December 2020

Deep Learning-based Reconstruction of Volumetric CT Images of Vertebrae from a Single View X-Ray Image

Mingren Xiang
University of Wisconsin-Milwaukee

DEEP LEARNING-BASED RECONSTRUCTION OF VOLUMETRIC CT IMAGES OF VERTEBRAE FROM A SINGLE VIEW X-RAY IMAGE

by

Mingren Xiang

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

December 2020

ABSTRACT

DEEP LEARNING-BASED RECONSTRUCTION OF VOLUMETRIC
CT IMAGES OF VERTEBRAE FROM A SINGLE VIEW X-RAY IMAGE

by

Mingren Xiang

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Professor Zeyun Yu

Computed tomography is often used in medical fields today because it creates more detailed information for doctors than regular X-ray images. However, one major side effect is that patients may be exposed to a large dose of radiation because it takes hundreds of X-ray images to compute a CT scan. Another shortcoming is that patients are required to lay down on the CT machine for the scan, which is usually not the ideal position when diagnosing spine related issues such as cervical spondylosis and lumbar disc herniation. The prime motivation for this study is to reconstruct CT images using only one or a few X-ray images by using deep learning models trained to map projection radiographs to the corresponding 3D anatomy. My work demonstrates the feasibility of the approach with 20 Dicom sets of human vertebrae. The training set of the deep learning model consists of pairs of information, where each pair is made up of a 3D volume and a manually generated radiograph. The deep learning model for this study is CNN (Convolutional Neural Network) based encoder-decoder framework. The encoder converts high-dimensional data into embedded feature maps whereas the decoder reconstructs high-dimensional 3D output we desire. After training, the network can take in single or multiple 2D x-ray images and output an array of intensity values that represent a 3D CT image. MATLAB 3D viewer is used to visualize the result. We performed 50 experiments, averaging 3 model training for each experiment. The results generated by the model have an acceptable accuracy but there is a lot of room for improvement. The best PSNR (Peak Signal-to-Noise Ratio) value we obtain is 17.34 dB. While a state-of-the-art 3D reconstruction usually has a PSNR value above 30 dB. In addition, this paper summarizes the challenges and limitations that my teammates and I faced. I will also introduce methods that the team used to overcome these barriers. Since this is still an ongoing research project, the team will continue the work on improving the result. The end goal is to apply this study on real medical cases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis project benefited from conversations with many people in the Big Data Analytics and Visualization Laboratory at the University of Wisconsin- Milwaukee. Professor Zeyun Yu gave me lots of high-level advice from his years of experience. I would like to offer my gratitude to my teammate Mrinal Kanti Dhar, who works with me on this project. I would also like to thank our lab group for providing their GPU, their knowledge, help, and their insightful suggestions.

1 Introduction

1.1 CT Reconstruction from X-rays

CT scans can provide accurate three-dimensional information on the size and position of the target volume, and the position of any critical organs or structures of interest. As the name implied, computed tomography is an imaging modality that reconstructs a 3D volume from a set of X-rays, ranging from hundreds to thousands of 2D images captured in a full rotation of the X-ray apparatus around the body, This approach effectively transfer information from a 2D plane to a 3D view. This gives computed tomography some key advantage over regular X-rays. Because X-rays project every piece of information on a 2D plate. While bones are visible, soft tissues are often difficult to find. While in a CT 3D view. Everything from the bone to tissues is visible because there is no information overlap between them. That is why often computed tomography is used in medical fields today to accurately diagnosing diseases such as cancer because CT provides more detailed information and creates better views for doctors than regular X-ray images.

While CT technology remains to be one of the groundbreaking diagnosis tools in medical fields. Inevitably there are a couple of shortcomings. One major side effect is that patients may be exposed to a large dose of radiation because it takes hundreds of X-ray images to compute a CT scan. Another shortcoming is that patients are required to lay down on the CT machine for the scan, which is usually not the ideal position when diagnosing spine-related issues such as cervical spondylosis and lumbar disc herniation. Also, CT is expensive, for both the patient and the hospital. According to a survey conducted by The Fiscal Times, the starting price for a CT scanner begins from

$65,000 for a refurbished one that will only give you small images quickly. A larger and brand-new CT scanner can go as high as $2.5 million [1]. Thus, hospitals and medicals centers in the undeveloped area like African countries might not afford a CT machine. One solution to overcome these disadvantages of CT scan is to digitally reconstruct a 3D volume out of one or a few X-ray images, as taking X-ray images is much cheaper and requires less exposure to the radiation. The X-ray machine can also take pictures of the patient in any angels. If the reconstruction is successful, we can provide the doctors CT with 3D volumes without taking an actual CT scan. This is the prime motivation for this study

The goal is to reconstruct CT images using only one or a few X-ray images. 3D reconstruction using sparse 2D data is always been a challenging task because when you project information from 3D to 2D. Information will inevitably be lost in the process. So usually it takes a large set of projections from different angles to reconstruct a CT volume to make up for the loss. The variety of angles of the projection is the key to the accuracy of the reconstruction as one 2D projection from a certain angle can only capture the limited amount of information of the original 3D volume. For opaque surfaces, 3D reconstruction with ultra-sparse 2D projection is nearly impossible since the information outside the projection angle will be completely lost and unknown to us. X-ray images, however, are different than opaque surfaces since the projection is transparent. As figure 1 shown below

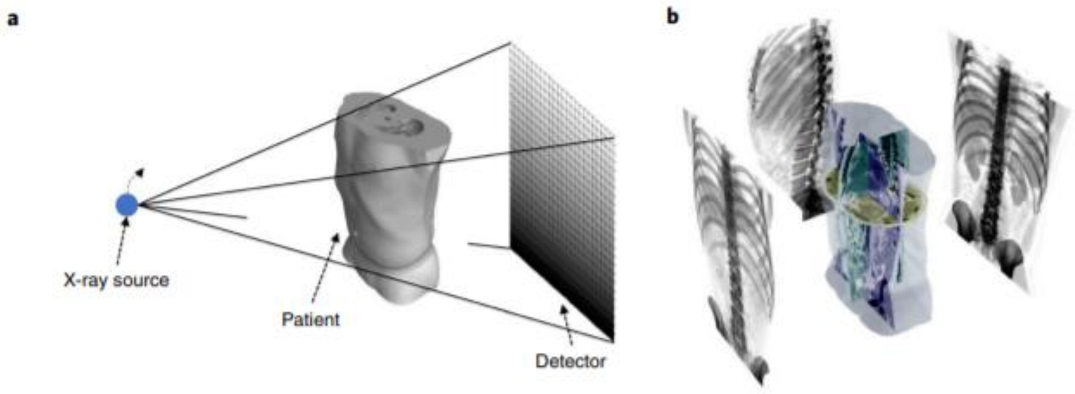

Figure 1: X-ray imaging principle and Results [2]

If we take an X-ray image from an arbitrary angle. The images we get are transparent so information outside the projection angles is also present in this projection. Unlike an opaque surface, transparent projections map every information of the original 3D to a 2D plane. The key is to find out the relationship of the mapping so we can reconstruct the 3D volume using just one projection. This is a very suitable task for a deep learning network, which is why we choose to rely on deep learning models to help us learn the mapping relationship,

1.2 Deep learning

With the exploration of data volume and faster computing power, deep learning has been widely used to replace the knowledge-based application with pre-defined logic. With a dataset that has a large enough volume, deep learning has shown state-of-the-art results after sufficient training and tuning. The key objective for a deep learning model is to find the minimum value of the loss function, defined by us according to the end goal of the project. Usually, the loss function measures how far away from the model's prediction to the true answer, called ground truth or targets. The method is called deep learning as the model consists of a deep neural network with millions of variables on it. The term “learning” means we are constantly tuning these variables on the neural network to get closer to the ideal result. We use training datasets to drive these tuning processes of the variables on the network. This process is known as “learning ” Training datasets usually consist of pairs of information. The input and the ground truth. By providing a large amount of training data, the model will learn the relationship between your input and ground truth. At first, the model will give random results but as the training goes on. The accuracy of the model will be higher. The common metrics to measure the training performance is training loss and training accuracy. Once training is done, we move on to the testing phase. in the testing phase. We provide testing inputs that the model never encounters before. When we got an output from the model using the testing input. We then evaluate the performance of the model.

