu Z., Zhou J., Lu Y., Fujioka C., Kitagawa T., Kihara Y., Iida M. et al. (2019) Deep learning–based image restoration algorithm for coronary ct angiography. European radiology 29, pp. 5322–5329.

[18] Singh R., Digumarthy S.R., Muse V.V., Kambadakone A.R., Blake M.A., Tabari A., Hoi Y., Akino N., Angel E., Madan R. et al. (2020) Image quality and lesion detection on deep learning reconstruction and iterative reconstruction of submillisievert chest and abdominal ct. American Journal of Roentgenology 214, pp. 566–573.

[19] Team N.L.S.T.R. (2011) Reduced lung-cancer mortality with low-dose computed tomographic screening. New England Journal of Medicine 365, pp. 395–409.

[20] Raman S.P., Mahesh M., Blasko R.V. & Fishman E.K. (2013) CT scan parameters and radiation dose: practical advice for radiologists. Journal of the American College of Radiology 10, pp. 840–846.

[21] Rampinelli C., Origgi D. & Bellomi M. (2012) Low-dose CT: technique, reading methods and image interpretation. Cancer Imaging 12, p. 548.

[22] ICRP (2007) The 2007 recommendations of the international commission on radiological protection, vol. 37. ICRP Publication 103. Ann. ICRP 37 (2-4).

[23] Yanagawa M., Honda O., Kikuyama A., Gyobu T., Sumikawa H., Koyama M. & Tomiyama N. (2012) Pulmonary nodules: Effect of adaptive statistical iterative reconstruction (ASIR) technique on performance of a computer-aided detection (CAD) system—comparison of performance between different-dose CT scans. European Journal of Radiology 81, pp. 2877–2886.

[24] Ono K., Hiraoka T., Ono A., Komatsu E., Shigenaga T., Takaki H., Maeda T., Ogusu H., Yoshida S., Fukushima K. et al. (2013) Low-dose CT scan screening for lung cancer: comparison of images and radiation doses between low-dose CT and follow-up standard diagnostic CT. SpringerPlus 2, pp. 1–8.

[25] Dangis A., Gieraerts C., Bruecker Y.D., Janssen L., Valgaeren H., Obbels D., Gillis M., Ranst M.V., Frans J., Demeyere A. et al. (2020) Accuracy and

reproducibility of low-dose submillisievert chest CT for the diagnosis of COVID-19. Radiology: Cardiothoracic Imaging 2, p. e200196.

[26] Kang Z., Li X. & Zhou S. (2020) Recommendation of low-dose CT in the detection and management of COVID-2019. European radiology 30, pp. 4356–4357.

[27] Macovski A. (1983) Medical imaging systems. Prentice Hall.

[28] Sagheer S.V.M. & George S.N. (2020) A review on medical image denoising algorithms. Biomedical signal processing and control 61, p. 102036.

[29] Li T., Li X., Wang J., Wen J., Lu H., Hsieh J. & Liang Z. (2004) Nonlinear sinogram smoothing for low-dose X-ray CT. IEEE Transactions on Nuclear Science 51, pp. 2505–2513.

[30] Wang J., Li T., Lu H. & Liang Z. (2006) Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose X-ray computed tomography. IEEE transactions on medical imaging 25, pp. 1272–1283.

[31] Buades A., Coll B. & Morel J.M. (2005) A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, IEEE, vol. 2, pp. 60–65.

[32] Dabov K., Foi A., Katkovnik V. & Egiazarian K. (2007) Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing 16, pp. 2080–2095.

[33] Kim B., Han M., Shim H. & Baek J. (2019) A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose ct images. Medical physics 46, pp. 3906–3923.

[34] Mitchell T. (1997) Machine Learning. McGraw-Hill International Editions, McGraw-Hill.

[35] Tiulpin A., Thevenot J., Rahtu E., Lehenkari P. & Saarakkala S. (2018) Automatic knee osteoarthritis diagnosis from plain radiographs: a deep learning-based approach. Scientific reports 8, pp. 1–10.

[36] Wang J., Knol M.J., Tiulpin A., Dubost F., de Bruijne M., Vernooij M.W., Adams H.H., Ikram M.A., Niessen W.J. & Roshchupkin G.V. (2019) Gray matter age prediction as a biomarker for risk of dementia. Proceedings of the National Academy of Sciences 116, pp. 21213–21218.

[37] Zulkifley M.A., Abdani S.R. & Zulkifley N.H. (2020) Automated bone age assessment with image registration using hand X-ray images. Applied Sciences 10, p. 7233.

[38] Oh K.S. & Jung K. (2004) GPU implementation of neural networks. Pattern Recognition 37, pp. 1311–1314.

