Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

A new metal artifact reduction algorithm with a GAN approach

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
Elena Loli Piccolomini
Correlatore:
Marco Soldini

Presentata da:
Sofia Cannizzaro

Sessione unica
Anno Accademico 2020/2021

Contents

Introduction

Computed tomography (CT) is a diagnostic imaging test used to create detailed images of internal organs, bones, soft tissue, and blood vessels. It works by reconstructing an approximation of the original volume, or image, using projection data obtained from an X-ray beam that flows through the patient's body. Several efforts were made toward three-dimensional radiographic imaging, until the introduction of cone-beam CT, a novel 3D X-ray computed tomography technique that entered and changed the dentistry diagnostic area. However, some artifacts can substantially compromise the quality of cone beam CT images, even making them diagnostically worthless. In this dissertation, we will focus on metal artifacts (related to the presence of metal in the scanning area) and implement a metal artifact reduction (MAR) algorithm. This work was developed during an internship at CEFLA, a company specialized in medical equipment, particularly in the dental fields. This thesis has two main goals. First, we wish to research and analyze the most popular metal artifact reduction algorithms, with a focus on those that employ deep learning techniques. Second, after selecting a promising method, we wish to develop a MAR algorithm; to do so we will also need to generate a simulated dataset, to train and test the model. In this second part of our work we considered a Generative Adversarial network and developed some changes in the loss function and on some parameters to obtain more accurate results in our dataset: this can be done thanks to a detailed analysis of the corrected images and of the loss decrease. The results obtained are promising and, thanks to the changes made, are persistent even in data slightly different from those used in the training. This work also allows many further development to improve network performance, for example by creating a larger and more detailed dataset. This thesis is organized as follows.

- In the first chapter, we will discuss computed tomography and, in particular, how a CBCT scan works. We will also detail the main artifacts that arise in these types of scans, with a focus on metal artifacts.
- Then, in the second chapter, we will first present and categorize the main state-of-the-art MAR algorithm. Then we'll go over the Generative adversarial network (GAN) and some other deep learning concepts that will be employed in our model. Finally, we'll go over the specifics of our model and how it works. In this chapter, we will also explain how we generated our dataset.
- The numerical results obtained using the implemented approach will be shown in the final chapter. We examined the outcomes obtained with several versions of our model to determine which one is the most effective.

Introduzione

La tomografia computerizzata (CT) è un test di diagnostica utilizzato per creare immagini dettagliate di organi interni, ossa, tessuti molli e vasi sanguigni. Funziona ricostruendo un'approssimazione del volume, o dell'immagine originale, utilizzando i dati di proiezione, ottenuti da un raggio di raggi X che scorre attraverso il corpo del paziente. Diversi sforzi sono stati fatti verso l'imaging radiografico tridimensionale, prima dell'introduzione della cone-beam CT, una nuova tecnica di tomografia computerizzata che ha determinato un cambio di approccio nella diagnostica odontoiatrica. Tuttavia, alcuni artefatti possono compromettere in modo sostanziale la qualità delle immagini cone beam, rendendole persino prive di valore diagnostico. In questa tesi, ci concentreremo sugli artefatti metallici (dovuti alla presenza di metallo nell'area di scansione) ed implementeremo un algoritmo di riduzione degli artefatti metallici (MAR). Questo lavoro è stato sviluppato durante uno stage presso CEFLA, azienda specializzata in apparecchiature mediche, in particolare in campo odontoiatrico. Questa tesi ha due obiettivi principali. In primo luogo, desideriamo ricercare e analizzare i più diffusi algoritmi di riduzione degli artefatti metallici, con particolare attenzione a quelli che utilizzano tecniche di deep learning. In secondo luogo, dopo aver selezionato un metodo promettente, desideriamo sviluppare un algoritmo MAR; per fare ciò, sarà necessario generare un set di dati simulato per addestrare e testare il modello. In questa seconda parte del nostro lavoro abbiamo considerato un Generative Adversarial network e abbiamo sviluppato alcune modifiche nella loss function e in alcuni parametri, per ottenere risultati più accurati nel nostro dataset, grazie ad un'analisi dettagliata delle immagini corrette. I risultati ottenuti sono promettenti e, grazie alle modifiche effettuate, sono persistenti anche in dati leggermente diversi da quelli utilizzati nel training. Questo lavoro, inoltre, permette molti sbocchi per migliorare la performance della rete, ad esempio creando un dataset più ampio e dettagliato. Questa tesi è organizzata come segue.

- Nel primo capitolo parleremo della tomografia computerizzata ed, in particolare, di come funziona una scansione CBCT. Descriveremo inoltre in dettaglio i principali artefatti che emergono in questi tipi di scansioni, con particolare attenzione agli artefatti metallici.

- Quindi, nel secondo capitolo, presenteremo e classificheremo i principali algoritmi di MAR allo stato dell'arte. Quindi esamineremo i Generative adversarial network(GAN) e altri concetti di deep learning che verranno impiegati nel nostro modello. Infine, esamineremo le specifiche del nostro modello e il suo funzionamento. In questo capitolo spiegheremo anche come abbiamo generato il nostro set di dati.

- I risultati numerici ottenuti utilizzando l'algoritmo implementato saranno mostrati nel capitolo finale. Abbiamo esaminato i risultati ottenuti con diverse versioni del nostro modello per determinare quale sia il più efficace.

Chapter 1

Metal artifact in Computed thomography

1.1 Computed thomography(CT)

Computed tomography (CT) is a diagnostic imaging test used to create detailed images of internal organs, bones, soft tissue and blood vessels. This technique uses a series of X-ray measurements taken from different angles around the object to produce cross-sectional images or "slices" of anatomy, that are used for a variety of diagnostic and therapeutic purposes. In fact, as the word tomography itself suggests, from the ancient greek, témnó (or tómos) that means “to slice” and from gráphó that means “to write” or “to draw”, a computed tomographic system digitally draws a series of virtual images of a specific object. It has had a revolutionary impact in diagnostic medicine and has also been used successfully in industrial non-destructive testing applications. In 1972 Hounsfield patented the first CT scanner and he was awarded a Nobel Prize together with Cormack for this invention in 1979. Ever since, new developments have led to faster scanning, better dose usage and improved image quality.

To perform classical two-dimensional CT an X-ray source circularly rotates in a donut-shaped structure called a gantry. The X-ray tube rotates around the patient, that lies on a bed located in the opening of the gantry, and shots X-rays beams through the body.

Figure 1.1: Mr. Hounsfield with the first CT scan

On the opposite side of the source, the X-rays are picked up by digital detectors and transmitted to a computer. From a physical point of view the acquired data (called the projections) from the scanning process indicate the decrease in X-ray intensity along a number of linear pathways. Since different anatomical structures (different materials) have different attenuation coefficients (a measurement of a material's tendency to absorb X-rays of a given energy), we can reconstruct the object as a picture of the material's linear attenuation coefficients. Indeed, once a full rotation is completed, the multiple X-ray measurements taken from different angles are processed on a computer using reconstruction algorithms to produce a a 2D image slice of the patient. The most commonly used algorithm for tomographic reconstruction is filtered backprojection (FBP) and, as the name suggests, it consists of two steps, filtering of projection data followed by backprojection (BP). The latter can be seen as the dual, or in a mathematical sense the adjoint, of projection. Instead of projecting density values to a projection value, a projection value is backprojected over the image points along the ray. Eventually the image is stored and the bed moves forward into the gantry so that the scanning process can be repeated to produce other image slices.

An other important stage in reconstruction methods is forward projection: it provides the theoretical model of the actual data collecting process in computed tomography. Mathematically it is the adjoint operation of backprojection and, in image reconstruction, it is customary to iterate the computation of forward projections of the current image estimate and backprojections of the predicted projections.

Since the first CT scanner was developed, a number of improvements to the scanner's equipment have been made, resulting in the evolution of successive generations of CT scanners. The placement of the X-ray tube and detectors, as well as how they move in relation to each other, defines each generation of CT scanner.

1.2 Cone-beam CT

The previous section's two-dimensional methods can recreate a slice of the measured object. If a volume need to be reconstructed, the method must be performed slice by slice, with a little movement of the object or the source-detector system between each slice. Numerous efforts have been made toward three-dimensional radiographic imaging and although CT has been available, its application in dentistry has been limited because of cost, access, and dose considerations. The introduction of cone-beam computed tomography (CBCT) represent a true paradigm shift from a 2D to a 3D approach in data acquisition and image reconstruction.
It shows that for volumetric CT a two-dimensional detector and the use of rays that form a cone with its base on the detector and its apex on the source, is a more efficient acquisition setup, as shown in figure 1.2.

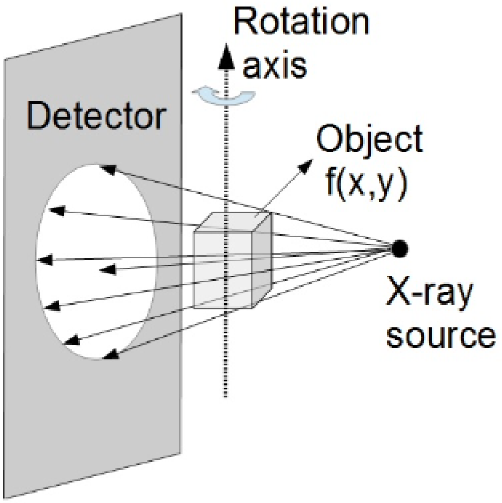


Figure 1.2: Cone-beam geometry

In this asset single exposures are made at certain degree intervals, providing individual 2D projection images, known as “frame” or “raw” images. The complete series of images is referred to as the “projection data”. The frame rate (number of photos acquired per second), the completeness of the trajectory arc and the rotation speed determine the number of images constituting the projection data. More projection data allows for more information to be used to reconstruct the image.
Most CBCT imaging systems use a complete circular trajectory or a scan arc of 360° to acquire projection data. This physical requirement is usually necessary to produce projection data adequate for 3D reconstruction using the FDK algorithm (which is one of the most used algorithm).

Once the basic projection frames have been obtained, data must be processed to produce the volumetric data set. This is referred to as reconstruction. The number of individual projection frames can range from 100 to over 600, each with over one million pixels, making data reconstruction computationally challenging.

For the reconstruction the images must be related to each other and assembled. One method involves constructing a sinogram (figure 1.3): a composite image that is obtained

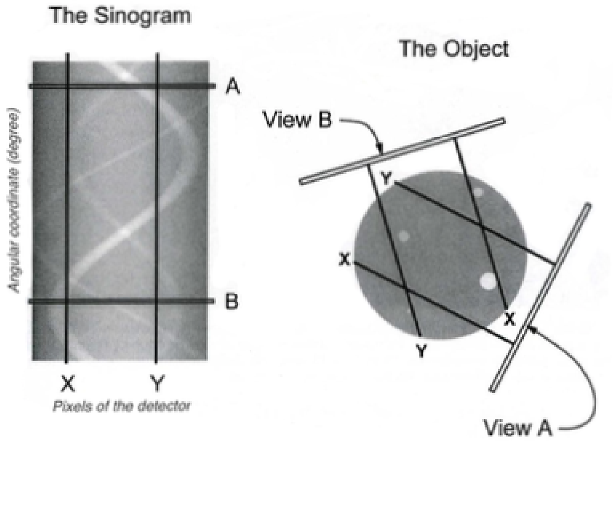


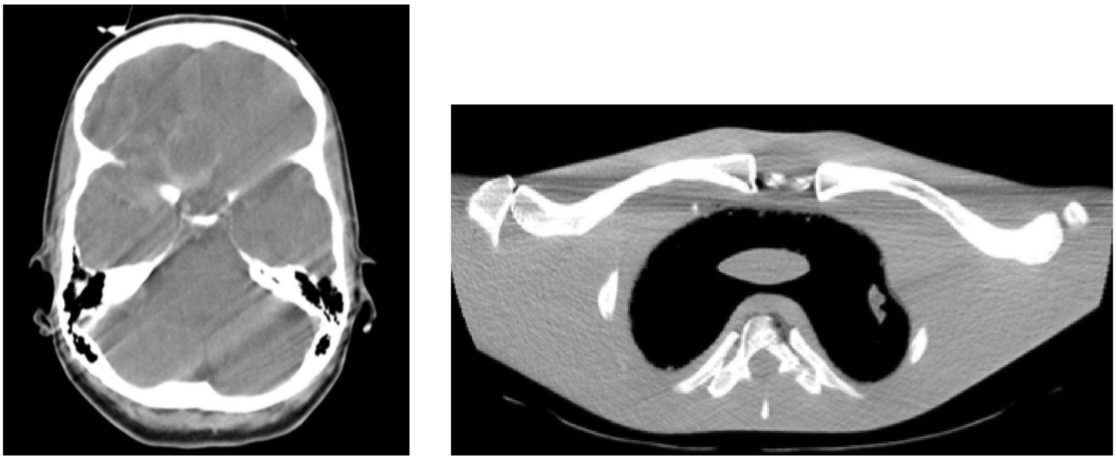
Figure 1.3: Construction of the sinogram

extracting individual rows of each projection image. The first row on the first projection image is used to form the first row on the composite image, the first row on the second image is used to form the second row on the composite image, and so on, until the first row on the *n-th* image forms the *n-th* row on the composite image. The resulting image comprises multiple sine waves of different amplitudes. The waves represent features in the object that are being rotated over 360°.

The final step in the reconstruction stage is processing the corrected sinograms. The sinogram is converted into a full 2D CT slice using a reconstruction filter method. The FDK technique is the most frequently used filtered back projection algorithm for cone-beam–acquired data. After reconstructing all of the slices, they may be recombined into a single volume for visualization.
With the availability of CBCT technology, the dental professional has a wide range of different image presentation. The volumetric data set is a collection of all accessible voxels, that is shown on screen by most CBCT systems as secondary reconstructed pictures in three orthogonal planes (axial, sagittal, and coronal).