As we know, finding the relationship of mapping between 3D volume and 2D projection could be a very challenging job. In recent years, more and more research group have turned their focus on training deep learning models to do 3D reconstruction. Many groups achieve results with very high accuracy. The next section will introduce previous work that inspires this thesis. We will briefly discuss the method they use and the results they obtain. Then the reaming chapter will give details information on the approach we take and discuss the results we got. This thesis will demonstrate that training deep learning models to reconstruct a 3D volume with ultra-sparse 2D projection is a feasible attempt given a deep neural network that has a large number of filters and a very large dataset to train on. In general, deep learning can help us solves two types of problem. Classification and regression. Classification is to classify a given input into two or multiple predefined categories. While in the regression problems, there are no predefined categories. Regression gives you an output based on your input. There are relationships that you can map between input and output. Such as linear regression. The relationship could be mapped by a linear equation. So, our problem falls into the fields of regression. Because The key is to our problem is for the deep learning model to learn the mapping between 2D X-ray to its 3D voxel value and position. Unlike simple regression problems, the mapping is much more complex and cannot be represented by a set of functions. The following section would introduce and analyze the relationship between our input and the prediction.

2 Previous Work

In this section, we will introduce previous work done by other researchers in the past who inspire our research on the topic of 3D CT reconstruction. Section 2.1 will list some of the traditional methods and section 2.2 focus on reconstruction done by training various deep learning models with different datasets.

2.1 Traditional Method on CT reconstruction

As discussed in the introduction section, 3D reconstruction from ultra-sparse 2D images is near impossible. It was not until recent years when researchers start to have some ground-breaking results in this field. One of the earliest works on CT reconstruction on single 2D projections uses statistical shaped analysis [3]. Novosad et al [4] and. Lamecker et al [5] both explored to use a statistical shaped model to reconstruct CT images using very few X-ray projections. The core logic of their works is an algorithm that tries to optimize a similarity measure/ This measure is meant to assess the difference between projections of the X-ray images and the shape of the 3D volume. As Novosad et al [4] described, they tried to measure the distance between the silhouettes of the object in the projections according to their observations from the experiments. In 2014, Karade and Ravi [6] prosed a new algorithm to reconfigure a 3D template surface mesh model to match the bone shape in orthogonal radiographs. The algorithm is also based on a statistical shaped model. Karade and Ravi then introduce Laplacian surface deformation trying to enhance their 3D model template and obtain a better result. All of these previous works provide very accurate results. But one common limitation among this traditional method is that a deep and large amount of knowledge of the 3D shapes and silhouettes of the object is required. If the shapes and silhouettes are lost or changed. Then the result will be skewed. So, this method might be ill-conditioned. Furthermore, the result is very sensitive to the quality of the input data. For example, the model can perform well with a normal piece of the femur but if the femur is fractured or deformed. Then the accuracy will drop dramatically that the results obtained will not be useful in real-life medical practices. Reconstruction using deep learning models can overcome this limitation if enough fractured or deformed examples are included in the training dataset with the normal bones.

2.2 CT Reconstruction using Deep Learning

Both Deep Learning [11] and CT reconstruction with ultra-sparse 2D X-ray are relatively new fields in the computer graphics community. Yet deep learning is taking over as the dominating method for research in the computer graphics community such as image classification, object detection, computer vision, and 3D reconstruction. As the deep learning community grows rapidly, there are new networks published by researchers every day such as U-Net [12] and ResNet [13]. These networks serve a different purpose but one common feature among all of them is that they are all CNN (Convolutional Neural Network) based network There are three key aspects of CNN, namely sparse interaction, parameter sharing, and equivalent representation [13]. Figure 2 shows an example of the CNN network.

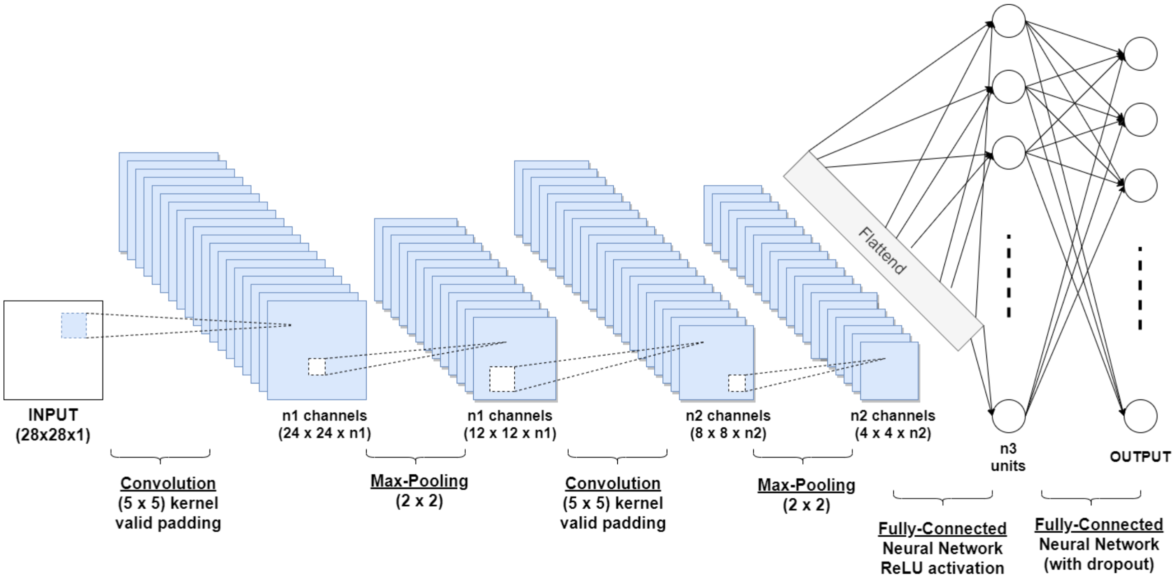

Figure 2: CNN Architecture

One of the key advantages of CNN compare to a fully connected network is that CNN can achieve a higher volume of parameters with less spatial and computational recourses, which is very important for the image-related task since constructing a deep learning network for such a task is usually computationally expensive.

Previous work on 3D reconstruction with deep learning construct different models to fit their data. One common feature among all of the work published is that the deep learning network follows a based encoder-decoder framework, where the encoder converts high dimensional data into feature mappings where information is embedded in the projections. The decoder converts the feature maps back to 3D shape so the output of the network is the volume we desired Wang et al [7] introduced a network to enhances the resolution of the 3D volume. The group constructed a hybrid framework that combines two CNN based network. The first one being 3D encoder-decoder Generative Adversarial Network (3D-ED-GAN). The second one is a Long-term Recurrent Convolutional Network (LRCN). The 3D-ED-GAN is used to construct the overall 3D shape and the goal of LRCN is to construct and finalize the details. While the work is not reconstructing 2D from 3D. The work serves as a proof of concept that an encoder-decoder framework can work with a large volume of 3D data and generating a high-resolution result. Henzler et al [8] introduced another CNN-based encoder-decoder framework that uses skip connection [12] and residual learning [12]. The group used cranial 2D X-rays and 3D CT of various mammalian species as training data. Only one projection is needed for the network to construct a promising result. Xingde et al promised a solution to construct 3D volume from 2 X-ray of human Chest images. They named the network X2CT-GAN. The unique approach of X2CT-GAN is that for each input data, a separate encoder is used rather than stacking input together as one, which is the mainstream way of dealing with multiple inputs. The result presented is very accurate, but a separate encoder is needed for every input. The network ends up being huge and it requires large commuting resources for training.

Among all previous work examined by us, Shen et al [2] at Stanford University have the best result, the PSNR (Peak Signal-to-Noise Ratio) [15] value is above 30 dB while only using a single projection as the input. Our project is largely based on the approach they were taking. We implement the network introduced in this paper and used our dataset. More details will be introduced in the proceeding sections

3 Data Preparation

3.1 Raw Data

Data preparation is the most important step in a deep learning project. As the quantity and quality of the dataset will dictate the performance of the model. We started this project using two Oral CT set as a proof of concept. One for training and another one for testing We quickly realized that a much bigger dataset is needed to have more robust results. The raw data we eventually choose is 10 sets of lumbar spine CT images provided by my advisor, Dr. Zeyun Yu. These CT sets are all DICOM format. Below is one of the raw data we use. The visualization is done by ImageJ 3D volume viewer.

