

**Final Year Project - Final Report**

# **Early Diagnosis of Scoliosis in Children from RGB-D Images Using Deep Learning**

**Instructor:**

Dr. Kenneth K. Y. Wong

**Group Members and UIDs:**

Huang Siyi, 3035335349

Li Gengyu, 3035331886

**Author:**

Li Gengyu

**Date:**

May 3, 2020

## Abstract

Scoliosis, as a common disease that can be observed on children, needs to be diagnosed as early as possible to avoid deterioration. Traditional methods require experienced doctors to spend lots of time studying the X-ray images of the back of the patients to calculate the Cobb Angle. Modern methods using deep learning technologies require the deep learning models to process and learn from the X-ray images of the back of the patients to make diagnosis. However, the radiation given out by X-ray machines when collecting X-ray images is harmful for children health. None of the existing methods can completely avoid this drawback because X-ray images are required in these methods to conduct the diagnosis.

In this report, we will summarize our final year project in detail about a new method to diagnose scoliosis by synthesizing the X-ray images from the RGB-D images of the back of the patients using deep learning models, which can rid the whole process of radiation. We used 6 anatomical landmarks to reveal the shape of the spine curve. To get these landmarks, we first did landmark detection on RGB-D images using the High-Resolution Nets, which is a noble framework for pixel-level classification tasks. After that, we used the *pix2pix* model to synthesize X-ray images from RGB-D images and anatomical landmarks. We have already achieved some good results on the dataset collected by ourselves and the supporting medical staff.

The source code and models are available at: <https://github.com/dawnnonme/FYP-Gradient-Explosion>. The dataset, best performance result in each stage are available at: [https://drive.google.com/open?id=1b8mpmkUcvyEuy7sQqAmVjR\\_NytFBfyTH](https://drive.google.com/open?id=1b8mpmkUcvyEuy7sQqAmVjR_NytFBfyTH).

## List of Contents

<b>1. Introduction</b>	<b>11</b>
<b>1.1. Background</b>	<b>11</b>
<b>1.2. Project Objective</b>	<b>12</b>
<b>2. Preliminary Works</b>	<b>12</b>
<b>2.1. The Cobb Angle</b>	<b>12</b>

<b>2.2. Anatomical Landmarks</b>	<b>14</b>
<b>2.3. Existing deep learning-based Methods to Diagnose AIS</b>	<b>14</b>
<b>2.4. High-Resolution Net: Pixel-level Classification Tasks</b>	<b>15</b>
<b>2.5. Conditional GAN: Image-to-Image Translation</b>	<b>16</b>
<b>2.6. From Surface Geometry to X-ray Images</b>	<b>16</b>
<b>3. Project Review</b>	<b>16</b>
<b>3.1. Project Motivation</b>	<b>16</b>
<b>3.2. Project Structure</b>	<b>17</b>
<b>3.3. Project Process and Outcomes</b>	<b>18</b>
<b>3.4. Workload Distribution</b>	<b>19</b>
<b>4. Methodologies</b>	<b>19</b>
<b>4.1. Data Collection</b>	<b>19</b>
<i>4.1.1. Microsoft Azure Kinect DK</i>	<i>20</i>
<i>4.1.2. RGB-D Images</i>	<i>20</i>
<i>4.1.3. Alignment of RGB-D images</i>	<i>23</i>
<i>4.1.4. Landmarks on RGB-D Images</i>	<i>25</i>
<i>4.1.5. X-ray Images and Landmarks</i>	<i>26</i>
<b>4.2. Landmark Detection</b>	<b>27</b>
<i>4.2.1. The High-Resolution Networks</i>	<i>27</i>
<i>4.2.2. Dataset Split</i>	<i>28</i>
<i>4.2.3. Data Normalization</i>	<i>28</i>
<i>4.2.4. Data Augmentation</i>	<i>29</i>
4.2.4.1. Rotation	29

4.2.4.2. Horizontal Flip	29
4.2.4.3. Translation	29
4.2.4.4. Rescaling	29
4.2.4.5. Depth Offset	30
4.2.4.6. Random Erasing to Background	30
<i>4.2.5. Training Strategies</i>	<i>31</i>
<i>4.2.6. Ground Truth</i>	<i>31</i>
<i>4.2.7. Loss Function</i>	<i>32</i>
<i>4.2.8. Performance Metrics</i>	<i>33</i>
<i>4.2.9. Result</i>	<i>33</i>
<b>4.3. X-ray Synthesis</b>	<b>33</b>
<i>4.3.1. Data Alignment</i>	<i>33</i>
<i>4.3.2. The pix2pix Model</i>	<i>35</i>
4.3.2.1. Generator	36
4.3.2.2. Discriminator	37
4.3.2.3. GAN Mode	37
<i>4.3.3. Dataset Split</i>	<i>37</i>
<i>4.3.4. Data Normalization</i>	<i>37</i>
4.3.4.1. Contrast Stretching on X-ray Images	37
4.3.4.2. Further Normalization	39
<i>4.3.5. Data Augmentation</i>	<i>39</i>
4.3.5.1. Rotation	39
4.3.5.2. Horizontal Flip	39

4.3.5.3. Translation	39
4.3.5.4. Rescaling	40
4.3.5.5. Depth Offset	40
4.3.6. <i>Training Strategies</i>	40
4.3.7. <i>Loss Function</i>	40
4.3.8. <i>Performance Metrics</i>	41
4.3.9. <i>Result</i>	43
<b>5. Experiments</b>	<b>43</b>
<b>5.1. Landmark Detection</b>	<b>43</b>
5.1.1. <i>Input Data Composition</i>	43
5.1.1.1. Motivation	43
5.1.1.2. Experiment Design	44
5.1.1.3. Result and Analysis	44
5.1.2. <i>Validity of Data Augmentation</i>	46
5.1.2.1. Motivation	46
5.1.2.2. Experiment Design	46
5.1.2.3. Result and Analysis	46
5.1.3. <i>Method for Data Normalization</i>	49
5.1.3.1. Motivation	49
5.1.3.2. Experiment Design	50
5.1.3.3. Result and Analysis	51
5.1.4. <i>Comparison with Other Models</i>	52
5.1.4.1. Motivation	52

5.1.4.2. Experiment Design	52
5.1.4.3. Result and Analysis	53
<b>5.2. X-ray Synthesis</b>	<b>53</b>
<i>5.2.1. X-ray Enhancement</i>	53
5.2.1.1. Motivation	53
5.2.1.2. Experiment Design	54
5.2.1.3. Result and Analysis	54
<i>5.2.2. Avoid Overfit</i>	58
5.2.2.1. Motivation	58
5.2.2.2. Experiment Design	58
5.2.2.3. Result and Analysis	58
<i>5.2.3. Data Alignment</i>	60
5.2.3.1. Motivation	60
5.2.3.2. Experiment Design	60
5.2.3.3. Result and Analysis	61
<i>5.2.4. Backbone for Generator and Image Resolution</i>	62
5.2.4.1. Motivation	62
5.2.4.2. Experiment Design	62
5.2.4.3. Result and Analysis	63
<i>5.2.5. Composition of Input Data</i>	65
5.2.5.1. Motivation	66
5.2.5.2. Experiment Design	66
5.2.5.3. Result and Analysis	66

5.2.6. <i>Weight of L1 Loss</i>	68
5.2.6.1. Motivation	68
5.2.6.2. Experiment Design	69
5.2.6.3. Result and Analysis	69
<b>6. Future Works</b>	<b>71</b>
<b>7. Conclusion</b>	<b>72</b>

### **Abbreviations**

1. AIS: Adolescent Idiopathic Scoliosis
2. CNN: Convolutional Neural Network
3. CGAN: Conditional Generative Adversarial Nets
4. D: Discriminator
5. DKH: The Duchess of Kent Children’s Hospital at Sandy Bay, Hong Kong
6. FC: Fully Connected Neural Network
7. G: Generator
8. GAN: Generative Adversarial Net
9. HRNet: High-Resolution Net
10. LSGAN: Least Squares Generative Adversarial Net
11. MSCOCO: Microsoft Common Objects in Context
12. MSELoss: Mean Squared Error Loss
13. ResNet: Deep Residual Networks
14. VGG: Visual Geometry Group

### **List of Figures**

- |  |    |
|--|----|
| Figure 1: Comparison between spine in scoliosis (left) and normal spine (right).   | 11 |
| Figure 2: Comparison between the Cobb Angle on a spine with scoliosis (left) and the Cobb Angle on a normal spine (right). | 13 |

Figure 3: 6 anatomical landmarks involved in the project and their names.	14
Figure 4: Flow chart of the project structure.	18
Figure 5: One of RGB-D images which consists of a RGB image (left) and a depth image (right).	21
Figure 6: Visualized heat map from a depth image.	21
Figure 7: 6 random depth images after filtering using range [1200,1800].	22
Figure 8: Horizontal translation between RGB images and depth images.	24
Figure 9: One of the anatomical landmarks in a .txt file.	26
Figure 10: Landmarks on one of the RGB-D images.	26
Figure 11: Architecture of the HRNet [5].	27
Figure 12: Visualized depth image (left) and Visualized depth image with random noise on background (right).	31
Figure 13: Example for the 6 ground truth heat maps.	32
Figure 14: 2 unaligned X-ray images.	34
Figure 15: An aligned data sample containing a RGB image (left), a depth image (middle), an X-ray image (right) and anatomical landmarks (not shown).	35
Figure 16: X-ray images before and after contrast stretching.	39
Figure 17: The loss curves for training (left) and validation (right) with 3 data compositions.	45
Figure 18: The loss curves for training (left) and validation (right) with or without data augmentations.	47
Figure 19: 9 randomly picked results images and landmarks in A3.	49
Figure 20: 9 randomly picked results images and landmarks in A0.	49
Figure 21: Distribution of the pixel values of the depth images (values that occurred rarely will have a very short bar so that could not be observed).	51