1.3 Artifact

The term "artifact" in computed tomography refers to any systematic difference between the CT approximation in the reconstructed images and the real attenuation coefficients of the object. Therefore it refers to any image distortion or inaccuracy that is unrelated to the topic being investigated. Because CT images are reconstructed from millions of individual detector measurements, they are intrinsically more prone to artifacts than traditional radiography. Because the reconstruction approach assumes that all of these data are accurate, any measurement inaccuracy will normally show up as a defect in the rebuilt image.



(a) CT image of the head shows motion artifacts.

(b) CT image of a shoulder phantom shows streaking artifacts.

Figure 1.4: Examples of images with artifacts, images taken from [23].

In the scientific literature, the following relevant artefacts are reported:

- **streaking**: due to an inconsistency in a single measurement.
- **shading**: due to a group of channels or views deviating gradually from the true measurement
- **rings**: due to errors in an individual detector calibration.
- **distortion**: due to helical reconstruction.

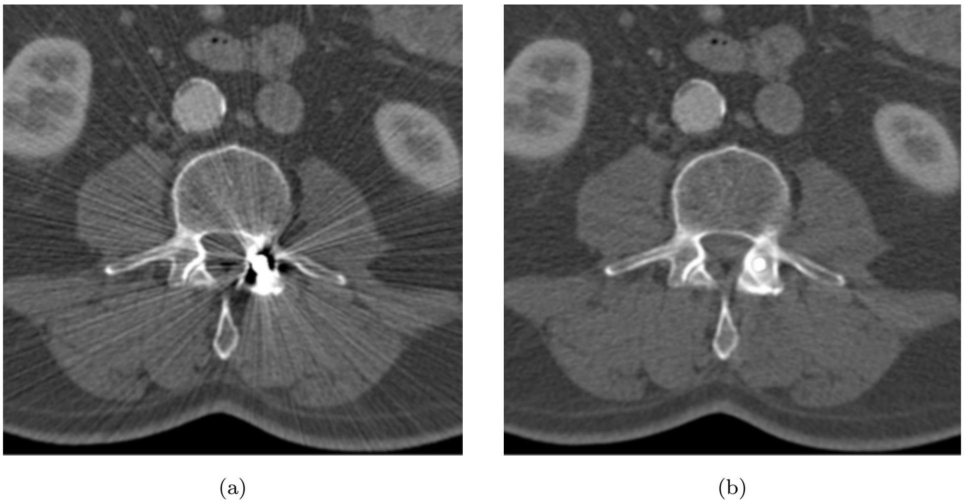
Artifacts can severely degrade the quality of computed tomographic images, making them diagnostically useless in some cases. Understanding why artifacts appear and how to prevent or suppress them is crucial for optimizing image quality. Artifacts are divided into four categories according to their causes:

- physics-based artifacts, which result from the physical processes involved in the acquisition of CT data.
- patient-based artifacts: caused by patient movement or the presence of metallic materials in or on the patient.

- scanner-based artifacts: which result from imperfections in scanner function.
- helical and multi-section artifacts, which are produced by the image reconstruction process.

Some types of artifacts are minimized by design elements introduced into modern CT scanners, while others can be partially corrected by scanner software.

1.3.1 Metal artifact


(a)

(b)

Figure 1.5: CT images of a patient with metal spine implants, reconstructed without any correction (a) and with metal artifact reduction (b), images taken from [23].

In this dissertation the main purpose was to study metal artifact reduction algorithm. Metal artifacts are due to the presence of metal objects in the scan field and appear on an image as a streaking effect, with areas of increased and decreased density obscuring nearby features. Metal objects in the field of view will strongly attenuate x-rays or even completely block their penetration, resulting in corrupt or missing projection data received by the detector. When an image is reconstructed using this incomplete data, it leads to unnatural changes in appearance, the artifacts. The number of interfaces between an X-ray beam and a piece of metal can also influence the amount of artifact produced. More artifact can be expected with more complex shapes or greater numbers of metallic part.

Chapter 2

Metal artifact reduction algorithm

2.1 State-of-the-art MAR algorithm

There are many existing MAR techniques that address the above-mentioned artifact causes: many methods and correction algorithms have been proposed and tested to improve image quality and recover information about underlying structures. Of course the most obvious way to reduce metal artifacts in CT is to avoid them in the first place, so patients are usually requested to remove any metal objects, such as jewelry, before scanning. Non-removable things, such as dental fillings, prosthetic devices, and surgical clips, can occasionally be excluded from scans of nearby anatomy using gantry angulation. The next chance to minimize artifacts is in the data acquisition phase, where modifications can be made to x-ray tube parameters, the detector, and the scan geometry. Multiple energies can also be employed to obtain information at different x-ray spectra [7].

Although these adjustments increase the intrinsic fidelity of the data, MAR with only scan acquisition adjustments still does not yield sufficient image quality in many clinical applications. For small objects with low density, such as surgical clips, that may only cause minor beam hardening or scatter, a physics-based pre-processing correction of the data may be sufficient; but larger and/or denser implants, such as dental fillings, will likely require the raw data to be corrected and/or the reconstruction algorithm to be improved.

These correction and reconstruction-based MAR approaches are by far the most widely researched. A normal CT scan collects projection data from various angles of an object and reconstructs the measurements into an image using filtered back-projection (FBP) or iterative estimation; however reconstruction with incorrect or incomplete data results in an image with artifacts. A common approach is to directly correct or replace the corrupt projection data in the sinogram (**projection completion**). Other methods are **iterative** algorithms that are advanced CT reconstruction techniques which use probabilistic forward and backward models to reduce error propagation during CT reconstruction. Lastly, if raw projection data is not accessible, the inpainting techniques are applied to already reconstructed CT scans and they replace artifact corrupted CT pixels with good-estimated values. There are several hybrid procedures that combine techniques and algorithms from multiple categories.

Recently, thanks to the increasing availability of computational resources, very promising results in the field of medical imaging have been produced using **machine learning**, including metal artifact reduction in CT scans. For example, the performance of convolutional neural networks (CNNs) has been assessed in combination with sinogram inpainting for artifact correction. The deep learning techniques are powerful in learning and capturing the detailed features and patterns of the metal artifacts.

Lastly there are also some example of morphological approach of post-processing of the image that can also be used after an other method of interpolation, for example [6] used two approach: first a morphological approach to reduce the reflection effect of metal in CBCT image and then contrast enhancement approach is used to get best visual image.

2.1.1 Projection completion algorithm

In projection completion new projection data must be synthesized to complete the sinogram and this can be done with different approach:

- INTERPOLATION, means interpolating the replacement values, either from neighboring projections or from a mathematical model.

- REPROJECTION, incorporating prior knowledge to guide the estimation of the data that replaces corrupt projections. A prior image is reprojected (that means forward projected) to generate projection data for sinogram completion.
- NORMALIZATION, employing a normalization step that compares raw sinogram data to a prior image sinogram.

The most common approach is sinogram interpolation, although it frequently introduces new artifacts because the whole information from the metal shadow cannot be recovered. These artifacts are accentuated when high-contrast structures, such as teeth, are present. Because interpolation is less challenging in homogeneous data, *Meyer et al.* [16] introduced a new type of projection completion technique, called normalized MAR (NMAR), that normalizes original projection data to prior image projection data.

In the first step the uncorrected image is reconstructed. The metal trace is then determined in the image domain (using thresholding) and identified in the original projections using a 3D forward projection. Forward projections must be performed in the exact geometry of the uncorrected projections, for a correct replacement of the metal projections. At this point the sinogram is transformed in order to perform the interpolation on a nearly flat, normalized sinogram. The normalization is based on a prior image's 3D forward projection, which should represent the images as closely as possible while including no artifacts. Finding a good prior image is a critical step for NMAR, and in order to do so, an artifact-free prior image is obtained by multithreshold segmentation of the original image, after smoothing to define regions of bone, air, and soft tissue. Then the normalized projections are subject to an interpolation-based MAR operation (the metal projections determine where data in the normalized sinogram are replaced by interpolation). Finally, the corrected sinogram is obtained by denormalization of the interpolated, normalized sinogram and reconstruction of this corrected sinogram yields the corrected image.

Even in severe cases, NMAR can significantly reduce artifacts close to the metal surface, but it is dependent on a strong prior image with correct segmentation. When compared to iterative approaches, this method improves image quality while being more computationally efficient.

2.1.2 Iterative method

In iterative method the aim is to ignore, or statistically down-weight data affected by the metal object and use more reliable data in the reconstruction. It starts with a presumed image and its forward-projection: this data is compared with the actual measured CT data according to statistical metrics, and the computed difference is itself used to create a new updated image with lower noise. The goal then is the optimization of an objective function to minimize the error between these data. This sequence is repeated until the difference becomes minimal. Examples of this objective function include minimum least squares error and maximum likelihood, which aims to find the distribution of linear attenuation coefficients from the projection data with the maximum probability. There may be various approach to that:

- CORRUPTION AVOIDANCE: completely ignore the subset of projection data that is corrupted by metal objects, this model only use data outside the metal trace to arrive at a reconstruction result.
- STATISTICAL COMPENSATION: use a statistical objective function to down-weights data corrupted by the metal object, but still includes them in the iterative reconstruction.
- KNOWLEDGE UTILIZATION : involve incomplete data but improving the method with further considerations, such as known component models. A lot of CT images contain implants so it's possible to use these precisely known physical models, for example using in the model the shape, size, and/or density information of the metal objects to help estimate the missing projections.

Such statistical iterative reconstruction may result in rather longer reconstruction time but significantly less image noise from the same raw data. *Zhang* proposed in 2019 an promising iterative method [1] that include the use of the NMAR correction method. In this approach tha starting point of the iteration is the NMAR corrected image and this is used also as a prior image to guide the algorithm, integrating it in a Bayesian inference framework.

2.1.3 Deep learning-based method

Deep learning has been successfully applied in the field of image restoration and denoising, providing a new method for MAR in CT images: although classical MAR methods have problems in recognizing the patterns of metal objects, deep learning algorithms can deal with this hard task well. Moreover, current improvements in the use of deep learning MAR algorithms do not require paired clinical CT scanning for the reduction of artifacts. As a result, adopting a deep learning methodology to design a MAR method that targets a certain pattern of metal artifacts and anatomical structure will be a promising option. For the first time, the concept of deep learning was used to metal artifact reduction in 2017 by *Park et al.* [9] that used a U-Net to correct metal-induced beam hardening. Then plenty of 2D Convolutional Neural Network, (CNN)-based MAR techniques, that worked in the projection or image domains, were suggested. For example *Zhang et al.* [11] proposed a CNNMAR model to first estimate a prior image by a CNN and then correct sinogram with the same method as NMAR. However, despite CNNs' high expressive power, these techniques suffer from secondary artifacts caused by inconsistent sinograms. Other research explored directly reducing metal artifacts in the CT image domain, for example in 2018 *Hengyong et al.* [10] proposed a method that includes a convolutional network in the image domain, whose output was forward projected to obtain a metal free sinogram and then used for interpolation of the metal mask. however these approaches only work when the metal artifacts; are moderate and hence can be efficiently reduced by a CNN.

Lin et al. [11] are the first to introduce dual-domain network (DuDoNet) [17], which works by learning two CNNs on dual domains to restore sinograms and CT images simultaneously. Their intuition is that image domain enhancement can be improved by fusing information from the sinogram domain, and inconsistent sinograms can be corrected by learning signal back-propagated from the image domain to reduce secondary artifacts. Variants of the dual-domain architecture have been designed as a result of this work. In 2020 *Park et al.* [12] proposed an other dual domain network for MAR where the input of the two network are a stack of three consecutive image so that it was taken into account the volumetric dimension while in previous work each image was elaborated

independently from the other belonging to the same volume.
Moreover generative adversarial networks (GAN) [4] have recently been developed for addressing MAR problems, because of their ability to generate high-quality images.

through various types of image generation and specialized loss functions.All of the networks mentioned above, however, are supervised and rely on paired clean and metal-affected images. Because such clinical data is difficult to obtain, simulation data is commonly used in practice. As a result, supervised models may overfit to simulation and fail to apply well to real clinical data. Thus, learning from actual, real-world data is of relevance. To this aim, *Liao et al.* proposed a new method, ADN [5], which separates content and artifact in latent spaces using multiple encoders and decoders and induces unsupervised learning

2.2 Generative Adversarial network

Generative adversarial networks are an example of generative models: the term applies to any model that learns to represent an estimate of a distribution using a training set of samples collected from that distribution and, as a result, a probability distribution p_{model} is produced. In some circumstances, the model explicitly estimates p_{model}, in others the model can only generate samples from p_{model}. The generative adversarial network (GAN) gained popularity as a result of the study released by *Goodfellow et al* (2014) [18] and it is a strong learning framework based on game theory. The main idea behind GANs is to set up a game for two participants. One of them is known as the generator. The generator generates samples that are supposed to be drawn from the same distribution as the training data. The discriminator is the opposing player and it is a network that determines whether or not a picture is "real". It takes a picture x as an input argument and D(x) indicates the chance that x is an actual picture. If the output is 1, it indicates that the input has a confidence level of 100% that it is a real picture, whereas 0 suggests that it is impossible for the input to be a real picture. The game is divided into two situations [19]. In one scenario, training examples x are drawn at random from the training set and fed into the first player, the discriminator, which is represented by the function

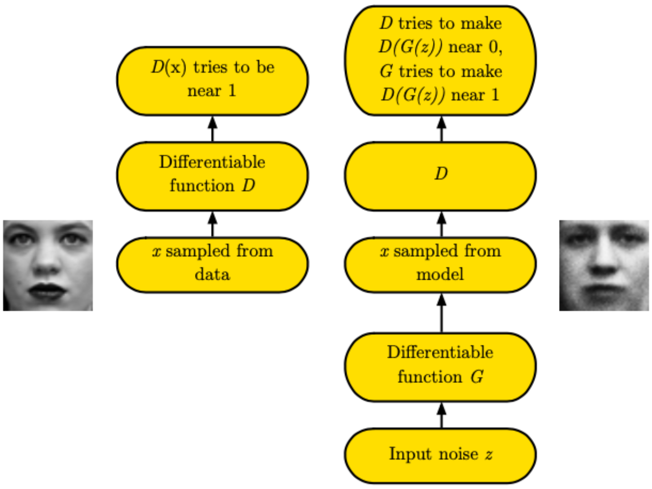

D(x) tries to be near 1
Differentiable function D
x sampled from data
D tries to make D(G(z)) near 0, G tries to make D(G(z)) near 1
D
x sampled from model
Differentiable function G
Input noise z

Figure 2.1: GAN framework [19]

D. The discriminator's aim is to output the probability that its input is real rather than false, assuming that half of the inputs it is ever presented are true and the other half are fake. The discriminator's aim in this first situation is for D(x) to be close to 1. In the second case, the generator's inputs z are drawn at random from the model's prior over the latent variables. The discriminator is then fed G(z), a fake sample generated by the generator. In this scenario, both players participate. The discriminator strives to make D(G(z)) approach 0 while the generative strives to make the same quantity approach 1.