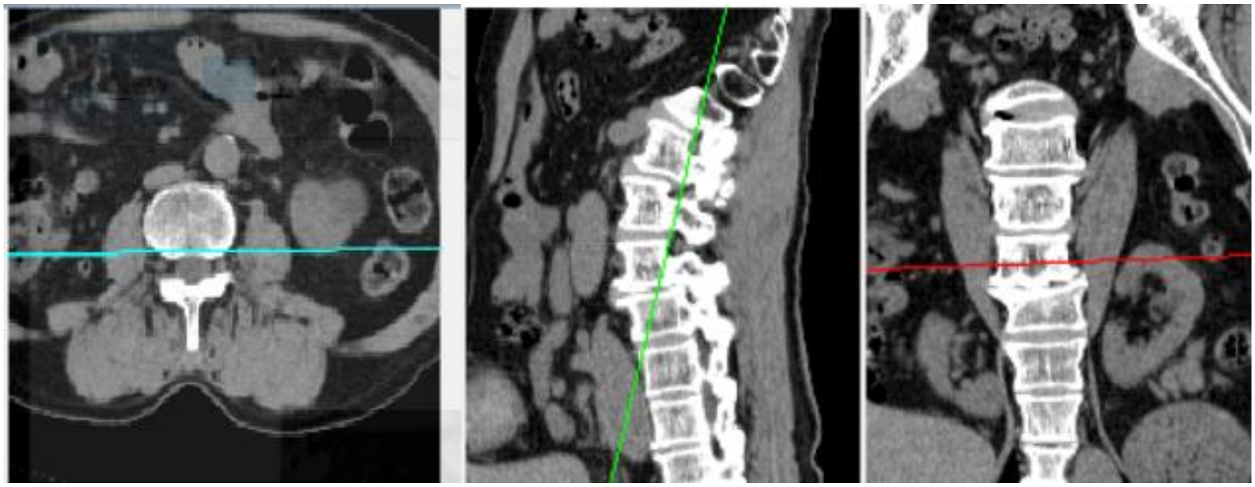

Figure 3: lumbar spine visualization from different view angles

Figure 3 shows the different views of one of the lumbar spine datasets. The left one is the view along the z-axis facing the XY plane. The middle picture is the view along the y-axis facing the XZ plane and the one on the right is the view along the x-axis facing the YZ plane. A typical range of the intensity of this dataset is from -10 to 2500. The smaller the intensity value, the darker it is on the image. We can observe that bone structures have a much higher intensity than the tissues and organs thus appears to be very bright in the images. This is ideal for us since our goal is to reconstruct bone structure so images with strong contrast are a good start. The challenge of this dataset and any other CT images is that intensity values are not evenly distributed. There are more data points in the XY plane and far fewer data points along the z-axis. So, the spacing between the voxels along the z-axis turns out to be bigger than the distance on the XY plane. Section 5 will introduce the method we use for post-processing to solve this issue.

At the beginning of this project, we tried to use the original raw data for training but quickly found out that data preprocessing is needed to fit our needs since we have limited resources and computing power to do the training. There are two main barriers to stop us from using the original DICOM data

- The raw lumbar spine images contain too many details for us to reconstruct. Since we only have 10 datasets. It's not a feasible approach to train a deep learning model to learn how to reconstruct the whole lumbar spine with such limited datasets.
- The original DICOM set is too big to fit in our network for training. The single DICOM slice usually has a size of 600 KB. There are at least 250 slices in one DICOM CT set, which means if we were to use the original CT volume as the ground truth for training. Each ground truth will be 150 MB. Given we usually trained hundreds of input pairs for training. Our GPU simply cannot handle this data stream.

The solution for these barriers is data segmentation where we break down the original CT set to smaller and management pieces. The details will be introduced in section 3.2

3.2 Data Segmentation as the Ground Truth

We know our ROI (Regin of Interest) is the spine. More specifically, the individual vertebrae on the spine. So, we decide to segment the individual vertebrae out as a new volume for ground truth. This decision marked the fundamental steps for this project. The approach to break down the original CT set into smaller and management pieces effectively solves the two barriers mention above and give us other advantages

- By segmenting the vertebrae, we illuminate the unwanted details in the original images. The new volume we get has far less noise from tissues and organs as we just focus on the vertebrae itself
- The size of the new volume is significantly smaller than the original volume.
- We effectively create more training data by segmenting the vertebras, as each dataset contains at least 5 vertebras to work with. We increase our training ground truth from 10 to 50

To implement this idea. Our first approach is to write scripts to automate this process. The idea is we sample data points on the spinal canal. The canal itself has a lower intensity value in the image. It’s also located in the center of the spine sounded by bones, so it is easy to locate it. Once data sampling is done. We apply 3D cubic spline [16] to sketch the curve of the canal. Then with every data point on the curve, we create a plane that’s in the direction of the tangent line of the curve. Then compute the average intensity on the curve and plot the result as a graph. The result is shown in Figure 4

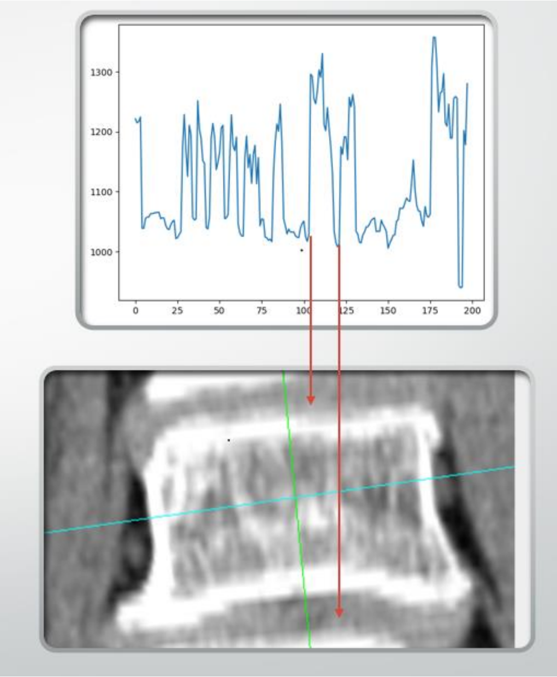

Figure 4: intensity change along the curve to indicate vertebra position

As Figure 4 shows, you can find vertebras in between the two local minimum of the graph. Once the location is found, we can cut the vertebras along the direction of the tangent line of the curve. This idea is significant since the human spine is not a straight line. Capturing the intensity along the curve is necessary for us to get accurate results.

For timing reasons, the above implementation plan was not complete in time so we eventually choose to segment the vertebra manually. As shown in Figure 5, we manually choose our region of interest and create arrays to store the intensity values of the individual vertebras. We repeat this manual process for every dataset and eventually get 50 vertebras as our ground truth. Now that the ground truth is ready. The next section will introduce how we obtain X-ray images out of these CT volumes to use as the training input

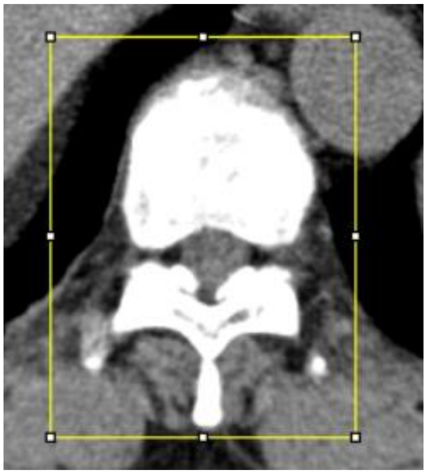

Figure 5: Manual segmentation

Figure 7 shows the 3D view of the result we get. We use these vertebrae as our ground truth for the training. The visualization is implemented by MATLAB volume viewer