Figure 22: The loss curves for training (left) and validation (right) with different data standardization methods. 52

### List of Tables

Table 1: Workload distribution in this project.	19
Table 2: Statistics of RGB images.	23
Table 3: Statistics of depth images over 2 measurements.	23
Table 4: Cases where SSIM and PSNR failed to measure the performance of different models in this project.	42
Table 5: Results of 3 data compositions on testing dataset.	45
Table 6: Results of model with or without data augmentations on testing dataset.	47
Table 7: Results of model with different data standardization methods on testing dataset.	52
Table 8: Results of different models on the testing dataset.	53
Table 9: X-ray images enhanced by 3 algorithms.	55
Table 10: Distributions of X-ray images shown in table enhanced by 3 algorithms.	56
Table 11: Synthetic X-ray images and the corresponding ground truth in B0 and B1.	57
Table 12: Mean histogram intersection and mean image hashing in B0, B2, B3, B4.	59
Table 13: Synthetic X-ray images and the corresponding ground truth in B0, B2, B3, B4.	59
Table 14: Synthetic X-ray images and the corresponding ground truth in B0 and B5.	61
Table 15: Mean histogram intersection and mean image hashing in B0, B6, B7, B8, B9, and B10.	63
Table 16: Synthetic X-ray images and the corresponding ground truth in B0, B6, B7, B8, B9, and B10.	65
Table 17: Mean histogram intersection and mean image hashing in B0, B11, B12, B13, B14, and B15 (L stood for anatomical landmarks).	66

Table 18: Synthetic X-ray images and the corresponding ground truth in B0, B11, B12, B13, B14, and B15. 68

Table 19: Mean histogram intersection and mean image hashing in B0, B16, B17, B18, B19.69

Table 20: Synthetic X-ray images and the corresponding ground truth in B0, B16, B17, B18, and B19. 70

Table 21: Final results of stage 2. 76

Table 22: Final results of stage 3. 80

## 1. Introduction

### 1.1. Background

Scoliosis, which is defined as a morbid medical condition where the spine of the patient curves sideways, can be easily found on children [1, 2, 3]. As is shown in figure 1 below, the X-ray on the left has a more twisty spine curve compared with the X-ray on the right, which shows the basic characteristic of scoliosis.



*Figure 1: Comparison between spine in scoliosis (left) and normal spine (right).*

Early diagnosis and treatment can prevent Adolescent Idiopathic Scoliosis (AIS) from deteriorating [1, 2]. Therefore, great importance has been laid on the early diagnosis of scoliosis in children. Conventionally, scoliosis is diagnosed by computing an angle called the Cobb Angle [1, 2, 3]. The Cobb Angle can be computed manually by studying the X-ray image of the back of the patient. Overtime, the Cobb Angle has been proved to be reliable and has been the golden rule to diagnose scoliosis for many years. However, this traditional method has a lot of drawbacks. For instance, manual work induces approximation, which makes it easy for error to occur [3]. It also necessitates massive workload and time to study the X-ray images, possibly

missing the best timing to treat this disease [3]. Most importantly, X-ray images are taken using X-ray machines, which give out radiation. Severe damage to human especially children who are not fully mature is likely to be caused by the radiation. To address these problems, we proposed our project.

As the development of deep learning in computer vision, more and more deep learning techniques have been designed to solve problems in medical images. In this project, to overcome the drawbacks of traditional methods, deep learning technologies were used.

This final report is written to give a detailed summary of our final year project — “Early Diagnosis of Scoliosis in Children using RGB-D Images by Deep Learning”. It covers all the process and outcomes from Sep 1, 2019 to May 2, 2020. This report will first give a brief description about our project. Then the review of some related works will be covered. After that, it will go through a detailed explanation about the project structure and the implementation methodologies. Finally, we will provide some limitations of our project and potential improvements that can be made in the future.

## **1.2. Project Objective**

The objective of this project was to design and implement a deep learning model to synthesize X-ray images from the corresponding RGB-D images of the back of the patients. To achieve this target, an intermediate step where landmark detection was done on RGB-D images would be needed to extract more information about the shape of the spine curve of the patients.

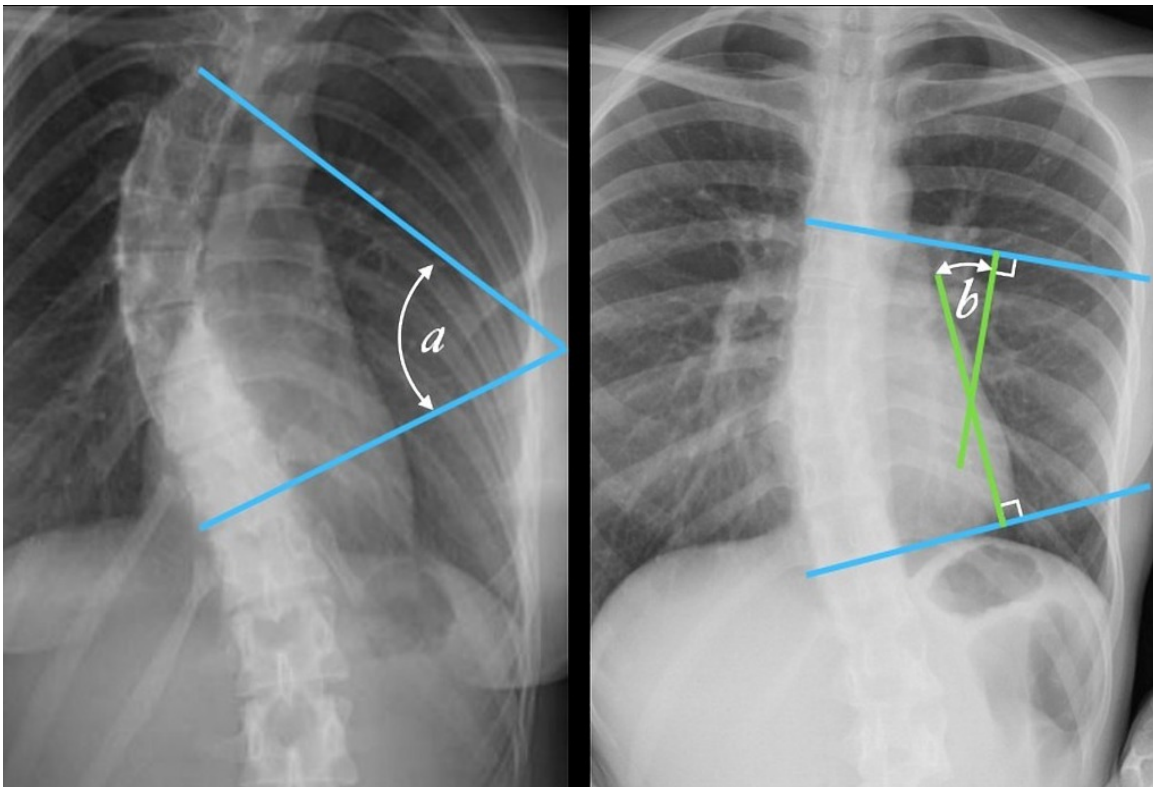
It should be highlighted that our project did not make any diagnosis for medical staff. Further research was still needed for medical staff to make the proper diagnosis using X-ray images generated in this project. In short, our project synthesized X-ray images without further analysis of the deformity of the spines.

More details will be provided in section 3 and 4.

## **2. Preliminary Works**

### **2.1. The Cobb Angle**

The Cobb Angle is the golden rule for diagnosis of scoliosis. Since its invention in 1948, the Cobb Angle has been tested on a great amount medical cases. High accuracy makes it a reliable metric in diagnosis of scoliosis [1, 2, 3]. As shown in figure 2, the Cobb Angle measures the angle formed by 2 perpendicular lines to the spine curve. Moreover, the larger the Cobb Angle, the more possible for the patient to suffer scoliosis. The angle  $a$  on the left is the Cobb Angle from a spine with scoliosis while the angle  $b$  on the right is the Cobb Angle from a normal spine. By comparison, angle  $a$  is much larger than angle  $b$ .

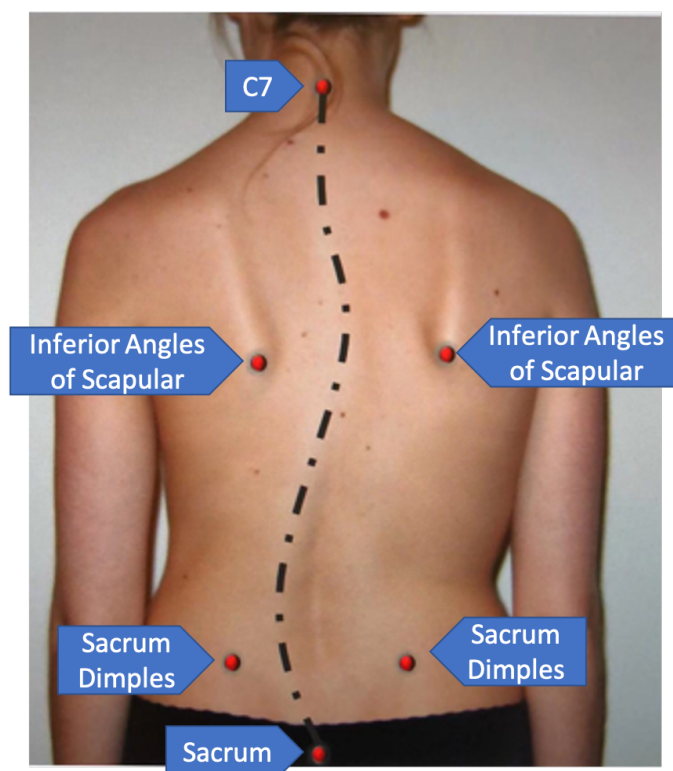


*Figure 2: Comparison between the Cobb Angle on a spine with scoliosis (left) and the Cobb Angle on a normal spine (right).*

The Cobb Angle is a basis for diagnosis of scoliosis. In this project, computing the Cobb Angle was not the target. However, the synthesized X-ray images would be a prerequisite to calculating the accurate Cobb Angle. Therefore, it was essential for us to synthesize X-ray images with high quality.