The capacity to create a realistic sample and distinguish between genuine and fake samples is greatly enhanced in the training stage due to the continual competition between these two models. A high chance of success for the discriminative model over the generative model implies that the generative model needs to be improved. A high performance of the generative model, on the other hand, necessitates an improvement of the discriminative model. After the training, a balance between the generative and discriminative models is reached.

Formally, GANs are a structured probabilistic model containing latent variables z and observed variables x. The two players in the game are represented by two functions, each of which is differentiable both with respect to its inputs and with respect to its parameters. The discriminator is a function D that takes x as input and uses $\Theta^{(D)}$ as parameters. The generator is defined by a function G that takes z as input and uses $\Theta^{(G)}$ as parameters. Both players have cost functions that are defined in terms of both players' parameters. The discriminator wishes to minimize $J^{(D)}(\Theta^{(D)}, \Theta^{(G)})$ and must do so while controlling only $\Theta^{(D)}$. The generator wishes to minimize $J^{(G)}(\Theta^{(D)}, \Theta^{(G)})$ and must do so while controlling only $\Theta^{(G)}$. Because each player's cost depends on the other player's parameters, but each player cannot control the other player's parameters, this scenario is most straightforward to describe as a game rather than as an optimization problem. In this context, a Nash equilibrium is a tuple $(\Theta^{(D)}, \Theta^{(G)})$ that is a local minimum of $J^{(D)}$ with respect to $\Theta^{(D)}$ and a local minimum of $J^{(G)}$ with respect to $\Theta^{(G)}$.

Almost all different GAN model use the same loss function for the dicriminator $J^{(D)}$, while they differ for the cost used for the generetor $J^{(G)}$. The cost function that the discriminator needs to minimize is:

$$J^{(D)}((\Theta^{(D)}, \Theta^{(G)}))) = -\frac{1}{2}E_{x\sim\rho_{data}}(log(D(x)) - \frac{1}{2}E_z(log(1 - D(G(z))).$$

When training a classic binary classifier with a sigmoid output, this is just the usual cross-entropy cost that is minimized. The main distinction is that the classifier is trained on two minibatches of data: one from the dataset, where all examples have a label of 1, and another from the generator, where all examples have a label of 0. One option to decide the generator cost function is to consider a zero-sum game in which the sum of all player's costs is always zero: this means $J^{(G)} = -J^{(D)}$. In this case it is possible to summarize the entire cost function with a value function:

$$V(\Theta^{(D)}, \Theta^{(G)}) = -J^{(D)}(\Theta^{(D)}, \Theta^{(G)})$$

and to describe the game as a minimax game:

$$\min_G \max_D V(\Theta^{(D)}, \Theta^{(G)}) = \frac{1}{2}E_{x\sim\rho_{data}}(log(D(x)) + \frac{1}{2}E_z(log(1 - D(G(z))). \tag{2.1}$$

The generator loss in the minimax game (equation 2.1) performe well from a theoretical points of view but poorly in practice. While minimizing the cross-entropy between a target class and the expected distribution of a classifier is actually extremely effective (since the cost never saturates when the classifier produces incorrect results), maximazing the same quantity is not as desirable. Indeed in the minimax game, the discriminator minimizes a cross-entropy while the generator maximizes the same cross-entropy. This is not a good asset for the generator because when the discriminator correctly rejects generator samples with high confidence, the generator's gradient vanishes. One solution is to continue using cross-entropy minimization for the generator but instead of flipping the sign on the discriminator's cost to obtain a cost for the generator, we flip the target used to calculate the cross-entropy cost.The cost for the generator then becomes:

$$J^{(G)} = -\frac{1}{2}E_z(log(D(G(z))).$$

This means that, while in the minimax game, the generator minimizes the log-probability of the discriminator being correct, in this game the generator maximizes the log-probability of the discriminator being wrong.

The training process consists of simultaneous stochastic gradient descent. On each step, two minibatches are sampled: a minibatch of x values from the dataset and a minibatch of z values drawn from the model's prior over latent variables. Then two gradient steps are made simultaneously: one updating $\Theta^{(D)}$ to optimize $J^{(D)}$ and one updating $\Theta^{(G)}$ to optimize $J^{(G)}$. In both cases, it is possible to use the gradient-based optimization algorithm of your choice.

2.2.1 Conditional GAN (cGAN)

Many problems in image processing and computer vision can be considered as "translating" an input image into a corresponding output image, *Isola et al.* in his work [21] investigate conditional adversarial networks as a general-purpose solution to image-to-image translation problems. These networks not only learn the mapping from input image to output image, but also learn a loss function to train this mapping. This makes it possible to apply the same generic approach to problems that traditionally would require very different loss formulations. CNNs learn to minimize a loss function – an objective that scores the quality of results – and so there is still some manual effort into designing the right loss function; in other words we still have to tell the CNN what we wish it to minimize. Using GAN, indeed, allows us to specify a more high-level goal as “make the output indistinguishable from reality”, and then automatically learn a loss function appropriate for satisfying this goal. So, while GANs learn a generative model of data, conditional GANs (cGANs) learn a conditional generative model and, unlike an unconditional GAN, both the generator and discriminator observe the input edge map. Formally the main difference is that GANs learn a mapping from random noise vector z to output image y, $G : z \rightarrow y$, while cGANs learn a mapping from observed image x and random noise vector z, to y, $G : x, z \rightarrow y$. The loss for cGANs can be expressed as

$$L_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,z}[log(1 - D(x, G(x, z))], \tag{2.2}$$

where G tries to minimize this objective and D tries to maximize it. It has been found optimal to add a terms at the generator’s loss so that it is forced to not only fool the discriminator but also to generate output close to the ground-truth image, for example an L1 loss term can be added:

$$L_{L1}(G) = E_{x,y,x}[||y - G(x, z)||_1].$$

The final objective is then:

$$G* = arg \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G). \tag{2.3}$$

Even though L1 loss may produce blurry results on image generation problems and fail to encourage high- frequency crispness it also, in many cases, accurately capture the low frequencies. For this reason, while using this loss for the generator, it may be a good idea to restrict the GAN discriminator to only model high-frequency structure, relying on the L1 term to force low-frequency correctness . As a result, *Isola et al.* [21] construct a discriminator architecture called PatchGAN that only penalizes structure at the patch scale; in fact, it is sufficient to limit our attention to structure in local picture patches to model high-frequencies. This discriminator attempts to determine if each N N patch in an image is authentic or fake. We run this discriminator convolutionally through the picture, averaging all outputs to get the final D result. It can be demonstrated that N does not have to be the same size as the image, it can be much smaller, and in that case it has fewer parameters, runs quicker, and may be used to arbitrarily big images.

2.2.2 Least Squares GAN

Regular GANs hypothesize the discriminator as a classifier with the sigmoid cross entropy loss function, but it may happen that this loss function will lead to the problem of vanishing gradients. To address this issue, Least Squares Generative Adversarial Networks (LS- GANs) [21] might be employed, which use the least squares loss function for the discriminator. The generator's goal is to reduce this loss as much as possible:

$$L_{LSGAN}(G) = E_{z \sim \rho_{model}}[(D(G(z)) - 1)^2]. \tag{2.4}$$

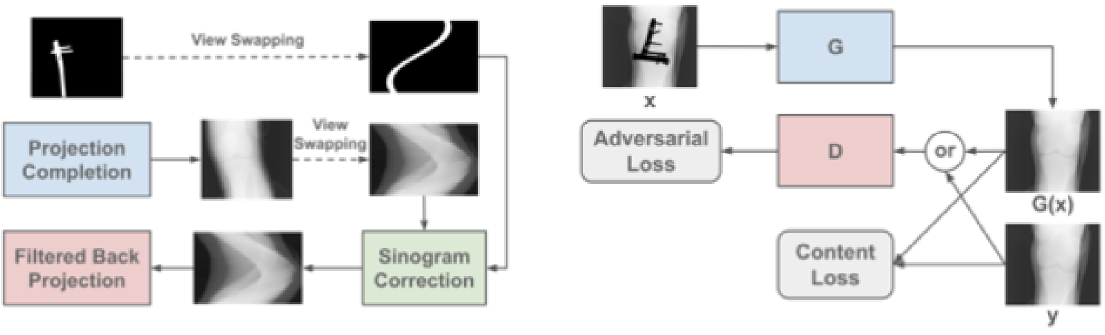
The principle is simple but powerful: while regular GANs cause almost no loss for samples that lie in a long way on the correct side of the decision boundary, the least squares loss function will penalize those samples even though they are correctly classified and doing so it can drive the false samples closer to the decision boundary. As a result of the penalization, the generator will generate samples closer to the decision boundary. Furthermore, for a successful GANs learning, the decision boundary should go across the manifold of real data, so that moving the generated samples closer to the decision boundary brings them closer to the manifold of real data. Lastly punishing samples that are far from the decision boundary might result in more gradients being generated when the generator is updated, which alleviates the problem of vanishing gradients. This enables LSGANs to function more stable during the learning phase.

2.3 Generative Mask Pyramid Network for MAR

A common method for reducing metal artifacts in CBCT scan is to replace the X-ray projection data inside the metal trace with synthetic data. Existing projection or sinogram completion methods, however, are not always capable of producing anatomically consistent information to complete the metal trace. To obtain better result *Liao et all.* propose a Generative Mask Pyramid Network [4] that use both projection-sinogram correction and adversarial learning to replace metal artifact affected regions with anatomically coherent content. They also present a mask pyramid network that enforces mask information throughout the network's encoding layers and a mask fusion loss that lowers early saturation for adversarial learning.

In this method MAR is considered as an image inpainting problem, so, while for image domain approaches there is a need for synthesized data that simulate the metal artifact image, in this case we simply apply random metal traces to mask out artifact free sinogram and train the network to recover the data within the metal traces.

As shown in figure 2.2 there are two major modules: a projection competition and a sinogram correction module: in both of them adversarial learning is introduced so that more structural and anatomically plausible information can be recovered from the metal regions. Furthermore it is implemented a mask-pyramid network (MPN), to use the mask geometry information at different scales, and a mask fusion loss to penalize early saturation.



(a) Method overview

(b) The base network

Figure 2.2

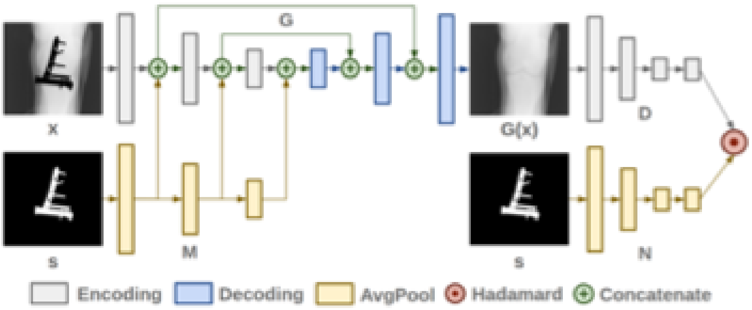
The projection completion module is an image-to-image translation model enhanced with a novel mask pyramid network. The projection completion module constructs anatomically realistic and structurally coherent surrogates inside the metal-affected regions given an input projection image and a presegmented metal mask. To refine the projection-corrected sinograms, the sinogram correction module predicts a residual map. This technique to joint projection-sinogram correction assures inter-projection consistency and makes use of context information between different viewing angles. It is worth noting that we execute projection completion first since it has been observed that projection pictures include more structural information, which aids in the learning of an image inpainting model.

2.3.1 The basis network

We structure the projection and sinogram correction problems within a generative image-to-image translation framework [21], inspired by recent advances in deep generative models . In Figure 2.3 the structure of the suggested model is presented. It is made up of two separate networks: a generator G and a discriminator D. G takes a metal-segmented projection x as input and produces a metal-free projection $G(x)$. The discriminator D is a patch-based classifier, which means that it is a type of discriminator that only penalizes structure at the scale of local image patches and that predicts whether or not the metal-free projection y or $G(x)$ is real or fake. Indeed the discriminator, like the PatchGAN [22] design, is built as a CNN with no fully-connected layers at the end to enable patch-wise prediction. The detailed structures of G and D are presented in the next sections.