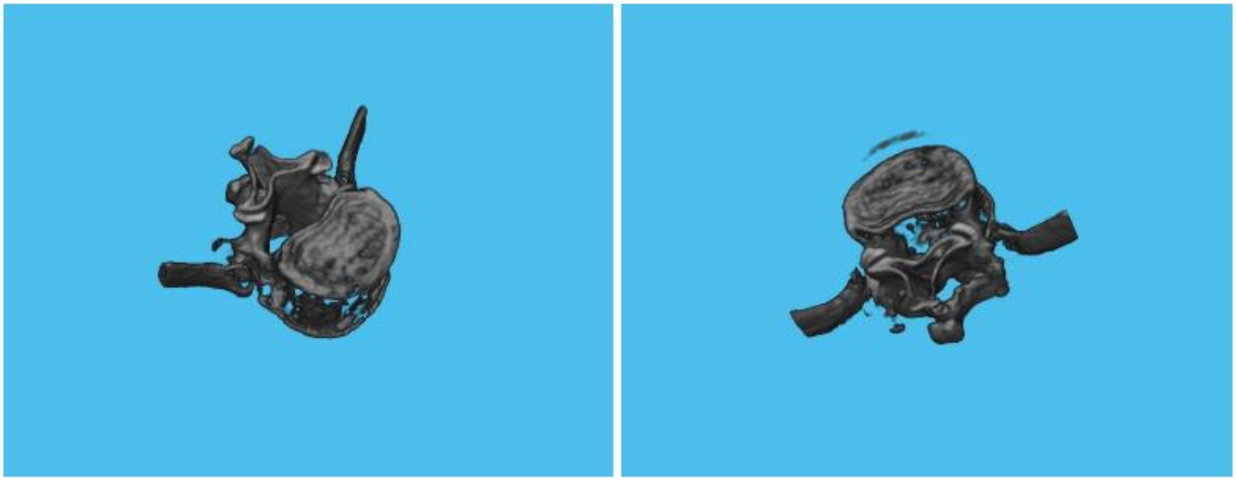

Figure 6: Vertebra visualization

3.3 2D Projection as the training input

Since the only data we have at hand is CT volumes, we must manually compute X-ray images out of these CT images to use as the training input. Though 3D reconstruction from 2D is a changing task, the reverse process is very straight forward. Milickovic et al [10] introduced a ray-tracing method to compute DDR (digitally reconstructed radiographs). The common method for computing projection using ray tracing is to either choose the maximum intensity or the average intensity along the ray and project the value as the 2D radiographs. We use MATLAB package to implement an average ray-tracing method to compute our X-ray images. We also use MATLAB to rotate the 3D object while keeping the sources of the average ray fixed. By doing that we can obtain projections from any angles, Figure 7 shows some of the X-ray images we computed. We have to resize the X-ray to 128 * 128 and convert the images to a PNG file with an intensity range from 0 to 255 to fit in our training networks. The reason will be explained the Section 7.2 limitations. Also, since the distribution of data points in the 3D volume is not even, the quality of the X-rays various depending on what angles we take the projection. Usually, we find X-ray images projected along the x-axis tend to have the best quality

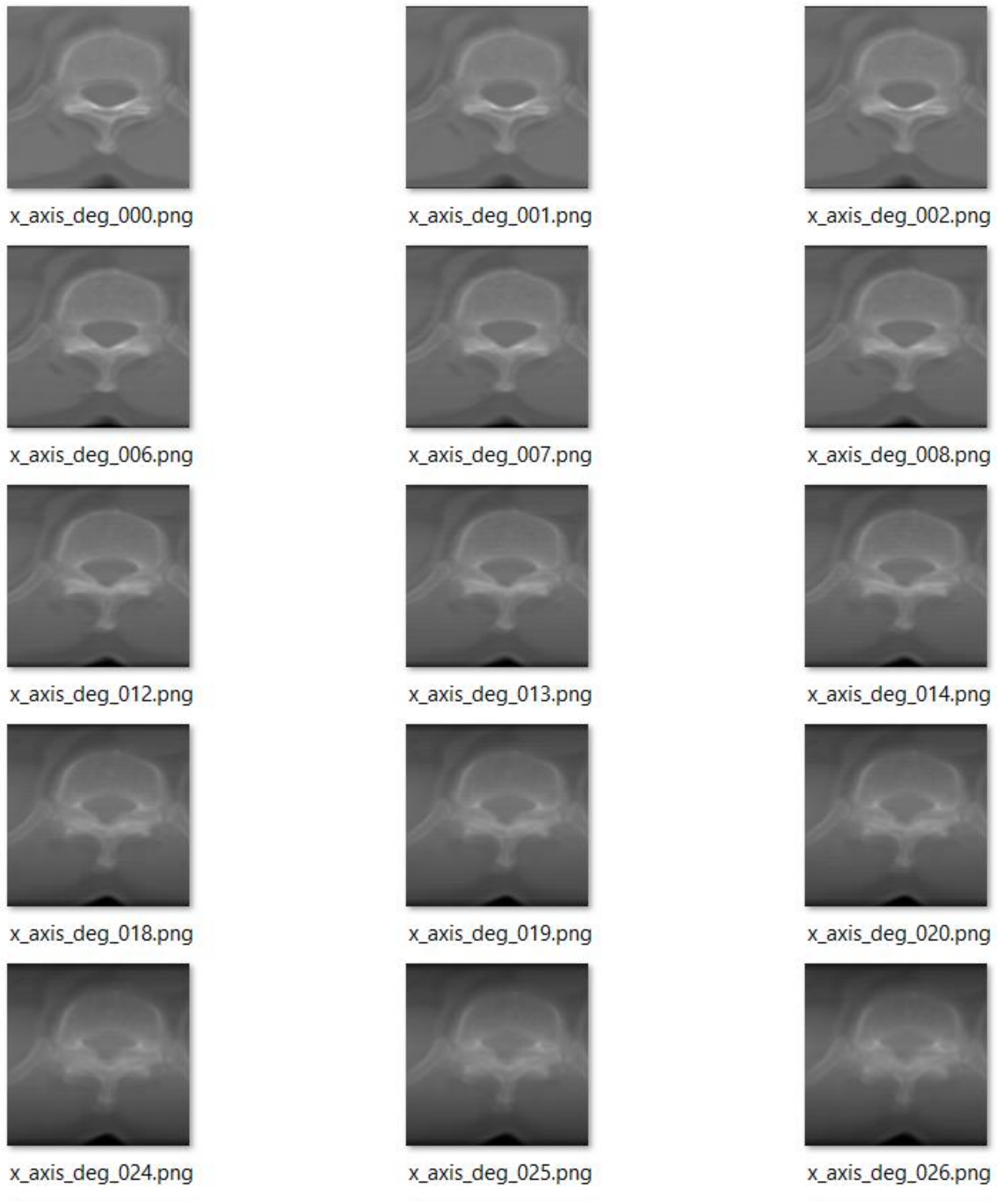

Figure 7: Manually Computed X-ray images

4 Deep Learning Model

4.1 CNN Based Encoder-Decoder framework

Our Deep learning model is implemented based on the network presented by Shen et al [2]. The model is publicly available on GitHub written in Pytorch. We use it as our base and implement the network using Keras. Some details are changed in the network to fit our needs. The overall structure is shown in Figure 8: Part a is the 2D input, in this case it is the manually created projection of a vertebra CT volume. Part b is the representation network, Part c is the transformation network, part d is the generation network and finally, e is the 3D volumetric image that represents the network's prediction. The model is s CNN based encoder-decoder framework. All parts of this model will be introduced in the following sections