## 2.2. Anatomical Landmarks

Anatomical landmarks are several key points on the back of a person. In this project, we only used 6 of them. As shown in figure 3 below, these 6 points contains essential information about the shape of the spine curve. C7 and Sacrum measures the starting and the ending points for the spine curve.



*Figure 3: 6 anatomical landmarks involved in the project and their names.*

As these 6 anatomical landmarks contained essential information about the shape of the spine curve, in this project, landmark detection was carried out on RGB-D images to detect these 6 landmarks, which played a significant role in the later process to synthesize X-ray images.

## 2.3. Existing deep learning-based Methods to Diagnose AIS

Some deep learning-based methods for estimation of the Cobb Angle have made some improvements for the traditional methods. Those deep learning-based methods have different architectures though, their rough ideas are similar. Instead of manual calculation of the Cobb Angle which will cause many errors, those methods use a deep learning model to learn to compute the Cobb Angle after processing the X-ray images or moire images of the spines of the patients.

Although the accuracy of the Cobb Angle was greatly improved by those methods, those methods still cannot avoid radiation. As the X-ray machine or the moire machine both gave out radiation, those deep learning-based methods were still not suitable for children.

Ran et al. proposed a simple deep learning-based method to conduct a 3-step Cobb Angle estimation on the moire images of patients [3]. The model was accurate in the estimation of the Cobb Angle. However, collection of moire images also involves radiation.

What is more, Hongbo et al. proposed BoostNet and MVC-Net for AIS assessment which took X-ray images of the patients [1, 2]. The former approach did spinal landmark detection on X-ray images while the latter approach took multi-view X-ray images and studied the correlation among them. These 2 methods had been proved to be accurate to predict the Cobb Angle. Although these 2 models were not the way we go for this project, they did inspire us in the early stage of this project.

Finally, Meelis et al. proposed a method to detect vertebrae and the spine curve of a patient in 3D images of the lumbar spine image of the patient [4]. This project showed a proof of concept that 3D information about the surface could be a good reference to estimate the spine curve. Although no 3D image was given to us in this project, the concept proved by this work inspired us that RGB-D images, which also contained 3D surface information, could also be a good reference to predict the spine curve.

#### **2.4. High-Resolution Net: Pixel-level Classification Tasks**

Ke et al. invented a noble framework named the High-Resolution Net (HRNet) in 2019 for pixel-level classification tasks such as landmark detection, object detection, object segmentation, etc. [5]. The idea to keep high-resolution representations and multi-scale fusion

during the whole training process became an essential idea to preserve the representation power of HRNet.

HRNet was used in this project to do landmark detection on the RGB-D images. More details will be covered in section 4.2.1.

## **2.5. Conditional GAN: Image-to-Image Translation**

Traditional generative adversarial networks (GAN) learns a mapping from Gaussian random noise to the target distribution [6]. However, the raw material of GAN is drawn from random noise, making it hard for GAN to learn the target data distribution. Conditional GAN (CGAN) tackles this problem by drawing the input data from some distributions that are believed to have relation with the target distribution rather than random noise [7]. Therefore, the basis for image-to-image translation is established.

Phillip et al. proposed a CGAN-based framework *pix2pix* for image-to-image translation in 2018 [7]. The *pix2pix* model was used in this project to synthesize X-ray images. More details will be covered in section 4.3.2.

## **2.6. From Surface Geometry to X-ray Images**

Brian et al. proposed a 2-stage deep learning-based method to synthesize X-ray images from surface geometry of patients [8]. The concept proved in this paper became the core motivation of this project. Synthesizing X-ray images from surface geometry using CGAN was proved to be possible in this work. Two deep learning models were designed and trained in dependence of each other to generate X-ray images from partial images and parameterized images in this work, which also directed us in the early stage of this project.

# **3. Project Review**

## **3.1. Project Motivation**

To avoid the potential damage induced by radiation given out by X-ray machines in the process to diagnose AIS, we proposed to use RGB-D images to synthesize X-ray images of the



back of the patients. As the instrument to take RGB-D images would not give out radiation, this problem could be resolved.

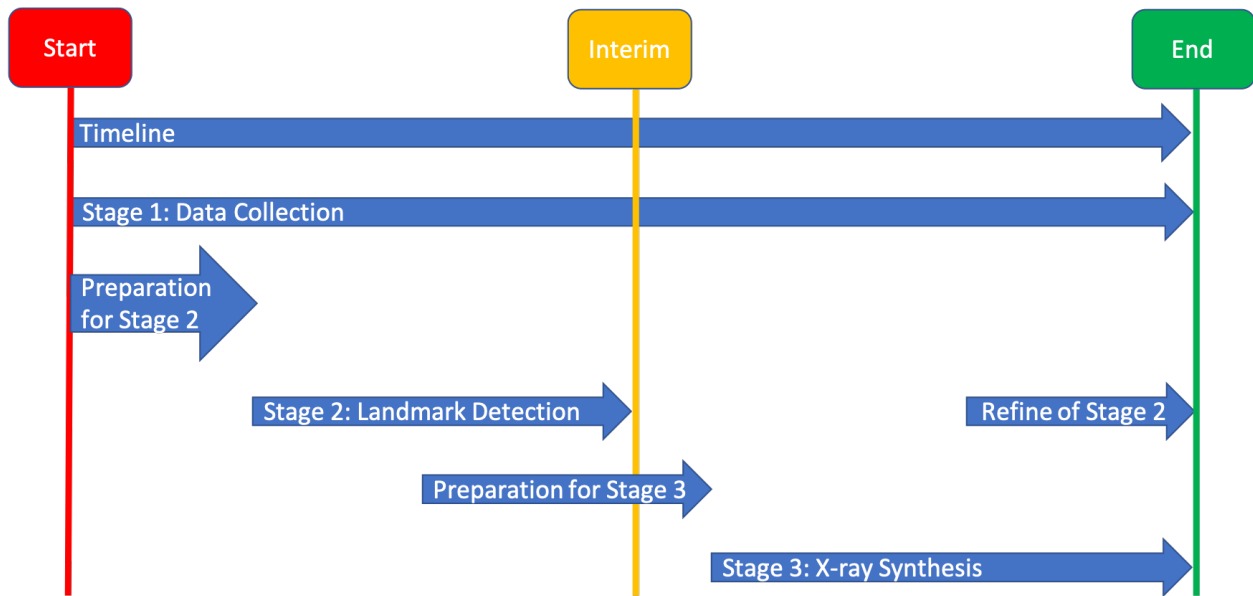
Then the medical process of diagnosis of AIS could be changed to taking RGB-D image of the back of the patient first, synthesizing X-ray image using the RGB-D image, and finally computing the Cobb Angle using other technologies. High accuracy and reliability of the Cobb Angle can still be retained while the potential damage induced by X-ray machines can be avoided.

Moreover, the depth images contained essential information of the surface geometry of the back, making it possible for the deep learning model to learn the shape of the spine curve.

### **3.2. Project Structure**

Our project was divided into 3 stages. From figure 4 below, these 3 stages are named data collection, landmark detection, and X-ray Synthesis. As shown in figure below, stage 1 spanned the whole project period to collect and preprocess as many data samples as possible. In this project, we defined a full data sample as a collection of a RGB-D image, an X-ray image, and 6 anatomical landmarks on the RGB-D image from the back of the same patient. Stage 2 was started after some preparation when related papers and technologies were learnt, and environments such as Python and PyTorch were set on our personal devices and the GPU farm owned by the faculty. Stage 2 temporarily ended at the interim of this project to shift our focus from stage 2 to stage 3 when some relatively good results had been achieved at that time with small amount of data samples. Stage 2 was refined near the end of the project when more data samples were available. Before the interim, we started to prepare for stage 3. After that, stage 3 was carried out till the end of the project.

In next subsection, detailed process and outcomes of each stages will be covered.



*Figure 4: Flow chart of the project structure.*

### 3.3. Project Process and Outcomes

In stage 1, we first set up environment for taking RGB-D images in a photo lab of the Duchess of Kent Children’s Hospital at Sandy Bay (DKH). After that, staff in the photo lab of DKH helped to take RGB-D images using Microsoft Azure Kinect DK and take X-ray images using X-ray machines from the back of the voluntary patients. Meanwhile, staff in photo lab also labeled the RGB-D images with 6 anatomical landmarks.

At the same time, we reviewed all the data samples and reported those samples that were judged as bad samples by the professional medical doctor.

At the end of the project, 560 full data samples have been collected, while additional 67 RGB-D images with their landmarks have been collected without corresponding X-ray images.

In stage 2, a HRNet-based model was designed and implemented to do landmark detection on RGB-D images. The RGB-D images collected in previous stage were fed as input to the model. The landmarks labeled on RGB-D images in stage 1 served as the ground truth for

training and testing. To find the best configuration for the model, several experiments were carried out, details of the experiments will be cover in section 5.

In stage 3, a *pix2pix*-based generative model was designed and implemented to synthesize X-ray images. X-ray images were labeled with 2 anatomical landmarks (C7, Sacrum). The RGB-D images collected in stage 1 and the landmarks detected in stage 2 will be the input of the model. Before that, we preprocessed the RGB-D images and the X-ray images with the available landmarks (6 on RGB-D images and 2 on X-ray images). Again, multiple experiments regarding different aspects of the model were conducted with detailed explanation demonstrated in section 5. The reason why only 2 anatomical landmarks were labeled will be given in section 4.3.