Mask pyramid network

Metallic implants come in a variety of forms and sizes, such as metallic balls, bars, screws, wires, and so on. The projected implants have intricate geometries and they are different in each projections since the X-ray projections are taken at different angles. As a result, while in ordinary image inpainting problems the mask shape is generally simple and fixed, in this case it varies a lot and the the network must learn how to fuse such different mask information from the metallic implants. Indeed, directly using



(a) The base network

Figure 2.3

metal-masked image as the input requires that each layer encode the metal mask information and transmit it along to the later levels. This encoding may not function well for unseen masks, and so the mask information may be lost. To address this problem and obtain sufficient amount of mask information in each layer, a mask pyramid network (MPN) may be introduced into the generator to feed the mask information into each layer explicitly. In Figure 2.3 the architecture of the generator G with this mask pyramid network is illustrated: the MPN takes as input the metal mask and it is composed of as many block (denoted by l^i_M) as the encoding block of the generator (denoted by l^i_G). Each block l^i_M of M is implemented with an average pooling layer that has the same kernel, stride, and padding size as the convolutional layer in l^i_G. Due to this it is possible to associate a block of the MPN, l^i_M , with its correspondent encoding block in G. When l^i_M and l^i_G are coupled, the output of l^i_M will be concatenated to the output of l^i_G; this can be done because the MPN block have same kernel, stride, and padding size as the convolutional layer and so the metal mask output of l^i_M has the same size as the feature maps from l^i_G. Furthermore, due to this reason, l^i_M also takes into account the same receptive field of the convolution operation in l^i_G. In this way, the mask information is passed to the later layer and will be used by l^{i+1}_G. Furthermore the mask information need to be passed also to every decoder block of the generator. This can be done through the use of skip connection. Skip connections, as the name suggests, skips some of the layers in the neural network and feeds the output of one layer as the input to the next layers. In this case, similar as in the U-net, the concatenation of the output of l_M^i and l_G^i are passed as input to l_G^{-i}, which is the decoder block that have same input size as the output of l_G^i. For example the outputs of the first encoding block and MPN block are connected with the last decoding block while the outputs of the last MPN block and last encoding block are used as input for the first decoder. This, beyond passing the mask information also to the decoding part of the generator, makes the network use fine-grained details learned in the encoder part to construct an image in the decoder part.

As regard the discriminator there is still a mask pyramid network that have in each block an average Pooling layer with same kernel, stride, and padding size as the convolutional layer but, in this case, the output of the last MPN block is not coupled with the discriminator layer. The mask information is, indeed, used in the loss function where, thanks to an Hadamard product between the two outputs, only the part concerning the metal implant is considered.

Model details

We used the structure presented in [4] as a basis to develop small modification on some parameter to obtain better result. The complete generator structure can be seen in image 2.4. Each encoding block is composed of a Conv2D layer with kernel size 4, while the numbers between the brackets specify respectively the number of feature channels for each layer. For all of these Conv2D layers, the *stride* value has been set equals to 2 for each dimension and the padding has been set to *same*. In each encoding block the Conv2D layer is followed by a **BatchNormalization** layer and a **LeakyReLu** activation function. The AveragePooling layer, that take as input the mask, have same kernel size, stride and padding as the Conv2D layer they're convoluted with. The decoding block instead is composed of a UpSamping2D layer, with size 2 in each dimension, followed by a Conv2D layer with kernel size 3, *stride* 1 and *same* padding. The numbers between the brackets indicates the number of feature channels. Every Conv2D layer is followed by a **BatchNormalization** layer and a **ReLu** activation function. A *dropout* layer with 0.3 as *dropout parameter* has been inserted between the last three *Upsample* level to prevent the network from overfitting. The Last level have an UpSampling2D layer with size 2 in both dimension and a Conv2D layer with kernel size 3, *stride* 1, padding *same* and use the *tanh* activation function. The arrow in the figure represent the skip connections.

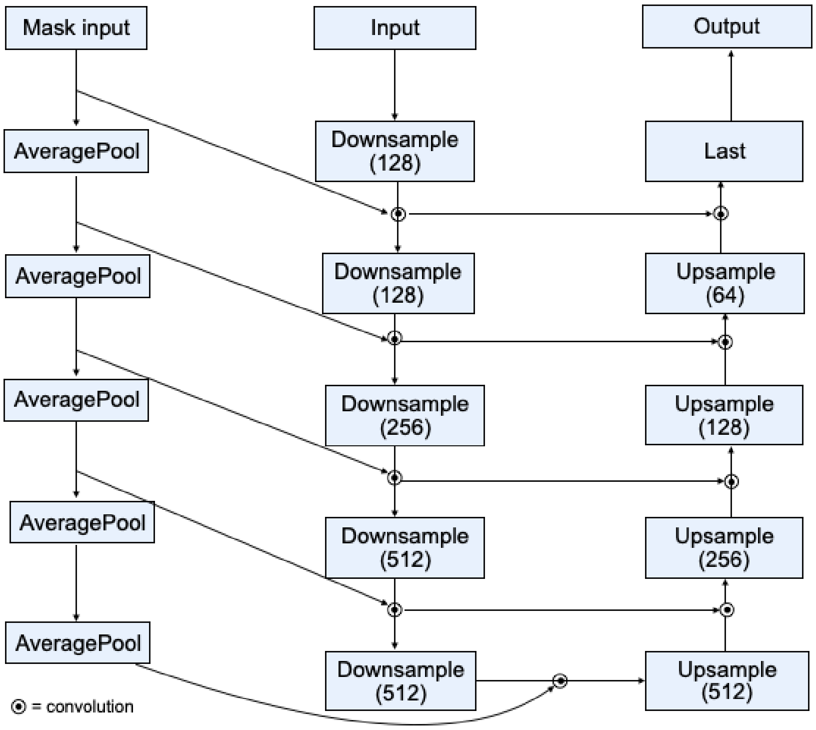


Figure 2.4: Generator structure

The discriminator structure is shown instead in Figure 2.5. The encoding block are the same as in the generator while the Last level is a encoding block that in the Conv2D layer have kernel size equals to 3, *stride* 1 and no padding. Also in the discriminator the AveragePooling layer have same kernel size, stride and padding as in the correspondent Conv2D layer.

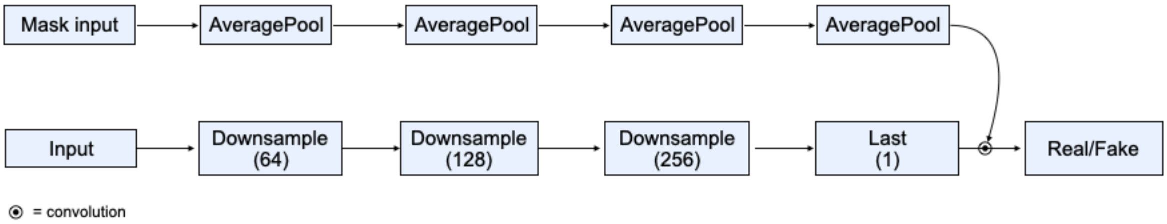


Figure 2.5: Discriminator structure

Implementation details

We implement the proposed model using TensorFlow with Google Colaboratory as the development platform for our neural network. Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs. We train the model with the Adam optimization method and for the hyper-parameters we set *learning rate*$= 0.0005$, $\beta_1 = 0.5$, $\lambda = 100$ and $batchsize = 16$.

2.3.2 Loss function

G and D are trained adversarially with LSGAN [21]. First of all we define the loss function without the use of the mask information:

$$\min_D L_{GAN} = E_y(||1 - D(y)||^2) + E_x(||D(G(x))||^2), \tag{2.5}$$

$$\min_G L_{GAN} = E_x(||1 - D(G(x))||^2). \tag{2.6}$$

Because we want the generator output G(x) to be similar to its metal-free counterpart y, we include a content loss to assure pixel-wise consistency between G(x) and y:

$$\min_G L_c = E_x, y(||G(x) - y||_1). \tag{2.7}$$

The loss is typically computed on the complete image in a traditional image-to-image architecture. The main problem with this approach is that a major amount of the generator's calculation will be spent retrieving previously known information and this makes the generator less efficient. Furthermore, because the generator does not know about the mask, this causes early saturation during adversarial training, in which the generator stops improving in the masked regions. We have two approaches to this problem. To begin, we just consider the content of the metal mask while computing the loss function. To do so we define an image that takes in the mask region the output of the generator and remain the same as the input outside the mask region:

$$\hat{y} = s \circ G(x) + (1 - s) \circ x,$$

where s is the mask and x is the input image. Then we can use this image to define a loss function:

$$\min_G L_c = E_x, y(||\hat{y} - y||_1). \tag{2.8}$$

Second, we modulate the output score matrix from the discriminator by the metal mask s so that the discriminator can selectively ignore the unmasked regions when deciding if an image is true or false. This is done through an other MPN, as described in the previus section. In this case, altohugh, we do not feed the intermediate outputs from MPN to the coupled blocks in D, but we apply the metal mask to the output. The adversarial part of the mask fusion loss is given as:

$$\min_D L_{GAN} = E_y(||N(s) \circ (1 - D(y))||^2) + E_x(||N(s) \circ D(\hat{y})||^2), \tag{2.9}$$

$$\min_G L_{GAN} = E_x(||N(s) \circ (1 - D(\hat{y}))||^2), \tag{2.10}$$

where $N(s)$ indicates the output of the MPN network in the discriminator.
The Hadamard products are correct because the MPN have as many block as the discriminator and each pooling layer have the same kernel, stride and padding as the convolutional layer so the output have the same size ad the discriminator output.
The total mask fusion loss is:

$$L = L_{GAN} + \lambda L_C, \tag{2.11}$$

where λ balances the importance of the adversarial loss and the content loss.

When training the model we tested different content loss function, for example we considered the L_2 norm, that is a technique where the sum of squared error is added into the loss function as a penalty term to be minimized. Indeed if the data is too complex to be modelled accurately the L2 may be a better choice than the L1 loss as it is able to learn inherent patterns present in the data. The fidelity term in this case is:

$$L_c = ||y - \hat{y}||_2.$$

2.3.3 The sinogram correction module

Although the previous sections suggested that the projection completion framework can yield an anatomically plausible outcome, it only analyzes the contextual information inside a projection. We notice that a collection of sinograms is formed by a stack of sequential projections. By making the completion results seem like sinograms, we utilize a simple yet effective model to guarantee inter-projection consistency.

Let x represent a sinogram created by the preceding projection completion step. As seen in figure 2.2, we can introduce a sinogram correction module that takes the sinogram obtained from the mask and x as input and generates the corrected sinogram. Then the result may be used to recostruct the metal-free volume. The framework used for this module is almost the same as the one used in the projection completion, we employ the same generator and discriminator construction as presented in figure 2.4 and in figure 2.5 respectively.

The main difference is that the generator predicts a residual map G(x), which is subsequently added to x to correct the projection completion results. For the objective function we use the same used in Equation 2.11 but with:

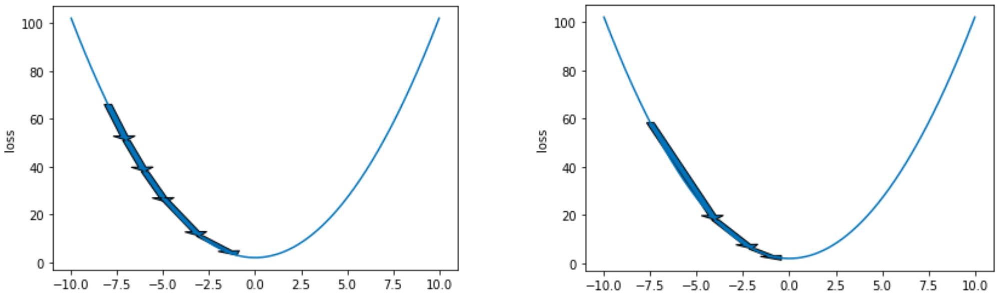
$$\hat{y} = s \circ (G(x) + x) + (1 - s) \circ x.$$

2.3.4 Learning rate schedule

In these models we set the learning rate in the training as constant at 0.0005 but, when training deep neural networks, it is often useful to reduce learning rate as the training progresses. This can be done by using pre-defined learning rate schedules. In this dissertation we considered the exponential decay that has the following mathematical form:

$$l_t = l_0 \times e^{-kt},$$

where t is the iteration step and k is an hyper-parameter, that we set equals to 0.5. We set as initial learning rate 0.0001 and we reduce it every 10 epochs (that is 3600 steps considering the size of our dataset and the batch size). Thanks to the learning rate scheduler the models converges faster and in a more accurate way as can be seen in figure 2.6. If we assume that the graph represent the loss function of our model, then it can be seen that, when the learning rate is constant, the convergence is slower and, if the learning rate we use is too big it may happen that the model does not converge. On the other side decreasing it while training may allows to reach the minimum in a faster way.



(a) *LR constant*

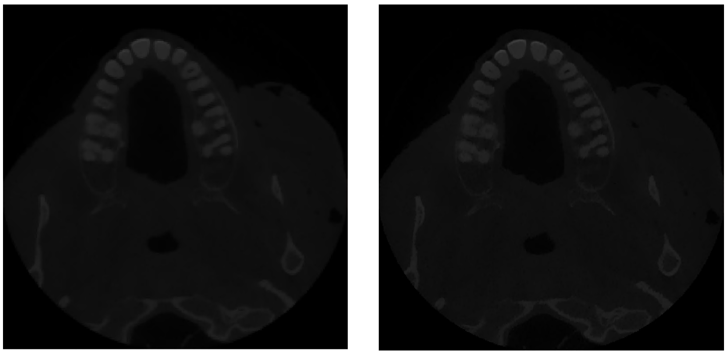
(b) *LR scheduler*

Figure 2.6: Example of convergence with and without learning rate scheduler.