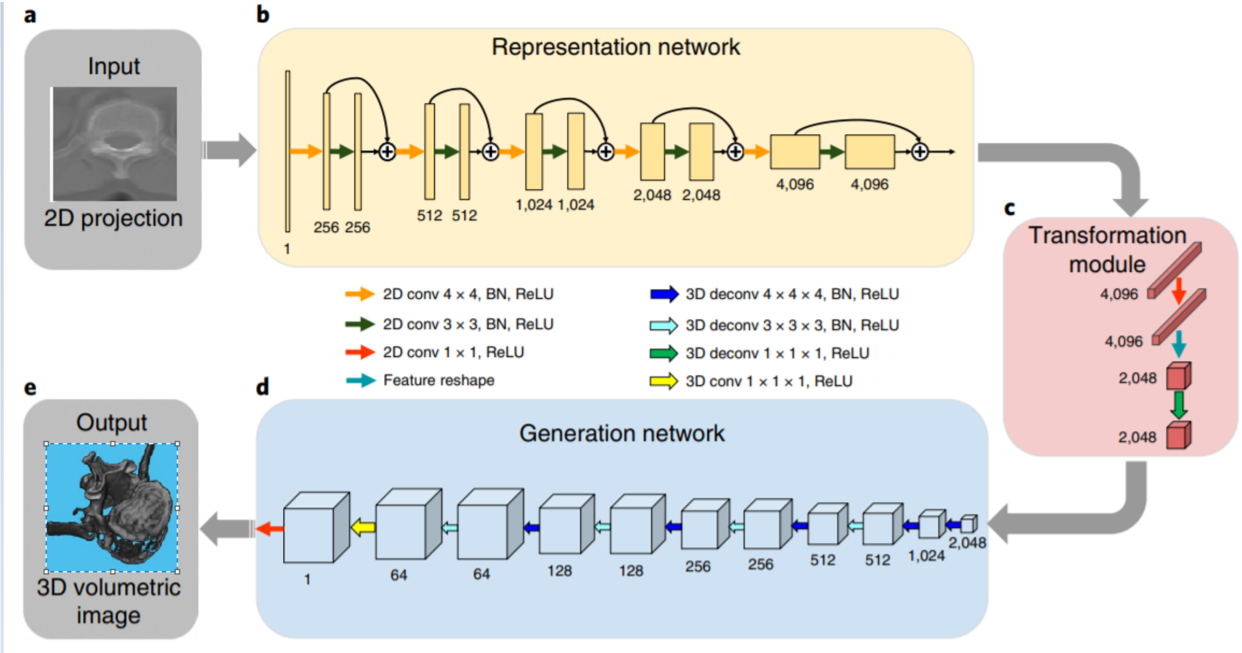

Figure 8: Deep Learning Model Overview [2]

4.2 Representation Network

The representation network is the encoder part of this network. This network takes the 2D input data as the sources and outputs the 2D feature map to the transformation network. There is no shape transformation in this network. The functionality of this network is to convert the original input to the embedded feature maps while downsampling the input data. The basic building block for this network is 2D convolution layers and 2D batch normalization layers. Skip connection is implemented to enhance the learning of the feature maps at each layer as the skip connection combined the feature map learned from the previous layer with the current layer. The network can take in a single or multiple X-ray projections as input. When multiple projections are used to feed in the network. The first layer of the representation network will always try to convert it to the same feature map by adjusting the filter sizes of the convolutional layers. Thus, no matter how many inputs we get. As long as the size of the images is the same, we will get the prediction with the same size.

4.3 Transformation Network

This transformation network is where this model is different from the regular 2D to 2D encoder-decoder framework. We can break down this layer into three parts

- 2D feature maps learning layer: A filter size of 1*1 is applied in this convolutional layer so the size of the feature map does not change. The shape of the feature map at this point is still 2D
- 2D to 3D transformation layer: This layer takes in the 2D feature maps as the input and used the transformation function to transfer 2D feature maps to a 3D representation. There is no learning parameter present in this layer

- 3D feature maps learning layer: A filter size of 1*1*1 is applied in this deconvolutional layer to learn the 3D representation of the feature map transferred by the previous layer. The shape and sizes of the 3D feature maps remain unchanged.

4.4 Generation Network

This layer is the decoder version of the model. It takes the 3D feature maps from the transformation network as its input and outputs the final 3D volumetric array with intensity values as the network's prediction. The functionality of this network is to convert the 3D feature maps to 3D volume while upsampling the data to match the desire 3D shape and size. The basic building block for this network are 3D convolution layers and 3D batch normalization layers

4.5 Loss Function

We can view the process of deep learning as an optimization problem. Deep learning networks use data-driven parameters to optimize a loss function specified by the user. The goal is to minimize the loss function. In theory, if the production is perfect. Then loss function will have a value of 0. In our case, we want to measure the difference between our prediction and the ground truth. So mean square error becomes our first choice since it computes the differences mentioned above for every voxel. So the minimum value for MSE will be 0 if our prediction matches the ground truth 100%. The figure below shows the equation of MSE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE	= mean squared error
n	= number of data points
Y_i	= observed values
$\hat{Y}_i$	= predicted values

Figure 9: MSE Equation [17]

In our case, n is the number of voxels in the 3D volume, Y values will be the observed and predicted intensity values respectfully. MSE is very commonly used in regression problems as the ground truth and the prediction are often numerical types so computing the difference between them is a straightforward yet very effective way to measure accuracy.

5 Model Training Details

5.1 Optimizing the model

As the time I am writing this thesis, the team has conducted 50 different experiments on this project. We focus on optimizing the network in the early stage of our experiments such as adjust the skip connection and adding drop out layer to prevent overfitting. We adjust the learning rate and implement the auto save checkpoints feature so the best performance network is saved as a checkpoint in a training

5.2 Training with different input

Our model in the earlier version can only take one X-ray image as the training input at a time. Midway through our project, we adjust our model to have the ability to take multiple inputs at once. Meanwhile, we also implement 3D object rotation in MATLAB so we can take projection from any angle. The focus shift from adjusting the network to change training input. We tried to experiment with three different combinations: Using projections along the x-axis only, using projection along the z-axis only and using projections along both the x-axis and z-axis. The training result is shown in section 6,1

5.3 Training platform

In the early stage of the experiment, the model was trained on the NVIDIA® GeForce® RTX 2080 Ti graphics card. This GPU has 12 GB of memory, which is not enough to train the network we build so we eventually move on to Google Colab professional version since Google offers a higher GPU Memory.

5.4 Training time

We conduct 50 experiments on this project, averaging 3 model training for each experiment. The average training times vary from 12 hours to a day. The total effort on model training in this project is roughly 2,400 hours

6 Result

In section 6.1, the training results of various experiments explained in section 5.2 are displayed in three separate tables. Section 6.2 shows the ground truth of one test vertebrae and three predictions generated by models that have the highest PSNR value.

6.1 Training Result

Table 1:Training dataset containing x and z axes projections

Exp Number	Number of projections used for training	Number of epochs	Training Loss	Validation Loss
35	360	52	0.0028	0.0049
36	720	86	0.0047	0.0057
37	216	54	0.0023	0.0027
38	504	26	0.0029	0.0035

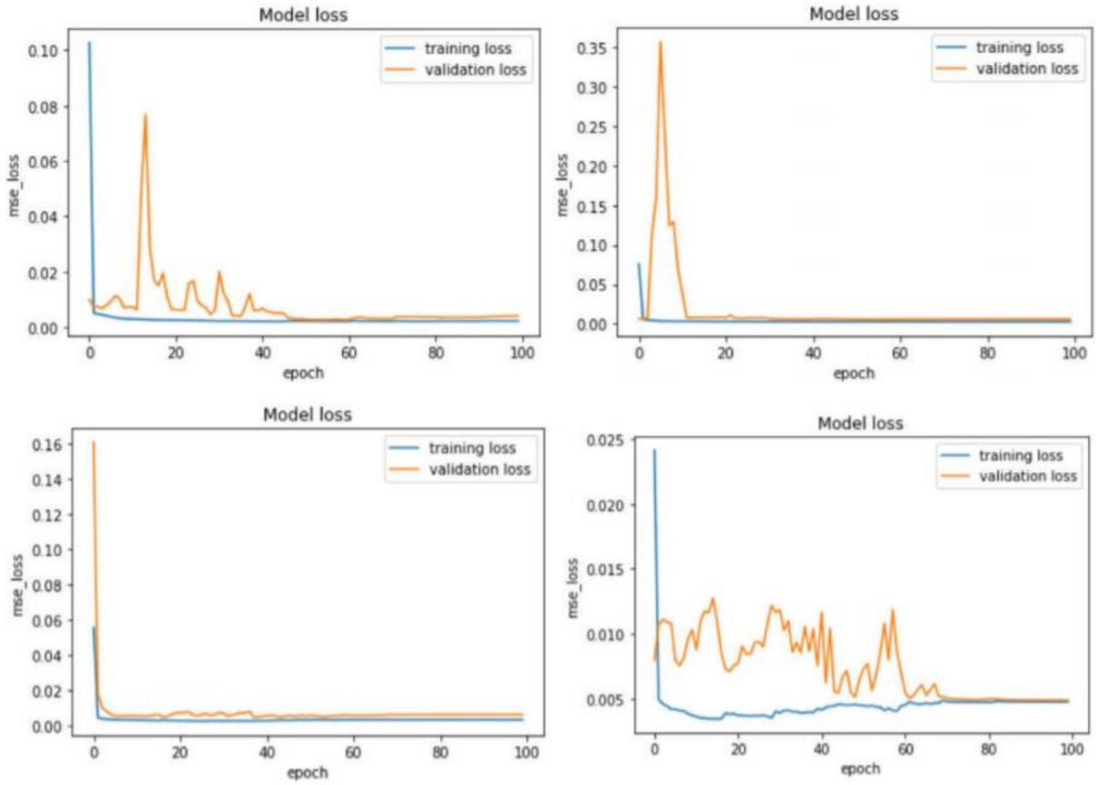

Figure 10: Loss curve for x and z axes projections

Table 2:Training dataset containing x-axis projections only

Exp Number	Number of projections used for training	Number of epochs	Training Loss	Validation Loss
39	360	86	0.0050	0.0057
40	256	26	0.0039	0.0054
41	180	30	0.0035	0.0052
42	108	70	0.0029	0.0044