### 3.4. Workload Distribution

The workload distribution is presented in table 1 below. The items in red were done by my teammate Huang Siyi, the items in blue were done by myself, and the items in purple were done collaboratively by us.

Stage 1	Stage 2	Stage 3	Other
Developed a small program for the medical staff to control the depth camera.	Preprocessed all the data.	Preprocessed all the data.	Composed detailed project plan.
Setuped the depth camera in the photo lab.	Designed and implemented data normalization and data augmentation.	Designed and implemented data normalization and data augmentation.	Built project website.
Got statistics of all the data.	Implemented the model.	Implemented the model.	Composed interim report.
	Conducted experiments and analyzed the result.	Conducted experiments and analyzed the result.	Communicated with the medical staff.

Table 1: Workload distribution in this project.

## 4. Methodologies

### 4.1. Data Collection

#### 4.1.1. Microsoft Azure Kinect DK

Microsoft Azure Kinect DK was the depth camera used in stage 1 to take RGB-D images from those voluntary patients. As was illustrated in its documentations, this depth camera could not only take RGB images but also take depth images using an advanced depth sensor [9].

As was mentioned in its documentation, Microsoft Azure Kinect DK had an advanced depth sensor [9]. Therefore, RGB-D images taken by it had high accuracy. Moreover, Microsoft Azure Kinect DK was portable to use because its size was incredibly small. As the room in photo lab of DKH was limited, this small depth camera, which was only 5 inches long and 1.5 inches thin, could save lots of space, making it convenient for staff to use without interrupting existing instruments. On top of that, this camera had an end-to-end software development kit (SDK), making it easy to write a short computer program to control the function of the camera [9]. As the staff in photo lab did not have any experience in computer science, expectation could not be put on them to handle a complicated software. By just a short program written by us, they could complete their job with few commands to our program.

#### 4.1.2. RGB-D Images

A RGB-D image consists of a RGB image and a depth image. One example of RGB-D images is shown in figure 5 below.

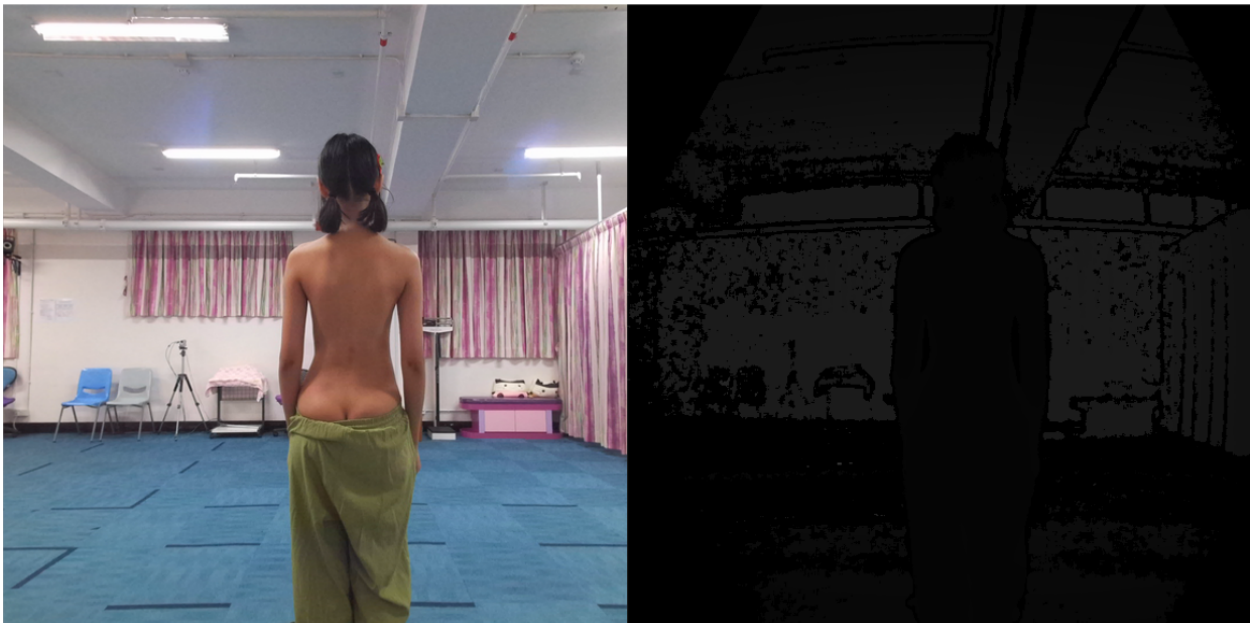


Figure 5: One of RGB-D images which consists of a RGB image (left) and a depth image (right).

The pixel value (intensity) of a depth image measures the actual distance from the point on actual object to the sensor of the depth camera. In our dataset, the pixel value measures the actual distance in millimeters. For example, if a pixel on person has a value on the depth image of 1500, then it will be 1500 millimeters or 1.5 meters from the point on the person to the sensor of depth camera. As shown in figure 6, in a visualized heat map from a depth image in our dataset, the brighter the area, the closer it is to the depth sensor.

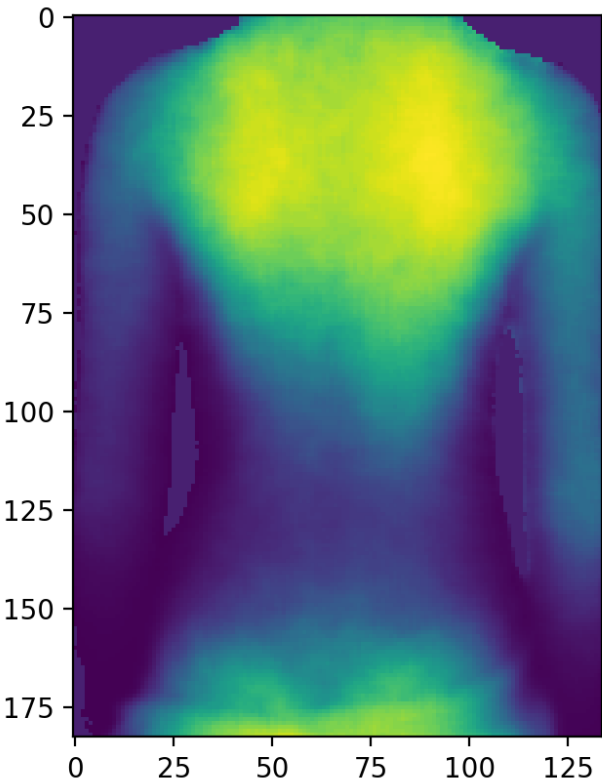
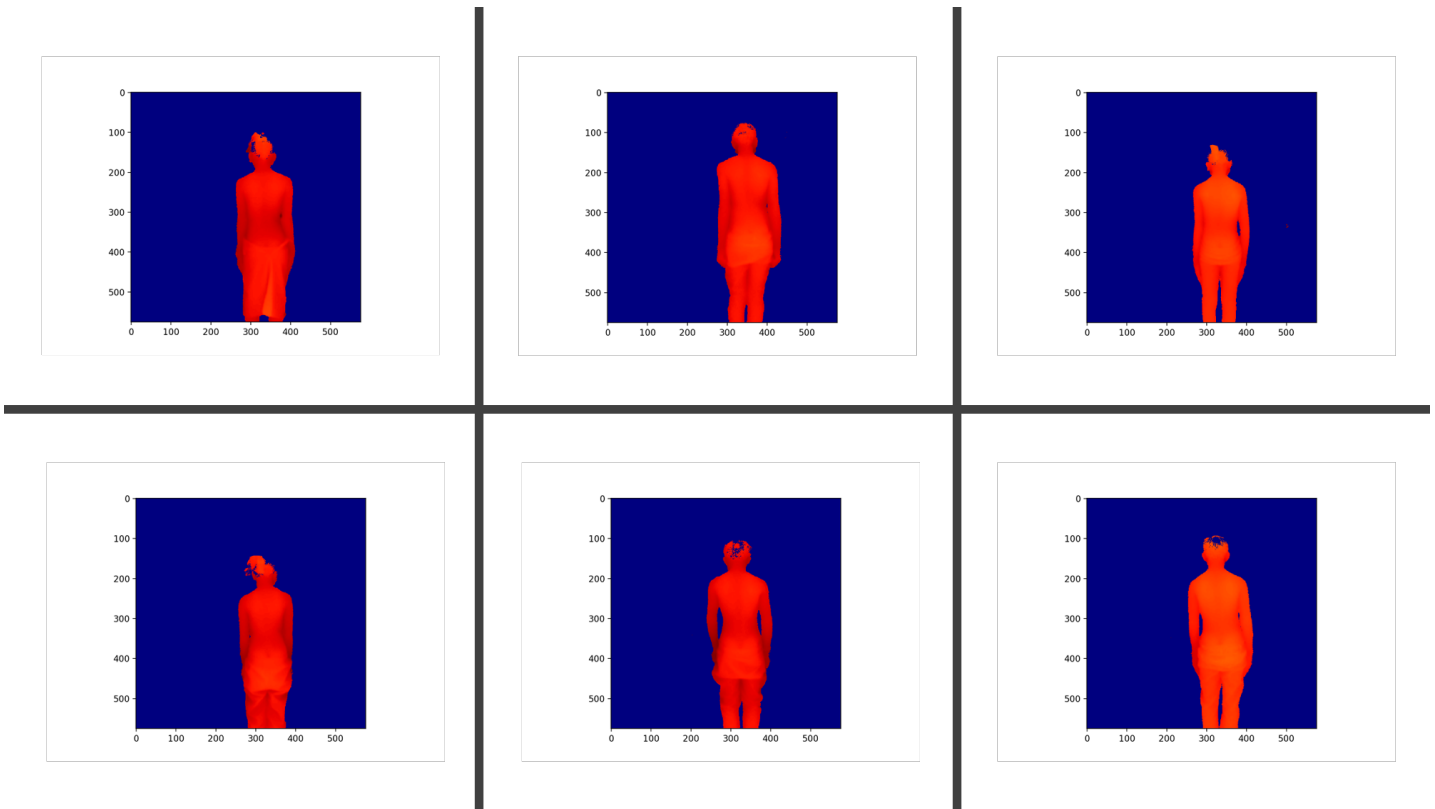


Figure 6: Visualized heat map from a depth image.