2.4 Training and testing dataset

In this section we are going to present the dataset that was utilized to train the network. Due to the fact that we implemented a supervised network and that it is difficult to gather real pairings of with and without metal CT scans, we resolved to create a simulated dataset. This implies we'll simulate metal insertion into a CT scan without any metal implants. Because our network operates in the projection and sinogram domains, rather than the image domain, it is sufficient to treat it as an inpainting problem and insert binary masks into the correct projection, instead of simulating the effect of metal corruption in the image domain.

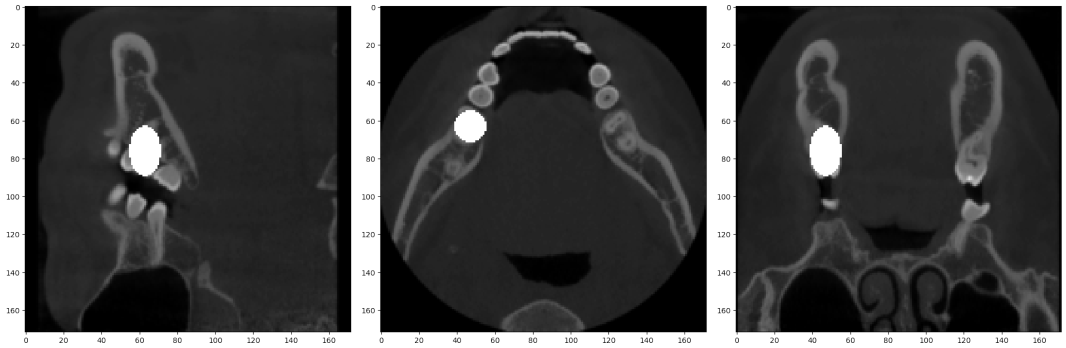
First of all we considered four volumes of dental scan reconstruction obtained with the CEFLA software. The first volume size is 440 × 440 × 416, while the others are 344 × 344 × 344. We decided to downsize these volumes to half their original size due to space and time limitations; as seen in Figure 2.7, this results in a loss of definition for the image and, as a result, for the data we will acquire. The volumes' ultimate dimensions are 220 × 220 × 208 and 172 × 172 × 172, respectively.



(a) *Volume1, resized* (b) *Volume1, original size*

Figure 2.7: Example of a slice of the original volume and resized one for volume 1.

To mimic metal insertion, we created some simulated implants. We used an ellipsoid to emulate an implant, which is obviously a simplification because implants have more complicated structures in reality. For each volume, we obtain four distinct implants for training and a few more in total for testing. Some of the simulated implants replicate



(a) (b) (c)

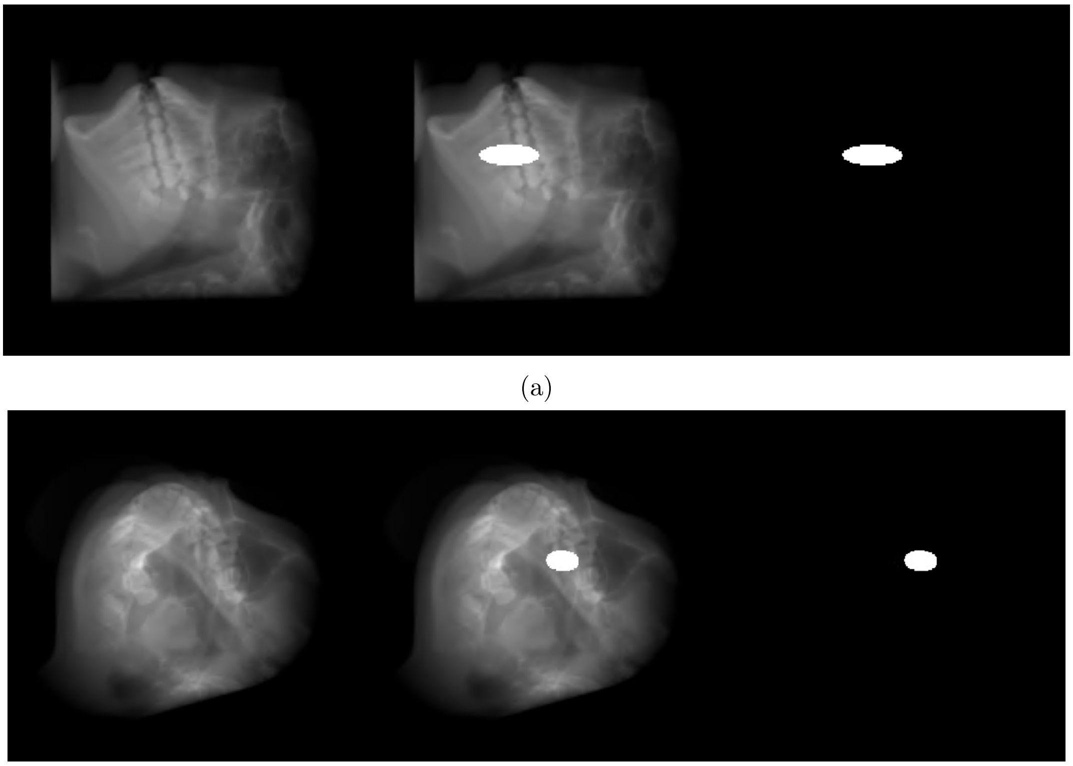
Figure 2.8: Example of slice of volume 4 with a simulated implants of size 5mmx8mm from different view.

the dimensions of real implants acquired from a CEFLA tool. Before saving the binary mask for this implant, we check that the position of the masks is consistent with the volume by inserting implants of that size and center in the volume. Figure 2.8 shows some examples of these images.

To forward project these volumes, in order to obtain the projections, we use the ASTRA Toolbox, that is a Python toolbox of high-performance GPU primitives for 2D and 3D tomography. It allows the user to perform basic forward and backward projection operations that are GPU-accelerated. First of all we define the geometry that will be used to perform the projection: it is a cone-beam geometry with the following parameter:

distance_source_origin	571mm
distance_origin_detector	408mm
detector_pixel_size	1.2mm
detector_rows	384
detector_cols	384
num_of_projections	360

We apply a forward projection with this geometry from the resized volume in order to obtain the metal-free projections. Then we project in the same way also the mask volume, and insert the binary mask into the metal-free projection to obtain metal-masked projections. Finally the input of the model will be an image composed of three correspondent projection image: the metal-free, the masked one and the mask. In total we have 16 volumes with the simulated implants, each one of them generates 360 projections images, which results in a the training dataset composed of 5400 images (see Figure 2.9)



(a)

(b)

Figure 2.9: Example of input images: two different projection angles for the same volume and implant.

Chapter 3

Numerical results

This chapter will be dedicated to the results of our experiments. We will show how different changes in the layouts of networks performed both on the training dataset and on some testing images. In order to evaluate the performance of the network and the accuracy of the metal artifact reduction we used the squared error metric (SE). It measures the sum of squares of the errors—that is, the summation of the squared difference between the estimated values and the actual value. In the following work we denote by $y \in \mathbb{R}^{n\times n}$ the ground-truth image (where n is the size of the image) and by $\hat{y} \in \mathbb{R}^{n\times n}$ the corrected one (which is an image that in the mask area have the output value of the generator and the input value in the remaining part).

Definizione 1. *The SE error is defined as:*

$$SE(y,\hat{y}) = \sum_{i,j=0}^{n} (y_{ij} - \hat{y}_{ij})^2. \tag{3.1}$$

3.1 Result of the basis network

In these experiments we use the basis network with no modifications. We first consider a volume with a large implant (volume_1) belonging to the training dataset.
In figure 3.1 we present some input projection image and, underneath them, the correspondent cropped corrected projection images, obtained using the basis network.

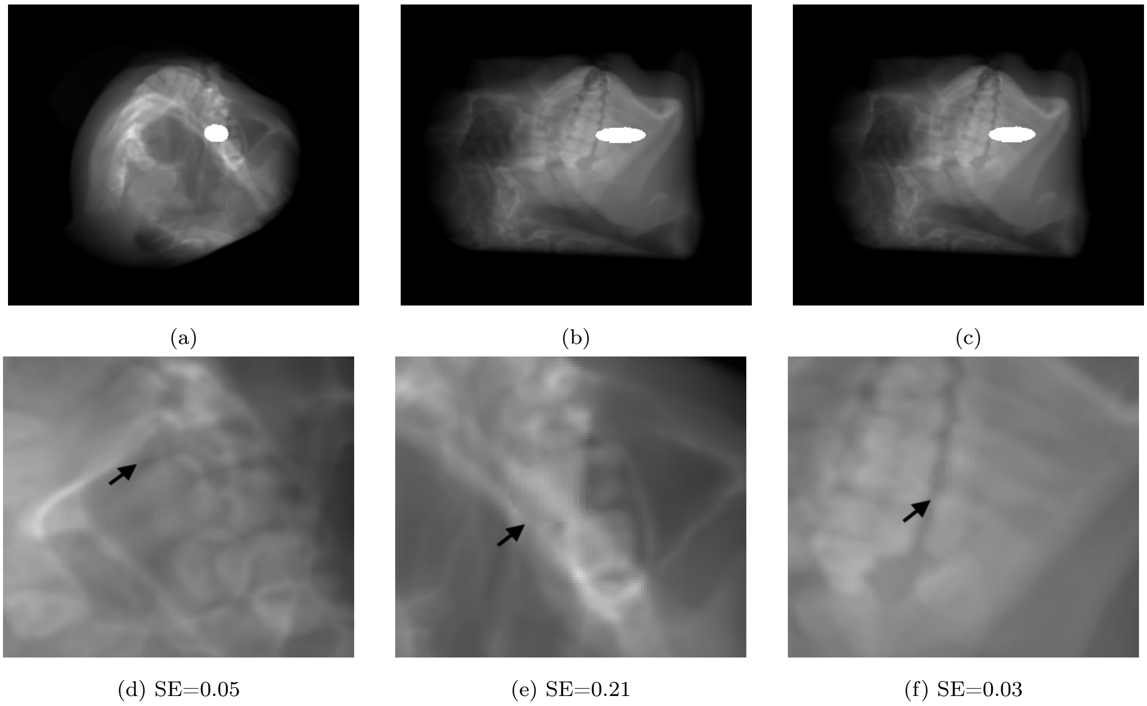
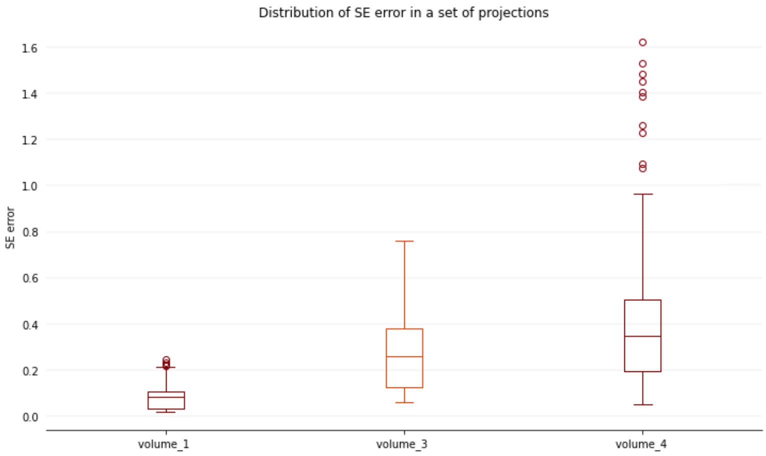


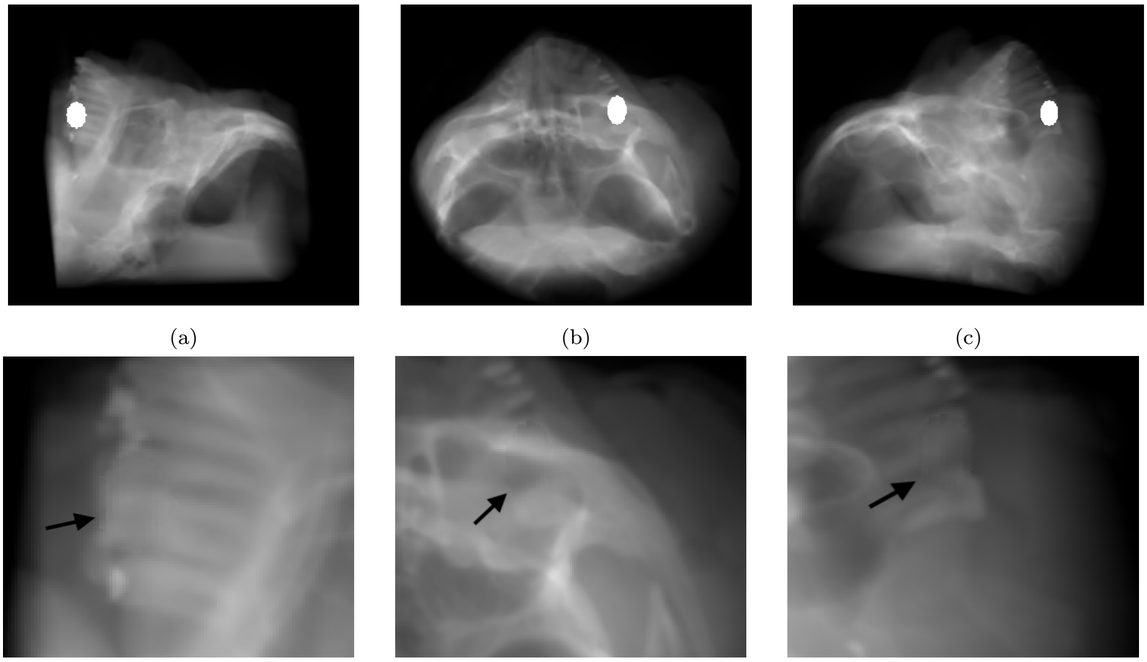
Figure 3.1: Examples of cropped corrected projection image of volume_1 with basis network and the correspondent input image.