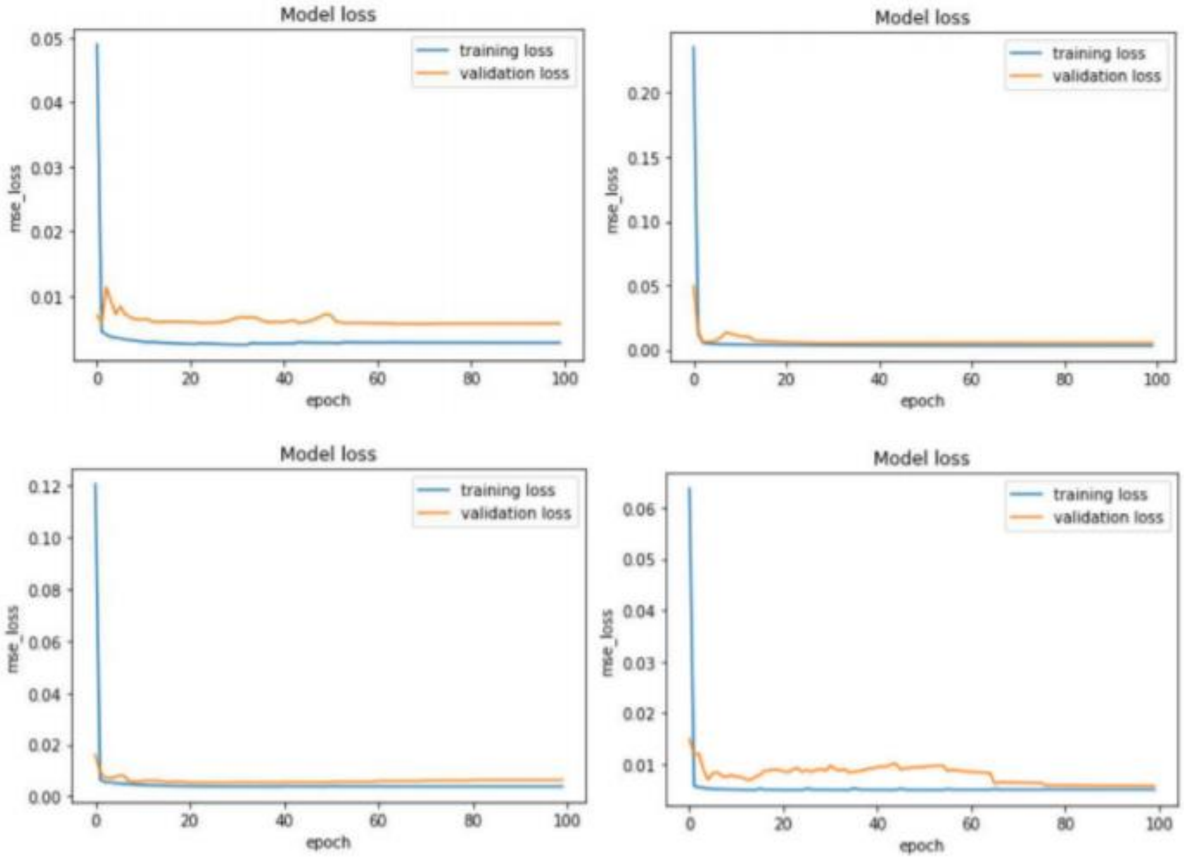

Figure 11:Loss curve for x-axis projections

Table 3:Training dataset containing z-axis projections only

Exp Number	Number of projections used for training	Number of epochs	Training Loss	Validation Loss
43	360	23	0.0047	0.1120
44	256	27	0.0037	0.0077

45	180	30	0.0048	0.0067
46	108	22	0.0031	0.0059

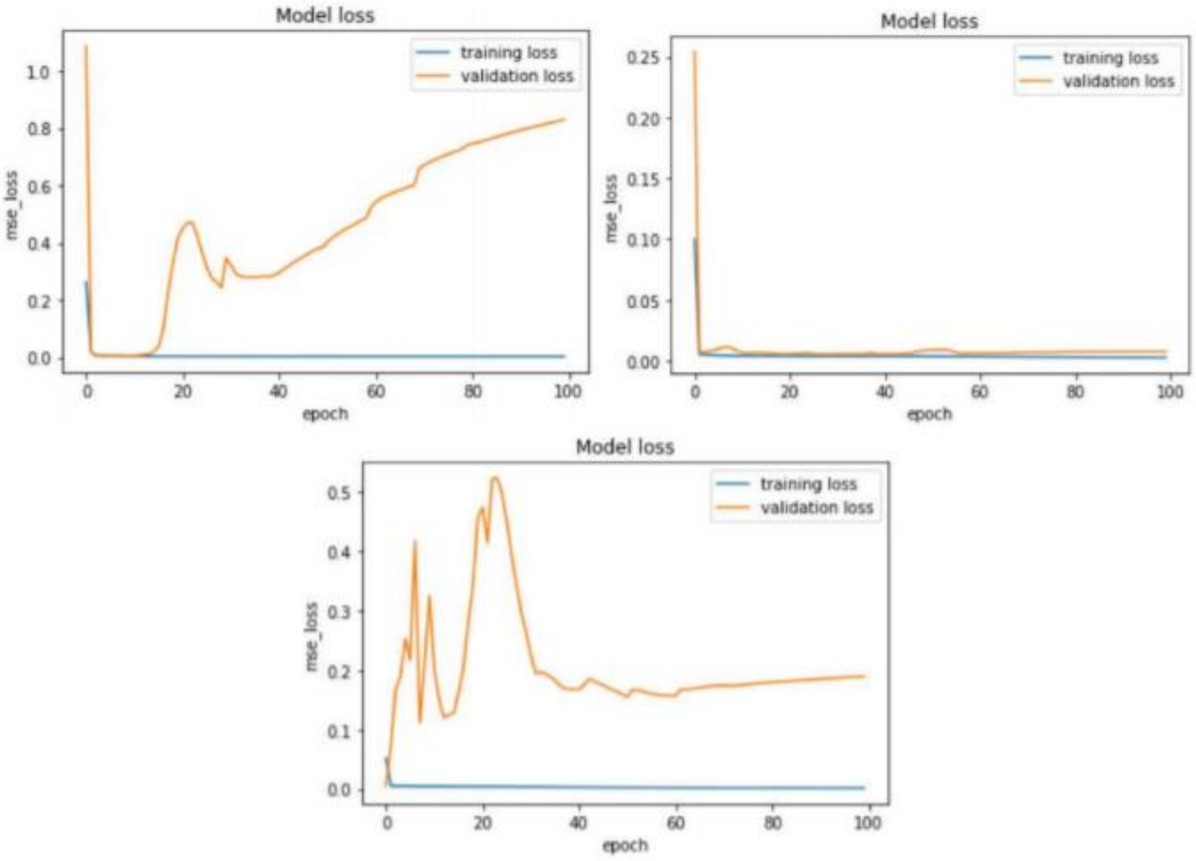

Figure 12:Loss curve for z-axis projections

Key observation:

- Training experiments with fewer projections tend to perform better than the ones with more projections. This is because we carefully choose the X-rays with the best quality for training. As the number of projections becomes larger. Inevitably there will be X-rays with worse quality. The reason for the quality variance is explained in section 3.3. This observation shows we don’t have enough high-quality data at hands so as the dataset gets larger, our result is skewed by noisy data

- Training experiments with X-rays projected along the x-axis have the best performance among the three groups. As mentioned in section 3.3, projection along the x-axis has the best quality. So training results with the x-axis projection have better performance than the other groups. This observation further proves that high-quality data is the key for a better model performance in deep learning project

6.2 Ground truth vs Prediction

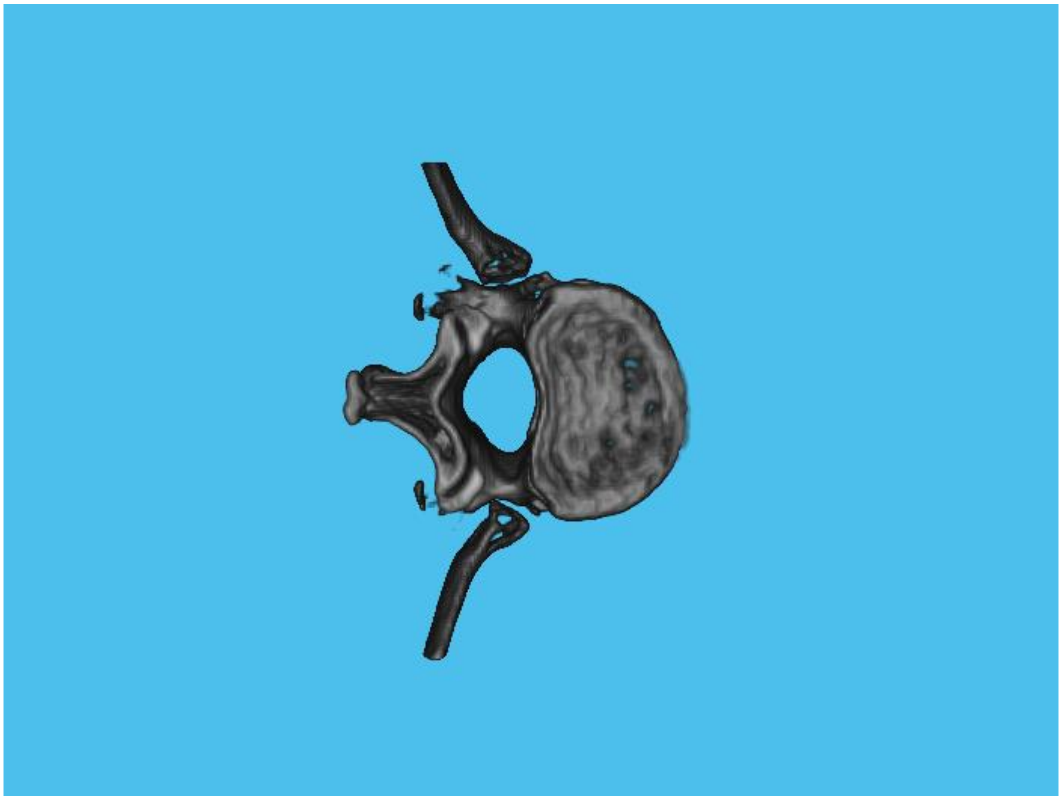

Figure 13: Ground Truth

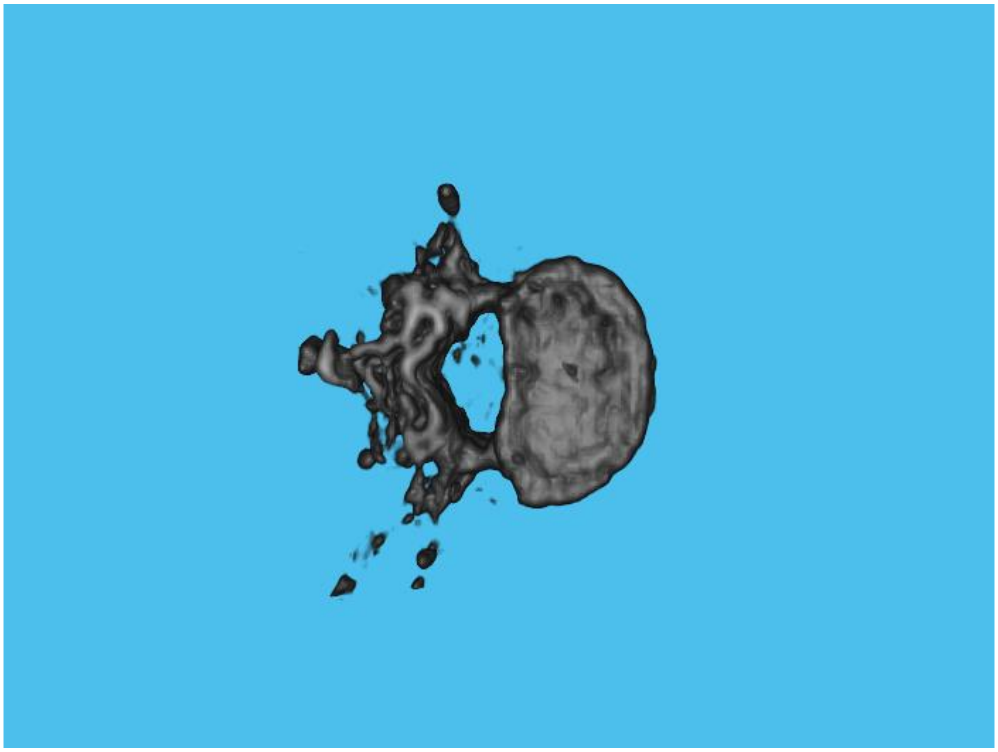

Figure 14: Prediction 1, PSMR: 12.87

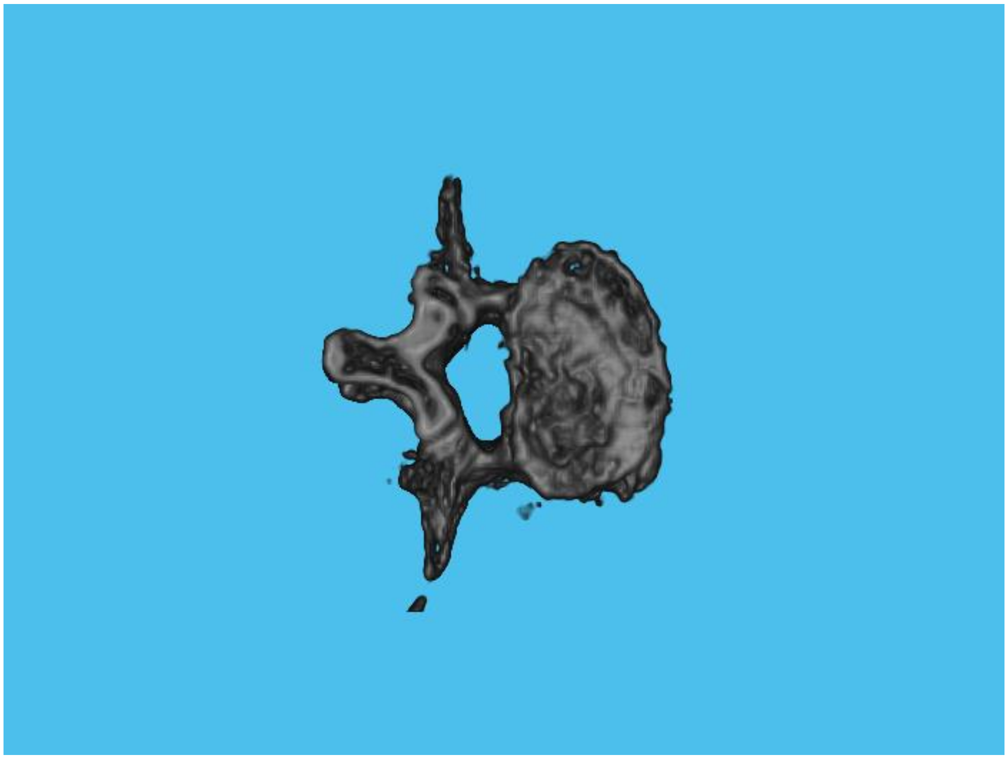

Figure 15: Production 2, PSNR: 14.74

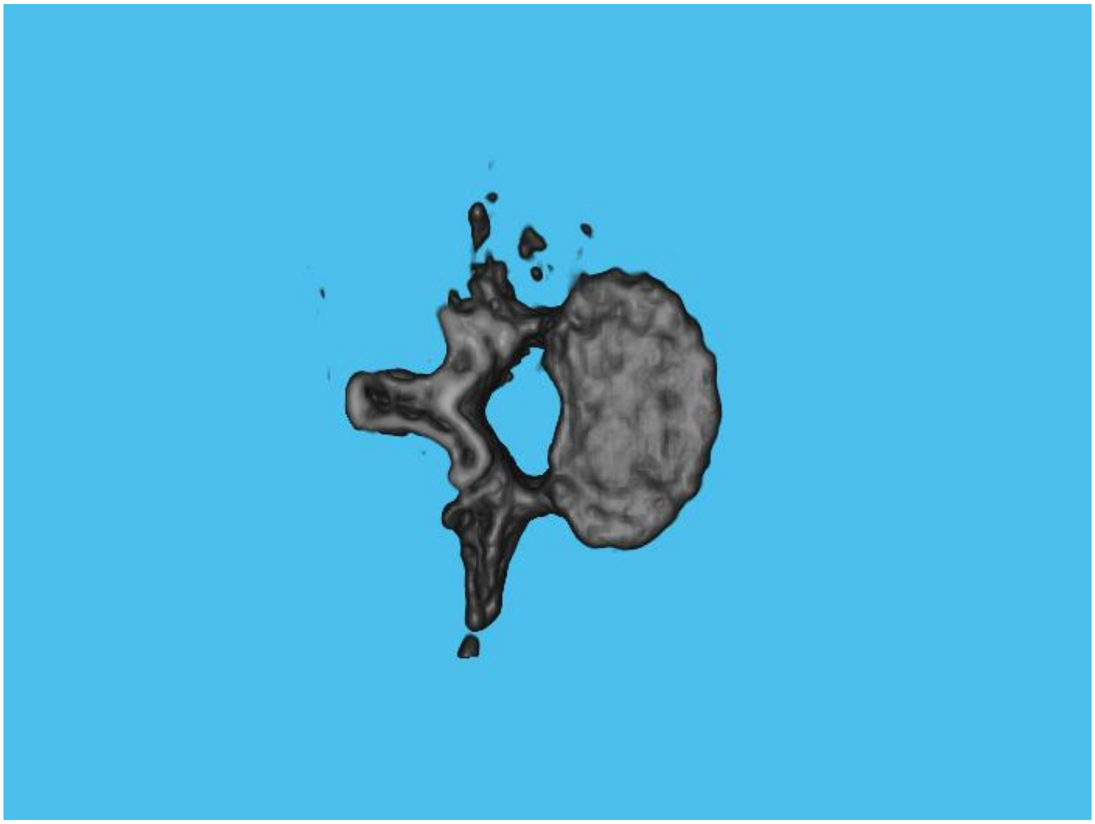

Figure 16: Production 3, PSNR: 13.05

PSNR (Peak Signal-to-Noise) [18], the equation is shown in the below figured, where MAX_f is the maximum intensity of an image. MSE here is mean square error, the equation shown in figure 9. PSNR is a common matrix for accessing the image quality of the target image compare to the original image, which is an ideal matrix to evaluate the quality of image reconstruction. Another reason we choose PSNR is it's based on MSE, which we use as our loss function. The ideal range for PSNR for the image reconstruction is above 30

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

Figure 17: PSNR Equation [18]

7 Conclusion

7.1 Proof of Concept

Based on the evaluation of the result, it is clear that the result can not be applied to real medical applications since the accuracy is too low. None the less, this study proves the feasibility of the approach to use deep learning to reconstruct CT volume given ultra-sparse X-rays as input. One of the biggest observations we got from this project is that a large amount of high-quality training data is the key to better model performance in deep learning projects.

7.2 Limitations

This section will introduce our biggest challenge while doing this research, namely, limitation on computing power. Based on our experiences with this project. Training deep learning models that involve 3D volume will require a significant amount of computing resources. Our single RTX 2080 ti GPU can not run the network so we switched to Google Colab, the problem with Google Colab is that it limit the training time to be 24 hours maximum per training. This limitation significantly affects our training result as training usually requires days or even weeks for the learning to converge to a reasonable result. So our challenge is that either we use our GPU to have unlimited training time but we have to shrink the network and downsample the inputs. Which will hurt the performance, or we use Google Colab to have a better GPU memory but are limited on training time, It is estimated by our team that if we desire results that is acceptable for medical use cases. Then a GPU of memory at least 16 GB is required, with training time being up to a week

7.3 Future Work