In this project, the pixel value of depth images varied from 0 to 13824, which meant the depth images contained objects or background from 0 to 13.8 meters from the depth sensor. To standardize the depth images, the medical staff in photo lab controlled the distance from the back

of the patients to the depth sensor to be 1.2 to 1.8 meters. Therefore, the pixel values on the back of the patients varied from 1200 to 1800. We also wrote a filtering program and checked manually to make sure that all the pixels on the back of the patients fell into this range and ignorable pieces of background fell into this range. Some examples of filtered results that were chosen at random are shown in figure 7 below. The pixels with value outside  $[1200,1800]$  are filtered with blue while the pixels on the back of the patients that fell in  $[1200,1800]$  are in red.



*Figure 7: 6 random depth images after filtering using range  $[1200,1800]$ .*

We had checked image by image that for every depth images in this project, the pixel values for the back of the patients fell in the range  $[1200,1800]$ .

Moreover, to standardize the RGB-D images before inputting to deep learning models, we wrote programs to get some statistical information about the RGB-D images. The statistics of RGB images are shown in table 2 below.

Attributes/Channels	R	G	B
Mean	124.033	125.955	137.415
Standard Deviation	50.665	45.107	46.480
Maximum	255	255	255
Minimum	0	0	0

*Table 2: Statistics of RGB images.*

For depth images, because the variance of whole images was too large while the pixels with large values were not on the back of the patients. Therefore, we measured the statistics in 2 ways. In the first way, we only considered the pixel values that fell in range [1200,1800]. In the second way, we considered all the pixel values on depth images. The results are shown in table 3 below.

Attributes/Method	Back	All
Mean	1345.884	2632.375
Standard Deviation	60.431	2496.437
Maximum	1800	13824
Minimum	1200	0

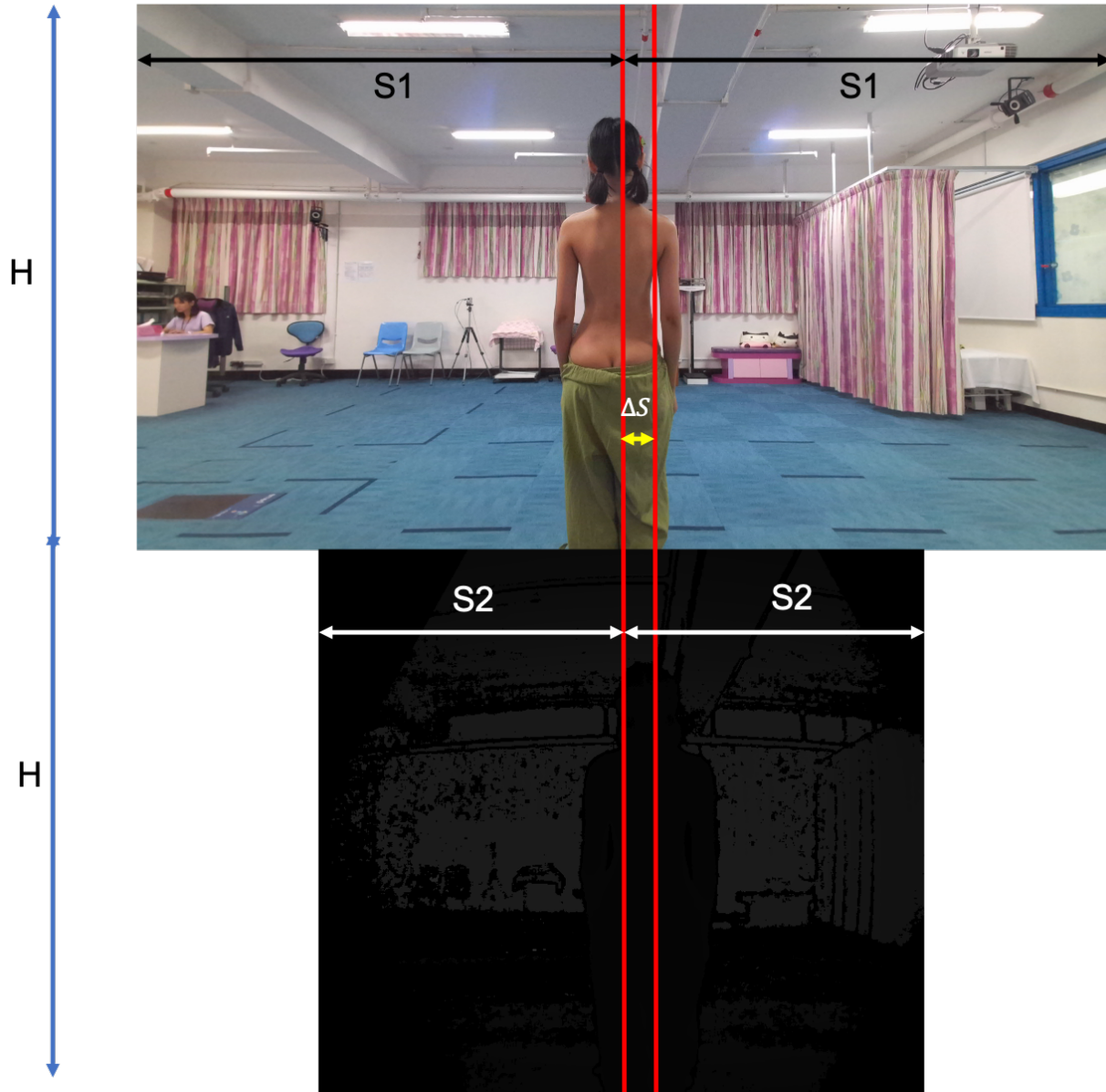
*Table 3: Statistics of depth images over 2 measurements.*

All these statistics were computed after the alignment of RGB-D images, which will be covered in next subsection.

#### **4.1.3. Alignment of RGB-D images**

Alignment was an important issue in this project. The initial resolution of the RGB images are  $1920 \times 1080$ , while the initial resolution of the depth images are  $640 \times 576$ . Depth images had some distort when captured, which was eliminated by us using the SDK provided by Microsoft. Therefore, without any distort, the difference between a RGB image and the

corresponding depth image was a horizontal translation induced by the distance between the RGB sensor and the depth sensor on Azure Kinect DK shown in figure 8 below.



*Figure 8: Horizontal translation between RGB images and depth images.*

To find the horizontal translation ( $\Delta S$  shown in figure), we checked the actual distance between the RGB sensor and depth sensor on Azure Kinect DK and computed the mapping from



the RGB coordinate system to depth coordinate system using the linear transformation equation below, where  $[x', y']^T$  meant the coordinate on depth images and  $[x, y]^T$  meant the coordinate on RGB images.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = F\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Then we first aligned them vertically. We resized the RGB images to  $1024 \times 576$  to match the height of the depth images. After examining the actually distance and calculations, we cropped 185 pixels horizontally from the left and 263 pixels horizontally from the right of the RGB images.

Then we cropped 32 pixels horizontally from the left and right of the depth images. By doing this data preprocessing, we got aligned RGB-D images with resolution being  $576 \times 576$ .

The reason for us to crop the RGB-D images to be square was that, because of the environment limits in the photo lab of DKH, the staff could not fix the depth camera after putting it vertically. By putting it vertically, the back of the patient could occupy more area of the image, which would potentially improve the performance of our deep learning model. However, this option was rejected by the medical staff because of the inflexibility in the photo lab. Therefore, to reduce some noise from the images, we cropped some area from both sides of the images, which would preserve the complete back of the patients and remove some background.

#### 4.1.4. Landmarks on RGB-D Images

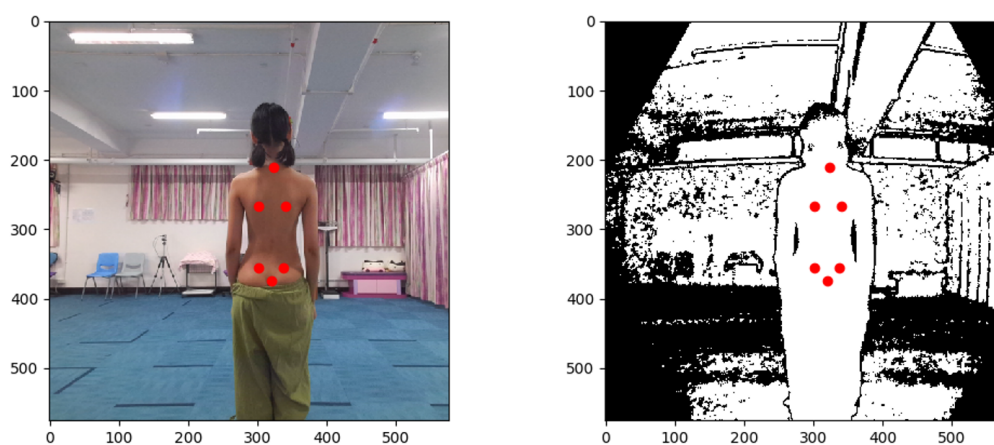
The landmarks for a RGB-D image were stored in a *.txt* file with 6 lines. As shown in figure 9 below, 2 floating point numbers ranged from 0 to 1 will be separated by a tab in each line.

```
0.4966128191766545  0.3540315106580167
0.47420531526836895 0.4448563484708063
0.5190203230849401  0.4457831325301205
0.4726420010422095  0.6190917516218721
0.5138092756644086  0.6209453197405005
0.4898384575299635  0.6515291936978684
```

*Figure 9: One of the anatomical landmarks in a .txt file.*

The 2 numbers in each line showed the coordinate of one anatomical landmark. With the first number times the width of the RGB images (1920) being the actual x-coordinate and the second number times the height of the RGB images (1080) being the actual y-coordinate.