Then we tested the model on new volumes, which don't belong in the training dataset. We considered first a volume (volume_3) with one implant that have a similar shape as one in the training dataset but that is positioned in a different position, in this case the mean SE value over the 360 corrected projection images is 0.28 and the result are shown in figure 3.3. Then we also considered an other testing volume (volume_4) with two implants instead (this set-up of implants wasn't considered in the training dataset even though there were volume with more than one implant), in this case the mean SE value is 0.38. In figure 3.4 the result over this set of projection image can be seen. We observe from the boxplot in figure 3.2 that the result are way worse than in the training dataset and, in particular, they became more inaccurate when the data become more different than the one used in the training. In the corrected images it can be seen that the border of the mask in the testing images are still visible even though the projection completion is realistic.



(g) *basis network*

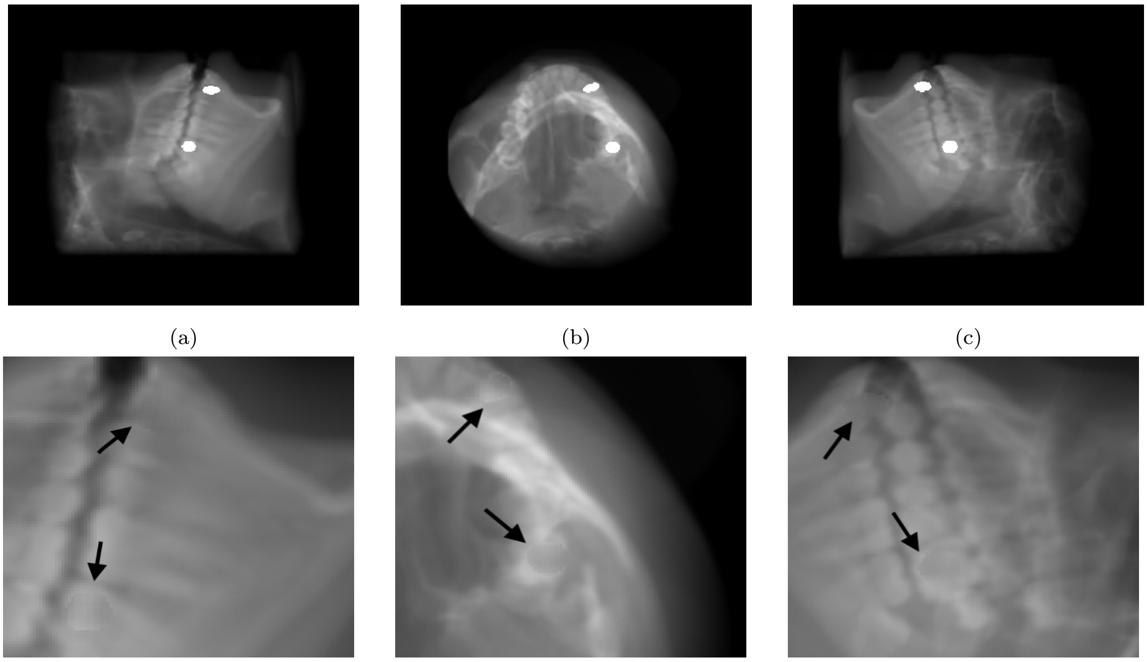
Figure 3.2: Boxplot of the SE error over the training and test dataset for the basis network.



(a) (b) (c)

(d) SE=0.23 (e) SE=0.31 (f) SE=0.34

Figure 3.3: Examples of cropped corrected projection for volume_3 with the basis network and the correspondent input image.


(a)

(b)

(c)

(d) SE=0.14

(e) SE=0.75

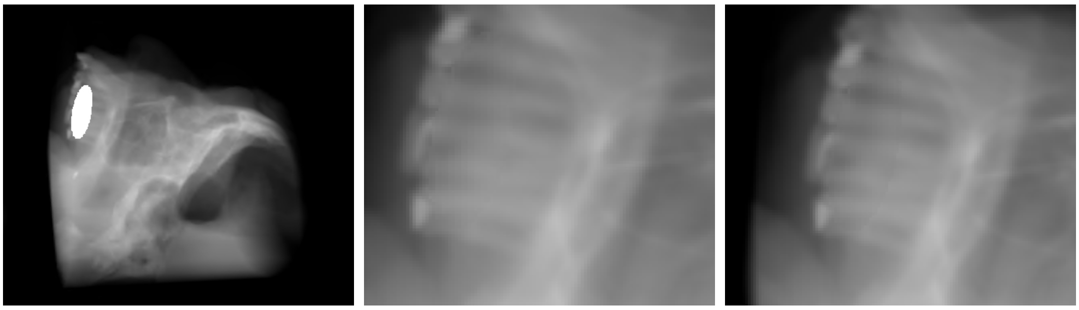
(f) SE=0.50

Figure 3.4: Examples of cropped corrected projection for volume_4 with the basis network and the correspondent input image.

3.2 Result of the L2 norm network

Because we expect the output to be close to its metal-free counterpart we have add an L1 fidelity term, but, as it can be seen from the previous result, the model performs accurately on training data but fails to perform well on test data: the problem of overfitting takes place. To this mean we thought about changing the fidelity terms with an L2 norm. We first evaluated this model in an other volume from the training dataset (volume_2), in figure 3.5 are shown the same projection image from volume_2 corrected with the basis network and with the L2 norm network.

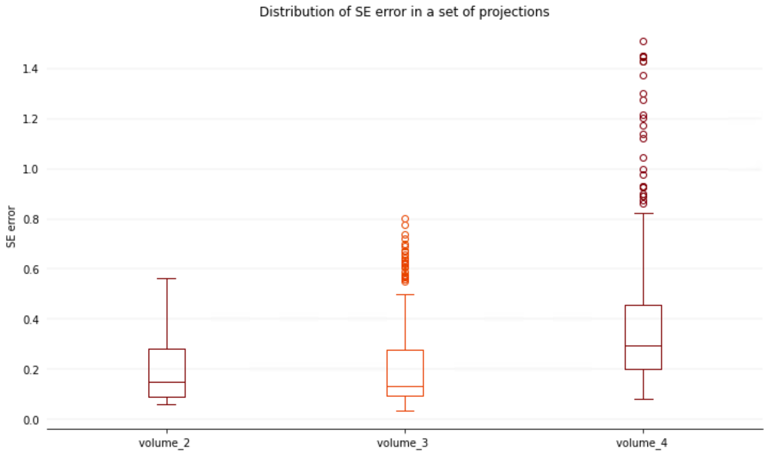
Then we considered the same test volume as in the previous example and we evaluate the same value. For volume_3 the SE error mean over the 360 projections images is 0.20, while for volume_4 the SE error mean is 0.37.



(a) input image (b) basis network, SE=0.12 (c) L2 norm network, SE=0.09

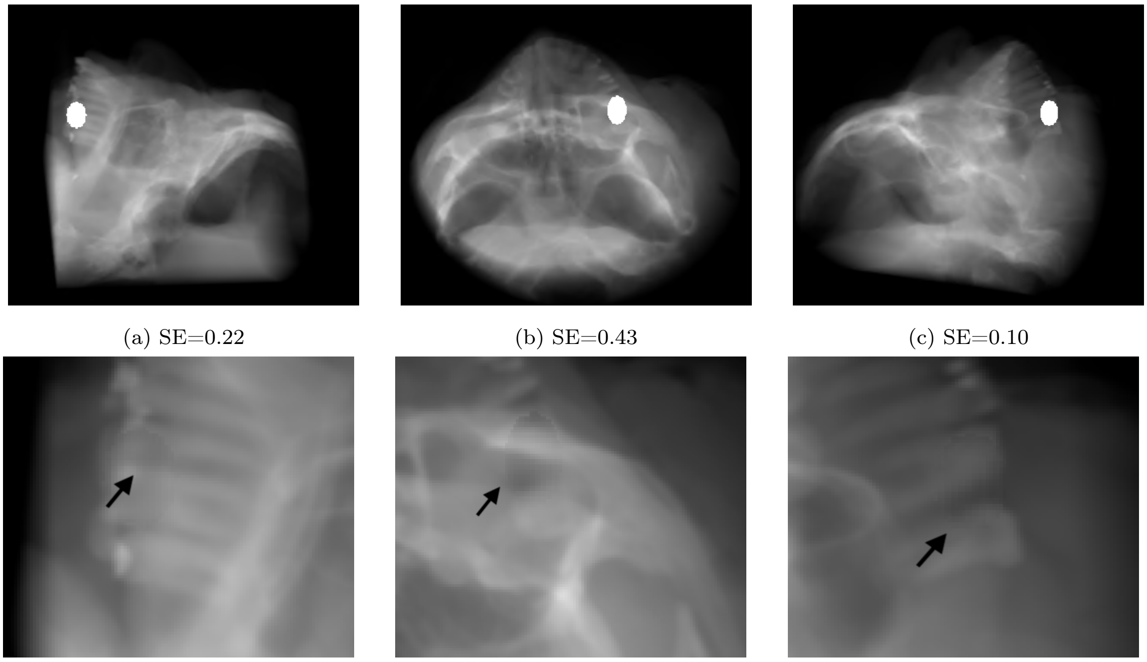
Figure 3.5

As it can be seen in the boxplot in figure 3.6, the result for volume_3 are way better than the one with the basis network and are similar to the result obtainied in the training volume (volume_2). This is probably because, even if it's not a data in the training, it's more similar to these than volume_4. In this case, instead, the model doesn't perform as well, even if a slightly improvement from the basis network can be seen. We show the results for these data in figure 3.7 and 3.8: for volume_3 we considered the same image as in the example before, to make the comparison more reliable, while we show new images for volume_4.



(d) *L2 norm*

Figure 3.6: Boxplot of the SE error over the training and test dataset for the L2 norm network.



(a) SE=0.22

(b) SE=0.43

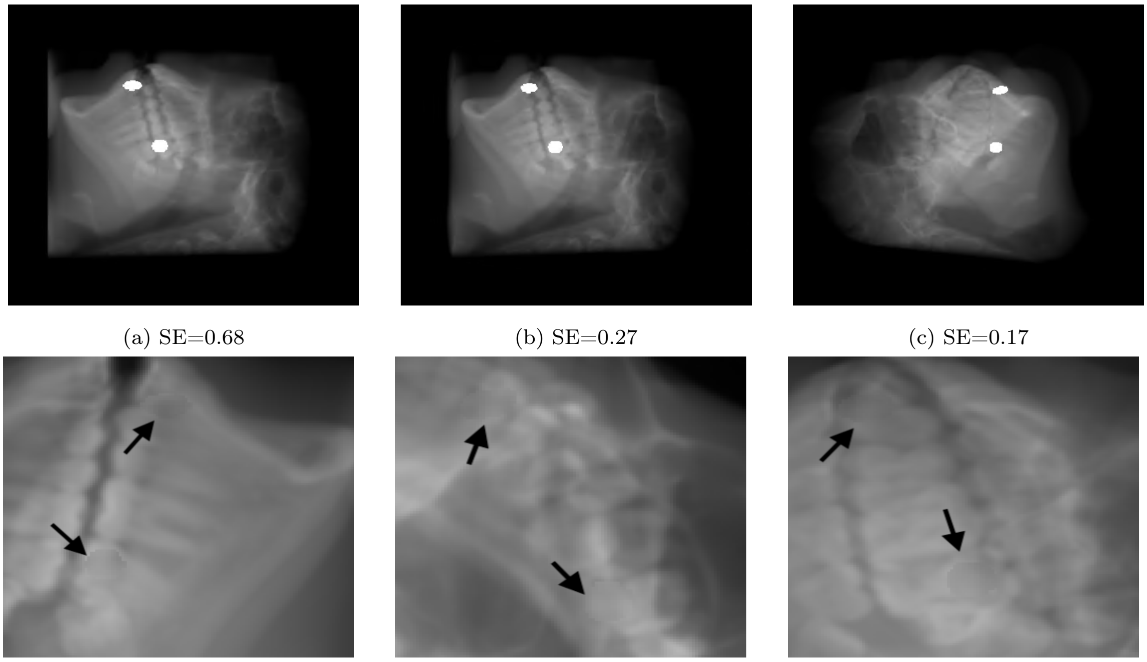
(c) SE=0.10

(d) SE=0.22

(e) SE=0.43

(f) SE=0.10

Figure 3.7: Examples of cropped corrected projection for volume_3 with the L2 norm network and the correspondent input image.



(a) SE=0.68

(b) SE=0.27

(c) SE=0.17

(d) SE=0.68

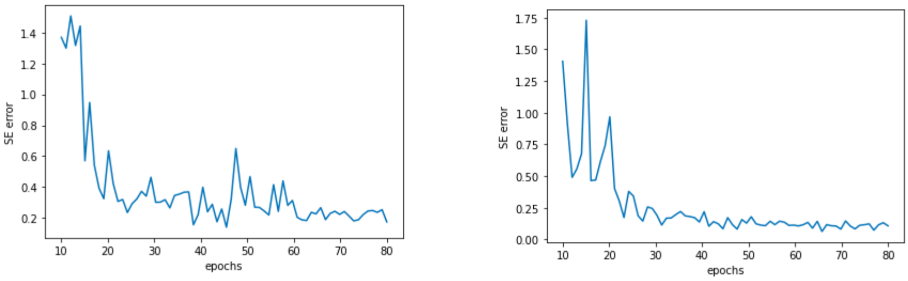
(e) SE=0.27

(f) SE=0.17

Figure 3.8: Examples of cropped corrected projection for volume_4 with the L2 norm network and the correspondent input image.