This is an ongoing project at the UWM Visualization Lab. The team finds a new dataset that has hundreds of chest CT data, so we are moving on to a larger dataset for training. The team is also exploring different methods for projecting X-ray images from CT volume so the image quality could be better. As we know the input quality dominates the model performance. Lastly, the team is planning to buy a new station with a better GPU to eliminate the limitation on the computing resources. With all these improvements, the hope is one day we can apply this project to help Doctors so that they only need to take a single X-ray image and we can provide an acceptable CT scan using our model.

References

1. Glover , Lacie. “Why Your MRI or CT Scan Costs An Arm and a Leg,” July 21, 2014.

2. Shen, L., Zhao, W. & Xing, L. Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning. Nat Biomed Eng 3, 880–888 (2019).

3. Stegmann, Mikkel B., and David Delgado Gomez. "A brief introduction to statistical shape analysis." Informatics and mathematical modelling, Technical University of Denmark, DTU 15, no. 11 (2002).

4. Novosad J., Cheriet F., Petit Y., labelle H.: Three dimensional reconstruction of the spine from a single x-ray image and prior vertebra models. IEEE Trans. Bio. Eng. 51, 9 (2004), 1628–39

5. Lamecker H., Wenckebach T. H., hege H.-C.: Atlas based 3D-shape reconstruction from X-ray images. In Proc. ICPR pp. 371–374 (2006),

6. Karade, V., Ravi, B. 3D femur model reconstruction from biplane X-ray images: a novel method based on Laplacian surface deformation. Int J CARS 10, 473–485 2015.

7. Wang, Weiyue, Qiangui Huang, Suya You, Chao Yang, and Ulrich Neumann. "Shape inpainting using 3d generative adversarial network and recurrent convolutional networks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2298-2306. 2017.

8. Henzler, Philipp, V. Rasche, T. Ropinski and T. Ritschel. “Single‐image Tomography: 3D Volumes from 2D Cranial X‐Rays.” Computer Graphics Forum 37 (2018): n. pag.

9. X. Ying, H. Guo, K. Ma, J. Wu, Z. Weng and Y. Zheng, "X2CT-GAN: Reconstructing CT From Biplanar X-Rays With Generative Adversarial Networks," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 10611-10620, doi: 10.1109/CVPR.2019.01087.

10. Milickovic, N., Baltast, D., Giannouli, S., Lahanas, M., and Zamboglou, N. CT imaging based digitally reconstructed radiographs and their application in brachytherapy. Physics in medicine and biology, 45(10), 2787–2800 (2000).

11. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron;, Deep Learning, MIT Press,2016.

12. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In International Conference on Medical image computing and computer-assisted intervention, pp. 234-241. Springer, Cham, 2015.

13. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

14. Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788, 2016.

15. A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," 2010 20th International Conference on Pattern Recognition, Istanbul, 2010, pp. 2366-2369, doi: 10.1109/ICPR.2010.579.

16. Defez, E., Villanueva-Oller, J., Villanueva, R. et al. Matrix Cubic Splines for Progressive 3D Imaging. Journal of Mathematical Imaging and Vision 17, 41–53 (2002).

17. “Mean Squared Error.” Wikipedia. Wikimedia Foundation, November 18, 2020. https://en.wikipedia.org/wiki/Mean_squared_error.

18. “Peak Signal-to-Noise Ratio.” Wikipedia. Wikimedia Foundation, October 29, 2020. https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.