If we plot the landmarks to RGB-D images, the effect is shown in figure 10 below.



*Figure 10: Landmarks on one of the RGB-D images.*

#### **4.1.5. X-ray Images and Landmarks**

X-ray images were taken by several X-ray machines due to the medical condition. As a result, the resolutions of the X-ray images varied. Therefore, we needed to align all the X-ray images. The landmarks on X-ray images were mainly used to align all the X-ray images because the landmarks revealed the region of the spine curves. However, due to the time limit of this project and limited medical condition, the medical staff were unable to label all the 6 anatomical landmarks on X-ray images. Instead, after discussion, the medical staff decided to only label C7 and Sacrum (the top and bottom landmarks) for X-ray images, which showed the vertical region where the spine curves spanned.

X-ray images could be treated both as 1-channel grayscale images and 3-channel RGB images. In both cases the pixel values fell in the range [0,255].

More details will be covered in section 4.3.1.

## 4.2. Landmark Detection

### 4.2.1. The High-Resolution Networks

The High-Resolution networks (HRNet) is a state-of-the-art deep learning model for pixel-level classification tasks such as landmark detection, object detection and object segmentation [5]. Ever since its invention in 2019, lots of projects including facial landmark detection and human pose estimation have been adopted it as the main architecture and achieved good performance.

It has also achieved good performances in 2 tasks (Keypoints Detection and Object Detection) given in the MSCOCO competition, which is regarded as one of the most authoritative competitions in computer vision area [5].

Due to its excellent performance on similar projects about landmark detection, we chose this framework for our project. Our project shared similar nature with those projects that had already been tested on the HRNet and only 6 landmarks were required to be detected.

The architecture of the HRNet is shown in figure 11 below.

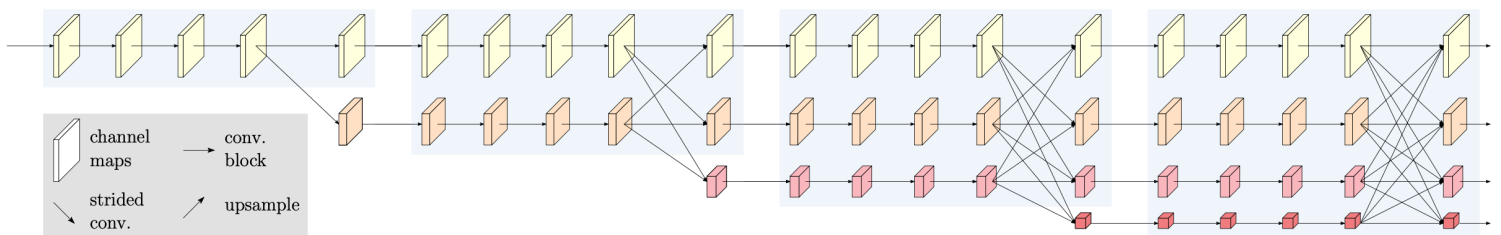


Figure 11: Architecture of the HRNet [5].

As shown in figure 11, the HRNet keeps a high-resolution representation (feature maps) during the forward pass, which is one of the big innovation that makes the HRNet successful. Unlike traditional framework such as ResNet or VGG which take high-resolution input and

generate low-resolution output, HRNet keeps a high-resolution representation to avoid potential information loss induced by downsample [5].

Moreover, many subnetworks shown in the figure reveal another big point that makes the HRNet better, which is multi-scale fusion. The HRNet fuses the feature maps in the same depth with different scales in parallel subnetworks. Multi-scale fusion strengthens the representation power of the feature maps in every depth level. Model becomes more robust regarding objects with different scales [5].

In this stage, the HRNet took  $576 \times 576 \times 4$  RGB-D images as input and generated  $144 \times 144 \times 6$  heat maps as output.

#### **4.2.2. Dataset Split**

In stage 2, out of the 560 full data samples, we randomly chose 520 full data samples as training dataset while the remaining 40 full data samples as validation dataset. The 67 RGB-D images without corresponding X-ray images served as testing dataset.

We did not apply cross-validation strategy because our data samples, compared to other landmark detection tasks, were not enough.

#### **4.2.3. Data Normalization**

Data normalization is widely used in all kinds of supervised learning tasks to map the distribution of data roughly to a standard Gaussian distribution. To achieve this goal, Z-Score function shown below will be applied to all the data samples in the dataset, where  $\mu$  is the mean of the dataset and  $\sigma$  is the standard deviation of the dataset.

$$x^* = z(x) = \frac{x - \mu}{\sigma}$$

$\mu$  and  $\sigma$  were computed over all the pixel values respectively on RGB images and depth images (details has been covered in section 4.1.2). As the distributions of RGB images and depth images differed greatly, we applied Z-Score standardization separately on RGB images and depth images. For depth images, we use the mean and standard deviation for all the pixel values instead of only the pixel values on the back of the patients.

#### **4.2.4. Data Augmentation**

Because the number of data samples were not sufficient, to virtually produce more data and make our model more robust, we applied different kinds of data augmentation to our dataset during the training process. The ground truth landmarks were transformed accordingly.

Application of data augmentation also helped to avoid overfit problem.

All the data augmentation policies were associated with a probability, meaning that even for the same data sample, in different iterations during training, different data augmentation policies would be applied.

These data augmentation policies made our model capable to handle the abnormal cases that could happen in the real-time cases.

##### **4.2.4.1. Rotation**

We randomly rotated our RGB-D images by  $-30^\circ$  to  $30^\circ$  with probability being 0.6. The landmarks were transformed accordingly.

##### **4.2.4.2. Horizontal Flip**

We randomly flipped our RGB-D images horizontally with probability being 0.6. The landmarks were transformed accordingly.

##### **4.2.4.3. Translation**

We randomly translated our RGB-D images in the box range  $[-200,200]$  pixels with probability being 0.6. The landmarks were transformed accordingly.

The box range  $[-200,200]$  made sure that the back of the patients would not be split into multiple parts after translation.

##### **4.2.4.4. Rescaling**

We randomly rescaled our RGB-D images with scale factor in range  $[0.75,1.25]$  with probability 0.6. The landmarks were transformed accordingly.

#### 4.2.4.5. Depth Offset

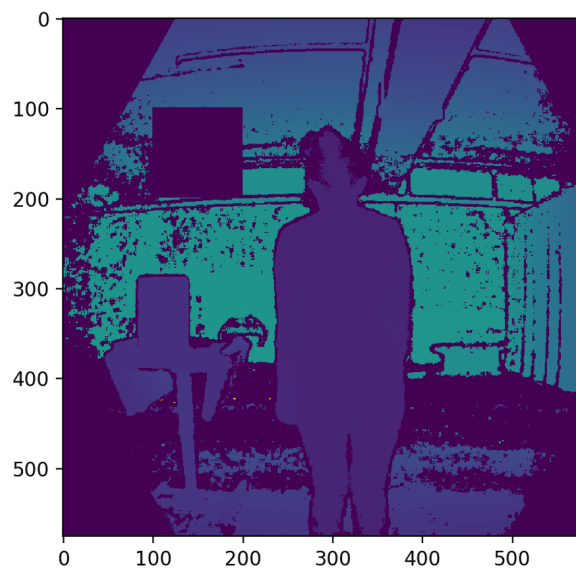
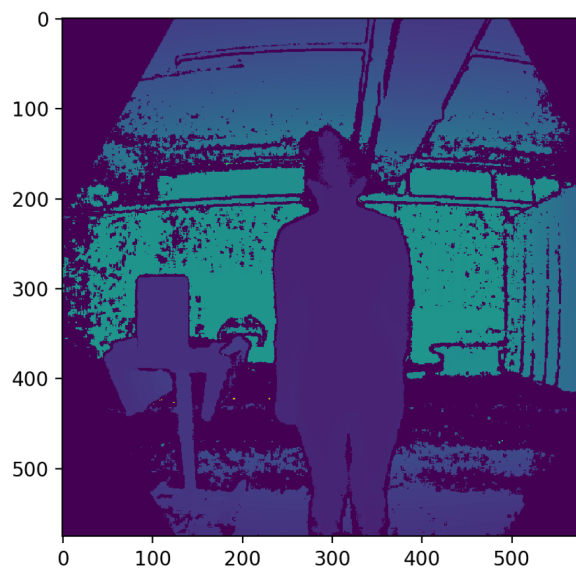
We randomly applied depth offset in range  $[-1000,1000]$  to the depth images with probability being 0.6. The landmarks were not affected.

Depth offset basically means adding a randomly picked value to all the pixel values in depth images. As the pixel value measures the distance in reality, adding some value to the whole depth image is equivalent to moving the whole scene forward or backward with respect to the depth camera.

#### 4.2.4.6. Random Erasing to Background

We randomly added random noise to the background of RGB-D images with probability being 0.6. The landmarks were not affected.