[39] Wu Y., Schuster M., Chen Z., Le Q.V., Norouzi M., Macherey W., Krikun M., Cao Y., Gao Q., Macherey K. et al. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 .

[40] Goodfellow I., Bengio Y. & Courville A. (2016) Deep Learning. MIT Press. URL: http://www.deeplearningbook.org.

[41] Zeiler M.D., Krishnan D., Taylor G.W. & Fergus R. (2010) Deconvolutional networks. In: 2010 IEEE Computer Society Conference on computer vision and pattern recognition, IEEE, pp. 2528–2535.

[42] Dumoulin V. & Visin F. (2016) A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 .

[43] Albawi S., Abed Mohammed T. & ALZAWI S. (2017) Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6.

[44] Zhou Y. & Chellappa R. (1988) Computation of optical flow using a neural network. IEEE 1988 International Conference on Neural Networks , pp. 71–78 vol.2.

[45] He K., Zhang X., Ren S. & Sun J. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[46] Ronneberger O., Fischer P. & Brox T. (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer, pp. 234–241.

[47] Szegedy C., Ioffe S., Vanhoucke V. & Alemi A. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, vol. 31.

[48] Iqbal H. (2018), HarisIqbal88/PlotNeuralNet v1.0.0. URL: https://doi.org/10.5281/zenodo.2526396, (Accessed 21.04.2021).

[49] Zhou Z., Rahman Siddiquee M.M., Tajbakhsh N. & Liang J. (2018) UNet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp. 3–11.

[50] Rumelhart D.E., Hinton G.E. & Williams R.J. (1986) Learning representations by back-propagating errors. nature 323, pp. 533–536.

[51] Kingma D.P. & Ba J. (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[52] Ruder S. (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 .

[53] Johnson J., Alahi A. & Fei-Fei L. (2016) Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision, Springer, pp. 694–711.

[54] Simonyan K. & Zisserman A. (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

[55] Deng J., Dong W., Socher R., Li L., Kai Li & Li Fei-Fei (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

[56] Zhang K., Zuo W., Chen Y., Meng D. & Zhang L. (2017) Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing 26, pp. 3142–3155.

[57] Ioffe S. & Szegedy C. (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift.

[58] Gondara L. (2016) Medical image denoising using convolutional denoising autoencoders. CoRR abs/1608.04667.

[59] Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M. et al. (2016) Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283.

[60] Chollet F. (2015), keras. `https://github.com/fchollet/keras`.

[61] Moen T.R., Chen B., Holmes III D.R., Duan X., Yu Z., Yu L., Leng S., Fletcher J.G. & McCollough C.H. (2020) Low dose CT image and projection dataset. Medical Physics n/a.

[62] Association N.E.M. (2020), Digital imaging and communications in medicine (DICOM) standard. NEMA PS3 / ISO 12052, (Available free at http://medical.nema.org/).

[63] Van der Walt S., Schönberger J.L., Nunez-Iglesias J., Boulogne F., Warner J.D., Yager N., Gouillart E. & Yu T. (2014) scikit-image: image processing in python. PeerJ 2, p. e453.

[64] Bradski G. (2000) The opencv library. Dr. Dobb's Journal of Software Tools .

[65] Yi X. & Babyn P. (2017) Sharpness-aware low-dose CT denoising using conditional generative adversarial network. Journal of Digital Imaging 31.

[66] Zeng D., Huang J., Bian Z., Niu S., Zhang H., Feng Q., Liang Z. & Ma J. (2015) A simple low-dose X-ray CT simulation from high-dose scan. Nuclear Science, IEEE Transactions on 62, pp. 2226–2233.

[67] Bevins N., Szczykutowicz T. & Supanich M. (2013) TU-C-103-06: A simple method for simulating reduced-dose images for evaluation of clinical CT protocols. Medical Physics 40.

[68] Hofmanninger J., Prayer F., Pan J., Röhrich S., Prosch H. & Langs G. (2020) Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. European Radiology Experimental 4, pp. 1–13.

[69] Yang J., Sharp G., Veeraraghavan H., van Elmpt W., Dekker A., Lustberg T. & Gooding M., Data from lung CT segmentation challenge.

[70] Yang J., Veeraraghavan H., Armato III S.G., Farahani K., Kirby J.S., Kalpathy-Kramer J. et al. (2018) Autosegmentation for thoracic radiation treatment planning: A grand challenge at aapm 2017. Medical Physics 45, pp. 4568–4581.

[71] Wang Z., Bovik A.C., Sheikh H.R. & Simoncelli E.P. (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, pp. 600–612.