3.2.1 Result with the learning rate scheduler

Now we consider the benefit that the model obtain, when using a learning rate scheduler. In figure 3.9 we show the SE error decay during training for the L2 norm model with and without the learning rate scheduler, we decide to do not include the first 10 value in the plot, in order to obtain a more accurate scale in the *y-axis*, due to the really high values of the first errors. When the learning rate (LR) is constant the error initially decrease, but then fluctuate a lot without decreasing anymore; on the meantime, when the learning rate keep decreasing, the error is more stable and decrease faster, allowing in our specific case (with limited GPU time) to probably obtain better result.



(g) *LR constant* (h) *LR scheduler*

Figure 3.9: Plot of the error evaluated during training on the test dataset, value from epoch 10 to 80.

Training the L2 norm model with the learning rate scheduler we obtain a new model (L2norm_LR), that obtain in volume 1 (in the training dataset) slightly worse result that the previous network. To check if the model, while it obtained substandard result in the training dataset, is performing better on testing we consider the result on the test volumes. While for volume_4 the result are way better than in the model without learning rate scheduler, for volume_3 it performs slightly worst. In figure 3.10 we present the same cropped image for volume_4 as the one in Figure 3.7 but corrected with the L2norm_LR model.

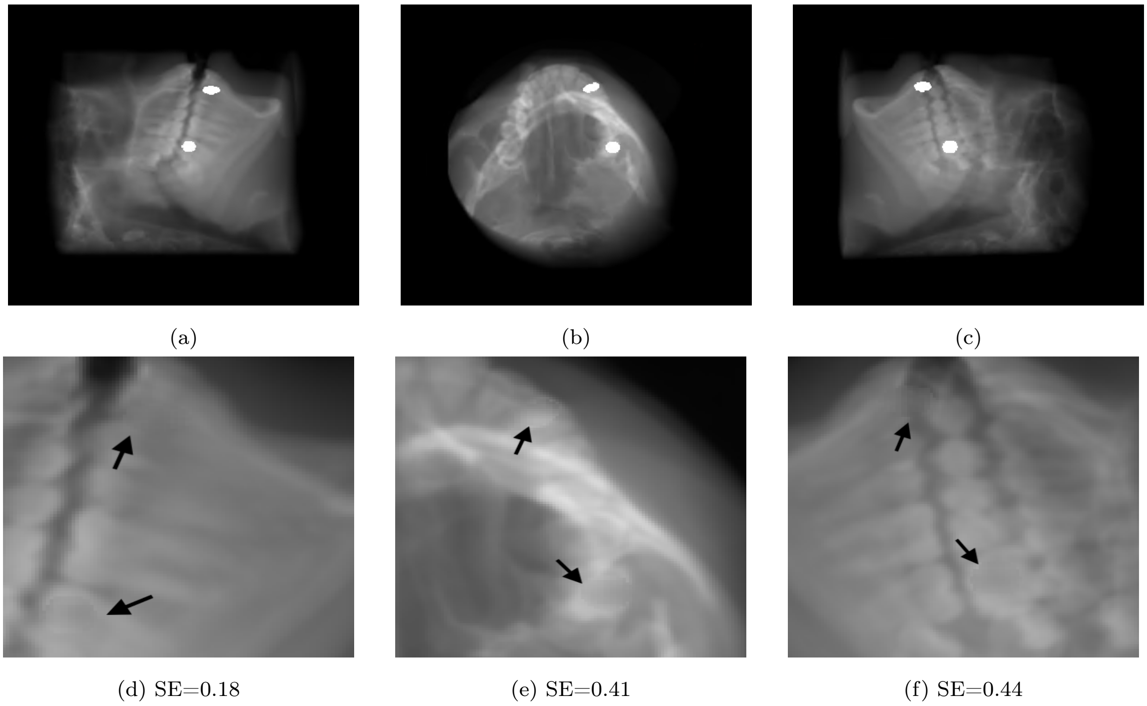
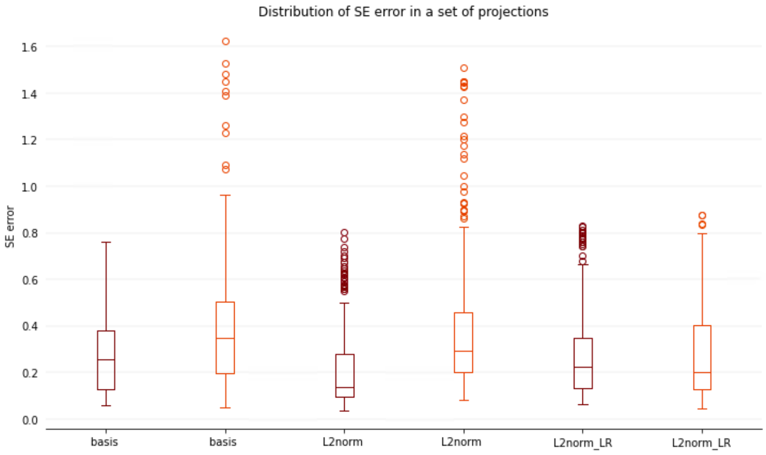


Figure 3.10: Examples of cropped corrected projection for volume_4 with the L2norm_LR network and the correspondent input image.

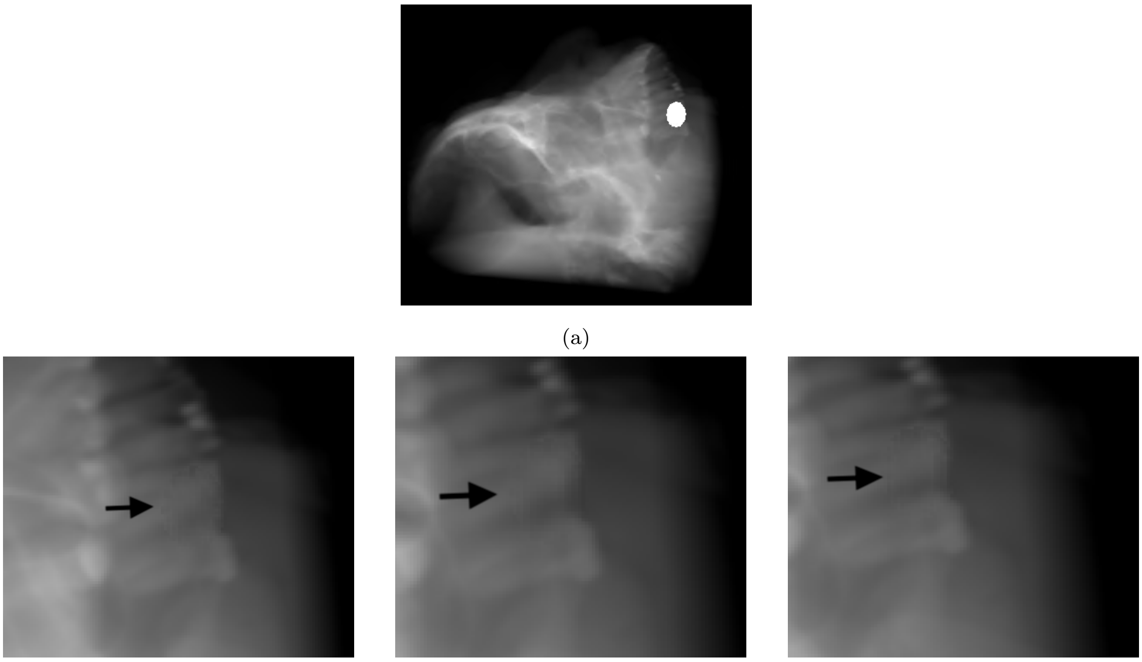
To make a more clear comparison between the various model we present in figure 3.12 and 3.13 the same projection image corrected with the three model presented till now, for both volume_3 and volume_4.

Lastly we present in Figure 3.11 some boxplots that represent the distribution of the SE error over the 360 projections images of the test dataset considered before, respectively for the basis network and the L2 norm model with and without learning scheduler. The boxplots in red represent volume_3, while the orange one volume_4. In this representation it's clearer that using the L2 norm gives better result for both data, while the use of the learning rate scheduler make the result between the two volume more similar, making the one for volume_4 better and the one for volume_3 a little worst. Furthermore, using the L2 norm, the distribution is skewed and the median is lower that the mean (for both volume), meaning that the the majority of the projections images have a relatively small SE error.



(g) *LR constant*

Figure 3.11: Boxplot of the SE error over test dataset for different models.



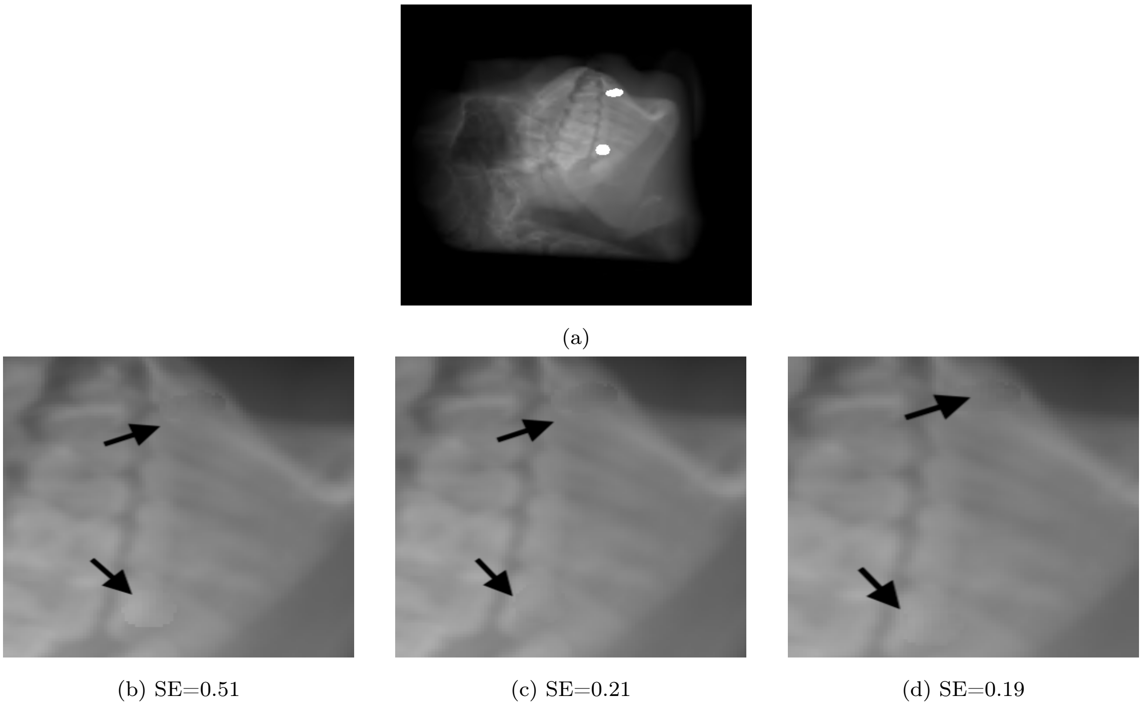
(a)

(b) SE=0.25

(c) SE=0.11

(d) SE=0.15

Figure 3.12: Examples of cropped corrected projection for volume_3 with the three presented network and the correspondent input image.



(a)

(b) SE=0.51

(c) SE=0.21

(d) SE=0.19

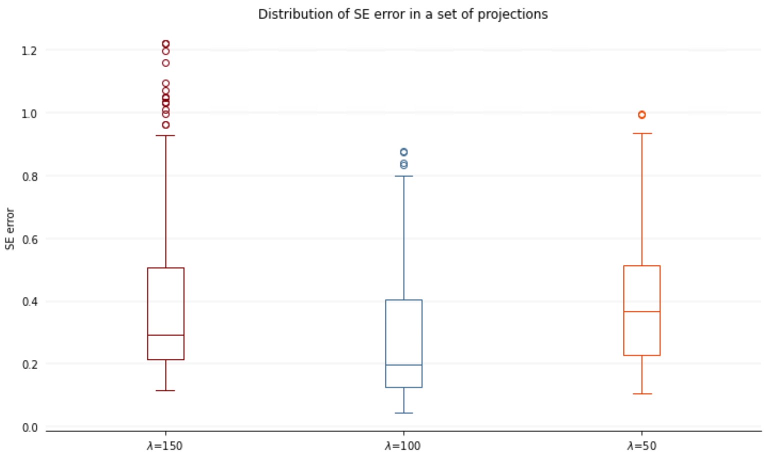
Figure 3.13: Examples of cropped corrected projection for volume_4 with the three presented network and the correspondent input image.

3.2.2 Setting the λ parameter

The λ parameter balances between the GAN loss and the fidelity term in fact:

$$L_{Gen} = L_{GAN} + \lambda L_C.$$

We first set it as 100, but we also tried different value to check if the model performed better with different ones. In image 3.14 we shows the SE error over the test volume for the L2 norm model with the learning rate schedule and different λ values. With $\lambda = 50$ the model perform quite poorly respectively to the others, even if the maximum and minimum are similar to the one with $\lambda = 150$, the median is higher, meaning that there are more projection image with a worse SE error. Although the model with $\lambda = 150$ is good it is still worse than the one considered before with $\lambda = 100$.