As shown in figure 12 below, we added some random noise to the background of the RGB-D images (the black box on the right image). As all the pixels with value fell in range  $[1200,1800]$  were likely to be on the back of the patients, we only applied this augmentation to those pixels with value outside  $[1200,1800]$ .



*Figure 12: Visualized depth image (left) and Visualized depth image with random noise on background (right).*

#### **4.2.5. Training Strategies**

The model was trained for 60 epochs in total. The initial learning rate is 0.0001 and would be reduced to 0.00001 and 0.000001 respectively at epoch 30 and epoch 50.

When loading the data, the batch size was 8 because of the memory limit of the GPU farm. In each epoch, the training dataset would be shuffled to make the sequence of loading data different.

In the backward pass, Adam optimizer instead of other popular optimizers such as SGD or RMSprop was used to obtain a more stable loss decrement curve. The weight decay and momentum for the optimizer were both set to 0.

Finally, the model would be validated on the validation dataset at the end of each epoch in order to avoid the overfit problem.

#### **4.2.6. Ground Truth**

The 6 anatomical landmarks were transformed to 6 2D Gaussian heat maps centered at each landmark with standard deviation being 1.5, which would be stacked to a 6-channel ground truth heat map.

At the end of the forward pass, this 6-channel ground truth heat map would be fed together with the output heat map of the HRNet to the loss function to invoke the backward propagation. One example of the 6 heat maps are shown in figure 13 below. Each bright point stands for the location of a landmark, which is the peak of 2D Gaussian distribution. All other areas are in purple because they are relatively far from the peak thus have low values.

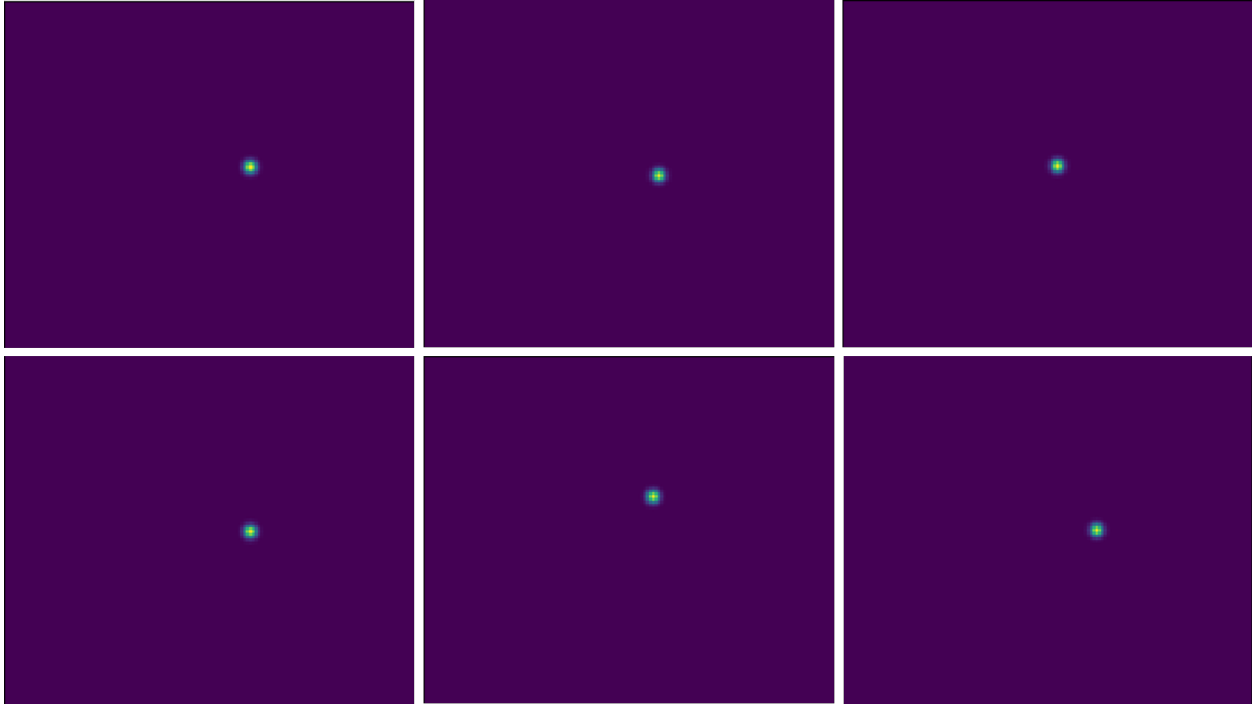


Figure 13: Example for the 6 ground truth heat maps.

#### 4.2.7. Loss Function

Our model was optimized against the mean squares error loss (MSELoss) or squared L2 distance. The formula of MSELoss is given below.

$$M, GT \in \mathbb{R}^{m \times n \times c}$$

$$MSELoss(M, GT) = \frac{\sum_{k=0}^{k < c} \sum_{i=0}^{i < m} \sum_{j=0}^{j < n} (M_{i,j,k} - GT_{i,j,k})^2}{mnc}$$

As shown in the formula, the MSELoss is computed as the mean value of the squared L2 distance between every pair of corresponding pixels on the input matrix  $M$  and ground truth matrix  $GT$ .

This loss function is widely used for landmark detection tasks because of its good ability to penalize pixel-level mismatch.



#### **4.2.8. Performance Metrics**

We numerically measured the performance of our model using the mean MSELoss on the testing dataset. The less the value of the mean MSELoss on the testing dataset, the better the model.

On top of that, we also asked the professional medical staff to manually review the output of the model on the testing and validation set in case of any problems. So far, we have received positive feedbacks about the output.

#### **4.2.9. Result**

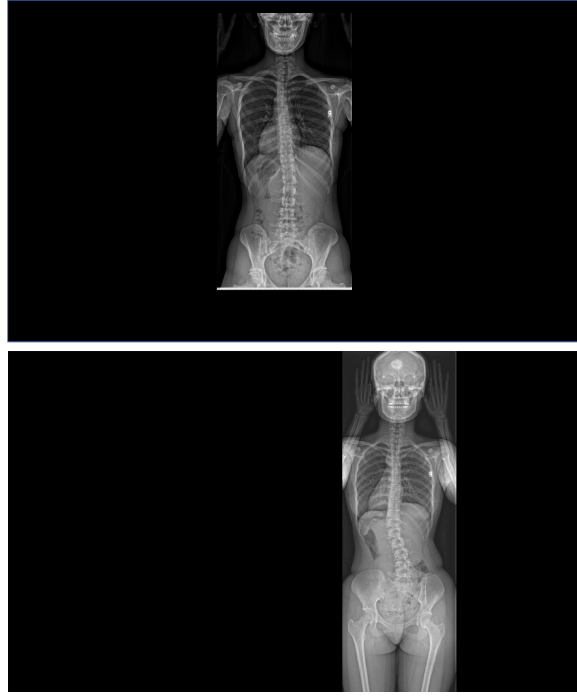
We achieved  $4.747 \times 10^{-5}$  in mean MSELoss on the testing dataset. More comparison and experiments will be covered in section 5.1.

The result landmarks visualized on RGB-D images will be shown in table 21 in the appendix.

### **4.3. X-ray Synthesis**

#### **4.3.1. Data Alignment**

As mentioned before, the X-ray images were taken using several X-ray machines due to the medical condition in the photo lab. Therefore, the resolution of the X-ray images varied greatly. To make things worse, it seemed that the medical staff did not follow the standard procedures to take X-ray images. For example, as shown in figure 14 below, the upper image has the patient roughly at the center of the image while the lower image does not. Additionally, the upper image does not contain the full head and the thigh of the patient while the lower image does. Similar issues occurred frequently in the X-ray images, making it hard to do accurate image-to-image translation as the pixels were not aligned.



*Figure 14: 2 unaligned X-ray images.*

To address the problem, we made use of the 2 anatomical landmarks (C7 and Sacrum) to align all the X-ray images. Medically speaking, C7 and Sacrum can be respectively treated as the start point and the end point of the spine. Therefore, C7 and Sacrum can collaboratively show the vertical area that the spine curve lies in. However, because of the time limit, the medical staff were not able to label all the 6 anatomical landmarks, making it impossible for us to accurately align the X-ray images horizontally. Therefore, we could only made an approximate horizontal alignment.

Suppose the coordinates of 2 anatomical landmarks were denoted by  $(x_1, y_1)$  and  $(x_2, y_2)$ , and the resolution of the X-ray images were denoted by  $W \times H$ . Vertically, we chopped off the areas represented by  $y < \lfloor y_1 - \frac{H}{150} \rfloor$  and  $y > \lfloor y_2 + \frac{H}{150} \rfloor$ . Assume the new vertical height was  $H' = \lfloor y_2 + \frac{H}{150} \rfloor - \lfloor y_1 - \frac{H}{150} \rfloor$  and pivot was defined as  $p = \lfloor \frac{x_1 + x_2}{2} \rfloor$ . Then horizontally, we chopped off the areas represented by  $x < \lfloor p - \frac{H'}{2} \rfloor$  and  $x > \lfloor p + \frac{H'}{2} \rfloor$ . After that, the new

horizontal width of the new image  $W' = \frac{H'}{2}$ . Finally, we used anti-alias technology to resize the images to  $128 \times 256$ .

After this alignment, all the X-ray images contained only the back of the patients. Nearly all other unnecessary areas had been removed. For the sake of conciseness, we manually checked all the X-ray images to guarantee all of them containing the appropriate area.

We also did the same procedures on the RGB-D images to roughly align the X-ray images with the RGB-D images to facilitate image-to-image translation. However, accurate alignment between these 2 datasets were not feasible because the body positioning of the patients would be different when taking the RGB-D images and X-ray images.

An aligned data sample is shown in figure 15 below.