(e) *LR constant*

Figure 3.14: Boxplot of the SE error over test dataset for different λ value.

3.3 The sinogram correction module

When thinking about the training of the sinogram correction model the problem of data arise, indeed, to train the model, a new training dataset would be needed, so that the output from the projection network of these new dataset would be used as training dataset for the sinogram module. The main problem is that, while in the 360 projection images obtained from the same volume, the metal object, i.e the mask, is present in all of these images, when considering the sinogram the mask appears only in few images. Considering the size of our simulated implant, for example, they are present only in approximately 40 images over the 384 that forms the sinogram volume. This means that, if we intent to create a new dataset, of the same size of the one used to train the projection network, i.e 16 volumes, while in the first case the useful training images would be 5700, in this case they will be around 600 images, which is probably not sufficient to train efficiently this network. Indeed when training the model with these data, the problem of over-fitting would arise and the result would be good only in the data used in the training, while performing poorly on new data.

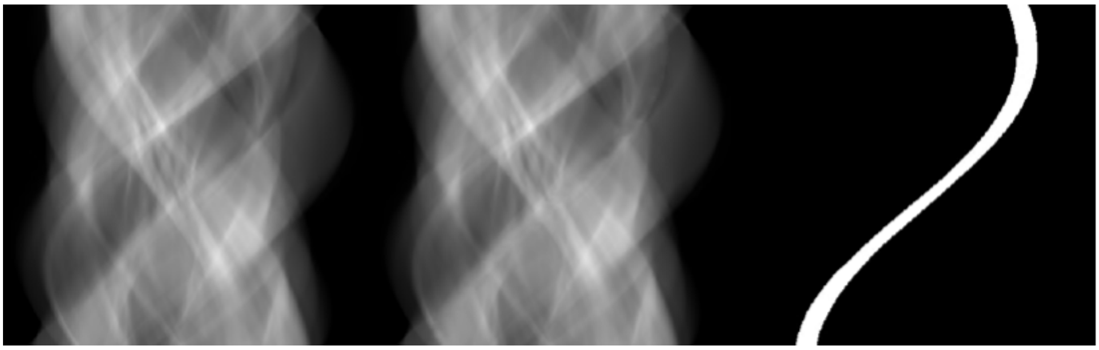
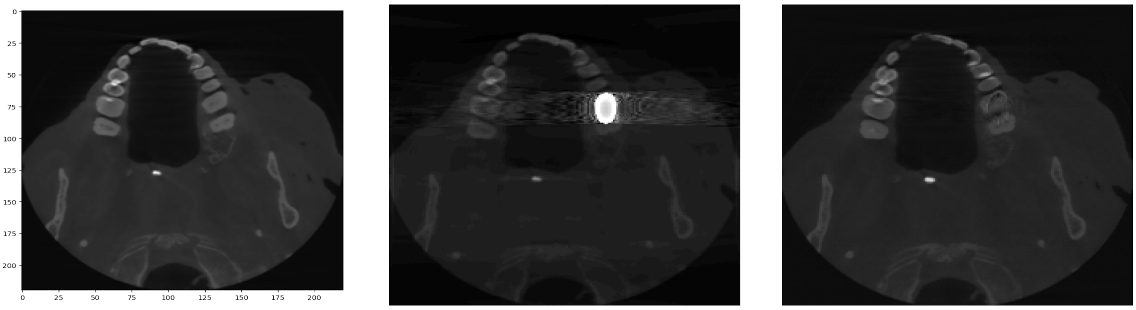


Figure 3.15: Example of the training image that would be used for the training of the sinogram correction module: the first image is the ground-truth image, the second one is a sinogram view obtained from the corrected projection and the last one is the mask.

3.4 Result on the reconstructed image

In the previous section we presented the result obtained in the projection domain, which is where our network operates. It can be usefull to present also the reconstructed images, to see how the model actually improves the reconstruction, when removing the metal mask. We use the AstraToolbox to reconstruct with CGLS3D_CUDA, which is is a GPU implementation of the CGLS algorithm for 3D data sets. It takes projection data as input and returns the reconstruction, after a specified number of CGLS iterations (in this case 50). It is important to notice that these results have the limitation to be obtained with the Astra Toolbox and not with a state-of-the-art reconstruction algorithm, like the one implemented in CEFLA.

In Figure 3.16 are presented the images obtained, after reconstruction, from the true volume projections, the ones with implants and the corrected ones, obtained using the L2norm_LR network,



(a) Reconstruction from the projection of the volume without implants

(b) Reconstruction from the projection of the volume with implants

(c) Reconstruction from the projection corrected by our network

Figure 3.16: Reconstructed images

Conclusions

In this paper, we discussed the subject of metal artifact reduction (MAR) in CBCT and some of the algorithms that have been developed to overcome it. We focused on the use of machine learning algorithms, specifically a generative adversarial network (GAN). We created a synthetic dataset of simulated projections from volumes with metal insertion generated with ellipsoids, to train this network. In addition, we developed several testing volumes to validate the results. We began with a model similar to the one in [4] and then added several alterations that result in the network's effectiveness, such as the L2 loss and the learning rate scheduler. Our results are promising, but they are limited by the small amount of samples utilized to train the networks as well as the hardware we used. Problems with hardware performance can be solved by adding extra memory or compute units to the device on which the network is being trained, but this technique might add complexity to the process and become economically extremely expensive. Furthermore, instead of simplification of ellipsoids, the training dataset can be expanded and a more realistic metal implant obtained. This could be an useful starting point for future developments, such as generating metal masks from CAD models of existing implants and inserting them in higher quantities of volumes, to add diversity to the dataset. Another essential aspect that might be thoroughly investigated is the sinogram correction module, which was not implemented in this work due to the previously mentioned constraint. This network can improve the realism of the outcome while avoiding the mask effect in the corrected projection. Furthermore, the ability to retrieve the reconstructed image using a more efficient reconstruction technique might help to better understand and correct the actual weaknesses and problems in this model. Furthermore, increased computer resources may aid in improving net performance by deepening the net structure. To summarize, we believe that the net implementation we proposed could serve as a solid starting point for further experimentation.

Bibliography

[1] Zhiquian Chang, "Prior-Guided Metal Artifact Reduction for Iterative X-Ray Computed Tomography", *IEEE Transaction on medical imaging*, vol 38, no 6, June 2019.

[2] M. D. Ketcha, M. Marrama, A. Souza, A. Uneri, P. Wu, X. Zhang, P.A. Helm and J. H. Siewerdsen, "Sinogram + image domain neural network approach for metal artifact reduction in low-dose cone-beam computed tomography", *Society of Photo-Optical Instrumentation Engineers (SPIE)*, vol 8(5), 10.1117, Sept/Oct 2021.

[3] Gao Liugang, Sui Jianfeng, Lin Tao, Xie Kai and Ni Xinye, "Metal Artifact Reduction Method Based on Noncoplanar Scanning in CBCT Imaging", IEEE, 10.1109 (14 Jan 2020)

[4] Haofu Liao, Wei-An Lin, Zhimin Huo, Levon Vogelsang, William J. Sehnert, S. Kevin Zhou and Jiebo Luo, “Generative Mask Pyramid Network for CT/CBCT Metal Artifact Reduction with Joint Projection-Sinogram Correction”, *arXiv 1907.00294v3*, 28 Nov 2019.

[5] Haofu Liao, Wei-An Lin, S. Kevin Zhou and Jiebo Luo, “ADN: Artifact Disentanglement Network for Unsupervised Metal Artifact Reduction”, *IEEE Transaction on medical imaging*, vol 39, no 3, March 2020.

[6] Anita Thakur, Vishu Pargain, Pratul Singh, Shekhar Raj Chauhan, P K Khare, Prashant Mor, “An efficient Fuzzy and Morphology based approach to metal artifact reduction from dental”, *International Conference on Computing, Communication and Automation (ICCCA2017)*, 2017

[7] Hiraku Iramina, Takumi Hamaguchi, Mitsuhiro Nakamura, Takashi Mizowaki and Ikuo Kanno, "Metal artifact reduction by filter-based dual-energy cone-beam computed tomography on a bench-top micro-CBCT system: concept and demonstration", *Journal of Radiation Research*, Vol. 59, No. 4, 2018, pp. 511–520, 10.1093, 30 April 2018

[8] Masoumeh Johari, Milad Abdollahzadeh, Farzad Esmaeili, Vahideh Sakhamanesh "Metal Artifact Suppression in Dental Cone Beam Computed Tomography Images Using Image Processing Techniques, *Journal of Medical Signals Sensors*, vol 8, issue 1, Jan-Mar 2018.

[9] Linlin Zhu , Yu Han, Lei Li, Xiaoqi Xi, Mingwan Zhu, And Bin Yan, "Metal Artifact Reduction for X-Ray Computed Tomography Using U-Net in Image Domain", *IEEE Transaction on medical imaging*, vol 9, July 2019, 10.1109.

[10] Yanbo Zhang and Hengyong, "Convolutional Neural Network Based Metal Artifact Reduction in X-Ray Computed Tomography", *IEEE Transaction on medical imaging*, vol 37, no 6, June 2018.

[11] Lequan Yu, Zhicheng Zhang, Xiaomeng Li and Lei Xing, "Deep Sinogram Completion With Image Prior for Metal Artifact Reduction in CT Images", *IEEE Transaction on medical imaging*, vol 40, no 1, Jan 2021.

[12] Dongyeon Lee, Chulkyu Park, Younghwan Lim and Hyosung Cho, "A Metal Artifact Reduction Method Using a Fully Convolutional Network in the Sinogram and Image Domains for Dental Computed Tomography", *Journal of Digital Imaging*, 10.1007, 2020.

[13] Yuanyuan Lyu, Jiajun Fu, Cheng Peng and S. Kevin Zhou, "U-DuDoNet: Unpaired dual-domain network for CT metal artifact reduction", *arXiv 2103.04552v1*, 8 Mar 2021.

[14] Sathyathas Puvanasunthararajah, Davide Fontanarosa, Marie-Luise Wille, Saskia M. Camps, "The application of metal artifact reduction methods on computed tomog-

raphy scans for radiotherapy applications: A literature review”, *Journal of Applied Clinical Medical Physics*, 10.1002, 30 Mar 2021

[15] Lars Gjesteby, Bruno De Man, Yannan Jin, Harald Paganetti, Joost Verburg, Drosoula Giantsoudi and Ge Wang, “Metal Artifact Reduction in CT: Where Are We After Four Decades?”, *IEEE Transaction on medical imaging*, vol 8, Oct 2016.

[16] Esther Meyer, Rainer Raupach, Michael Lell and Bernhard Schmidt, "Normalized metal artifact reduction in computed tomography", *Ameri- can Association of Physicists in Medicine*, 10.1118, 2010

[17] Wei-An Lin, Haofu Liao, Cheng Peng1 Xiaohang Sun, Jingdan Zhang, Jiebo Luo, Rama Chellappa and Shaohua Kevin Zhou, "DuDoNet: Dual Domain Network for CT Metal Artifact Reduction", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019

[18] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y, "Generative adversarial networks", *arXiv: abs/1406.2661*, 2014

[19] Ian Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks", *arXiv:1701.001604v*, 30 Apr 2017

[20] Ge Wang, Yi Zhang, Xiaojing Ye, Xuanqin Mou, "Machine Learning for Tomographic Imaging", IPEM–IOP Series in Physics and Engineering in Medicine and Biology, 2020

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", *arXiv:1611.07004v3*, 26 Nov 2018

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang and Stephen Paul Smolley, "Least Squares Generative Adversarial Networks", *arXiv:1611.04076v3*, 5 Apr 2017

[23] Julia F. Barrett and Nicholas Keat, "Artifacts in CT: Recognition and Avoidance", *RadioGraphics*,2004

[24] F. Edward Boas and Dominik Fleischmann, "CT artifacts: Causes and reduction techniques", *Imaging Med.*, 2012

[25] Henrik Turbell, Cone-Beam Reconstruction Using Filtered Backprojection, *Linkoping UniTryck*, February 2001

Ringraziamenti

Giunta alla conclusione di questo mio percorso di studi ritengo doveroso ringraziare tutte le persone che hanno contribuito a renderlo unico. Innanzitutto ringrazio la professoressa Elena Loli Piccolomini ed il mio tutor in CEFLA, Marco Soldini, per essere stati sempre presenti e disponibili ed avermi sempre supportato.

Ringrazio inoltre mio babbo, per avermi fatto appassionare alla matematica da piccola, e mia mamma, per avermi insegnato la perseveranza e l'impegno che sono stati necessari a studiarla davvero, la matematica. Vorrei, inoltre, ringraziare mia sorella per avermi sempre ricordato che non serve prendere tutto così seriamente ed il mio ragazzo, Marco, per aver sempre creduto in me, più di quanto facessi io.

Infine volevo ringraziare tutti i miei amici per essermi stati sempre vicini in questo percorso: i miei amici del "mare", per le grigliate insieme e per avermi fatto sorridere quando l'umore era a terra, gli amici di sempre (Robi, Sofi, Tommy), per essere una bellissima costante nella mia vita, e le fantastiche persone conosciute in erasmus, per l'esperienza unica vissuta insieme. Un ringraziamento speciale, però, agli amici del dipartimento di matematica, perchè senza le giornate di studio insieme in "auletta", le cene al Ranzani ed il nostro supportarci l'un con l'altro non so come avrei affrontato questi anni. Non ho nominato tutti personalmente ma, lo sapete, ognuno di voi ha avuto un ruolo fondamentale in questo mio percorso, e vi ringrazio tutti profondamente.