*Figure 15: An aligned data sample containing a RGB image (left), a depth image (middle), an X-ray image (right) and anatomical landmarks (not shown).*

#### **4.3.2. The *pix2pix* Model**

The *pix2pix* model is a CGAN-based, image-to-image translation framework [7]. Unlike traditional GAN which takes random noise as input [6], CGAN takes a pre-defined data as input, which reinforces the power of synthesis of GAN. In *pix2pix* model, images are taken as input to

generate synthetic images. Ever since its invention in 2018, it has been used in many image-to-image translation tasks such as image style transformation, image coloration, image enhancement, etc. [7]. One of the tasks that had already been carried out using this model before our project was the one carried out by Brian et al. in 2018 as described before [8]. In his project, he used surface geometry of the patients and some landmarks to synthesize the X-ray images of the patients, which was similar to our project as the RGB-D images also contained surface geometry information of the back of the patients. Brian and his team got acceptable result in his project [8].

Due to its good performance on similar projects, we chose *pix2pix* model as the main architecture of this stage.

In this project, the *pix2pix* model took 10-channel images as input. Each 10-channel image consisted of a 3-channel RGB image, a 1-channel depth image, and 6 1-channel heat maps generated using the same method presented in section 4.2.6 using 6 anatomical landmarks on RGB-D images. The model generate 3-channel X-ray images. The resolution of all the images including heat maps were  $128 \times 256$ .

#### **4.3.2.1. Generator**

Unlike the original paper of *pix2pix*, U-Net was not chosen in our project. Instead, we chose ResNet with 9 ResNet blocks as the backbone of the generator. The reason for us to abandon U-Net was that the U-Net family only support square images as input and output. It was true that we could use padding to transform the rectangle images into square images, which made it possible for U-Net to come into use. However, later experiments showed that it was not a good choice. More details about the experiments will be covered in section 5.2.

On the other side, ResNet family supported either square images or rectangle images. Moreover, ResNet showed a good adaptation on our rectangle training dataset. To summarize, ResNet showed its superiority over U-Net in our project. Hence, ResNet were chosen.

On top of that, we added some dropout layers with probability being 0.5 inside residual blocks to avoid overfit problem, which was effective as shown in the experiments later. More details about the experiments will be covered in section 5.2.

#### **4.3.2.2. Discriminator**

We used  $70 \times 70$  PatchGAN proposed in the original paper as the backbone for our discriminator [7]. This backbone would take the synthetic images or real images as several  $70 \times 70$  small patches to judge whether the input image was fake or real. The advantages of this architecture were great. Firstly, the input of the discriminator was reduced in size, accelerating the training speed [7]. Secondly, as the generator was a pure convolutional network without any fully connected layers, the resolutions of input and output images were not limited. Then if the discriminator processed images in  $70 \times 70$  small patches, the resolutions of the input images were also not limited, making it easier to try different image resolutions [7].

#### **4.3.2.3. GAN Mode**

In this stage, conditional Least Squares GAN (CLSGAN) was adopted. LSGAN which used least square error as the loss function stabilized the training process greatly compared to Vanilla GAN loss used in ordinary GAN.

#### **4.3.3. Dataset Split**

In stage 3, out of the 560 full data samples, we randomly chose 520 full data samples as training dataset while the remaining 40 full data samples as validation dataset. The testing dataset was the same as the validation dataset.

We did not apply cross-validation strategy because our data samples, compared to other image-to-image translation tasks, were not enough.

#### **4.3.4. Data Normalization**

##### **4.3.4.1. Contrast Stretching on X-ray Images**

As the X-ray images were taken by several X-ray machines, the contrast and brightness of the X-ray images also differed apparently. If we ignored this issue, it was a potential risk for our model to generate some X-ray images with extreme contrast or brightness, which were hard for human beings to study. To make thing worse, it was possible for the model to learn to

determine the contrast and brightness of the synthetic X-ray images by RGB-D images, which was clearly not a necessary thing we wanted the model to learn.

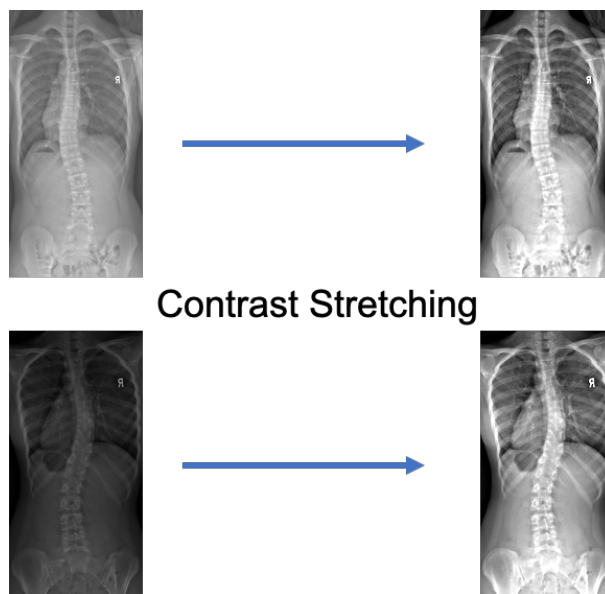
To tackle the problem, we applied contrast stretching on all the X-ray images before they were fed to the model. Contrast stretching is an image enhancing algorithm designed to enhance the contrast of an image. The core of contrast stretching is linearly mapping the intensity of an image to the full range which is usually [0,255]. To avoid the extreme case where the maximum or minimum intensity was too far away from the majority of the intensities, we applied contrast stretching based on the 2nd and 98th percentile of the images. Suppose the 2nd and 98th quantiles of the images were denoted by  $p_2, p_{98}$ , and the target intensity range was

$[r_{\min}, r_{\max}] = [0, 255]$  the mapping function was given by

$$x^* = (x - r_{\min}) \frac{p_{98} - p_2}{r_{\max} - r_{\min}}.$$

By rescaling the intensity of all the images, the brightness mismatch was also solved.

As shown in figure 16 below, contrast stretching had a good effect to transform the X-ray images with abnormal contrast or brightness to high-quality X-ray images. Most importantly, contrast stretching made all the X-ray images roughly have the same contrast and brightness, which facilitated the model to learn a data distribution in a fixed range.



*Figure 16: X-ray images before and after contrast stretching.*

#### **4.3.4.2. Further Normalization**

After contrast stretching, the range of the intensity of X-ray images were in range  $[0,255]$  represented by 8-bit unsigned integer. However, for convolutional neural network, it had been proved to be a good technique to normalize the input and output images into a small range which was usually  $[0,1]$  or  $[-1,1]$ . In this stage, all the images including RGB images, depth images, and X-ray images were linearly normalized into the range  $[-1,1]$ .

#### **4.3.5. Data Augmentation**

The data augmentation policies were similar to that in stage 2 with some adjustments. When training a GAN, usually a great amount of data samples would be needed. However, we only had 520 training samples at that time, which was especially small. Therefore, data augmentation was extremely important to strength the representation power of our model and avoid overfit. Hence, we increased the probability of each data augmentation policy.

Moreover, as the *pix2pix* was expected to achieve the best performance when the input images and output images were aligned, therefore, for spatial data augmentation, RGB-D images, X-ray images and anatomical landmarks would be transformed at the same time with the same parameters.

##### **4.3.5.1. Rotation**

We randomly rotated our RGB-D images and X-ray images by  $-30^\circ$  to  $30^\circ$  with probability being 0.65. The landmarks were transformed accordingly.

##### **4.3.5.2. Horizontal Flip**

We randomly flipped our RGB-D images and X-ray images horizontally with probability being 0.65. The landmarks were transformed accordingly.

##### **4.3.5.3. Translation**

We randomly translated our RGB-D images and X-ray images in the box range  $[-30,60]$  pixels with probability being 0.65. The landmarks were transformed accordingly.

#### 4.3.5.4. Rescaling

We randomly rescaled our RGB-D images and X-ray images with scale factor in range  $[0.75,1.25]$  with probability 0.65. The landmarks was transformed accordingly.

#### 4.3.5.5. Depth Offset

We randomly applied depth offset in range  $[-1000,1000]$  to the depth images with probability being 0.6. The landmarks were not affected.

### 4.3.6. Training Strategies

We trained the model for 200 epochs. In the first 100 epochs, the learning rate was 0.0002. In the second 100 epochs, the learning rate was reduced linearly from 0.0002 to 0.

When loading the data, batch size was 1 because instance normalization layer worked better than the batch normalization layer in *pix2pix* model. In each epoch, the training dataset would be shuffled to make the sequence of loading data different.

In the backward pass, Adam optimizer instead of other popular optimizers such as SGD or RMSprop was used to obtain a more stable loss decrement curve. The weight decay and momentum for the optimizer were set to 0 and 0.5 respectively.

Finally, the model would be validated on the validation dataset at the end of every 10 epochs in order to avoid the overfit problem.

### 4.3.7. Loss Function

The loss function and was given by the equations below:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cLSGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

As proposed in the original paper, the loss function was composed of the conditional LSGAN loss and a weighted L1 loss, which was different from the original LSGAN [7]. The experiments in the original paper showed that if only the conditional LSGAN loss was used, then


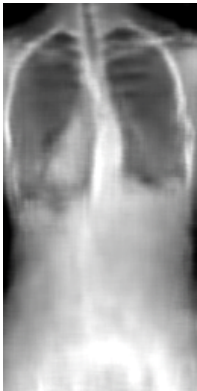




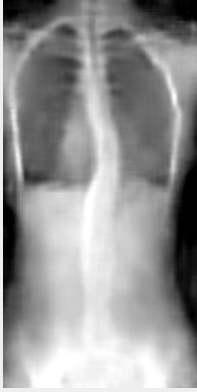

the synthetic images tended to be much sharper and have many artifacts which did not exist in the ground truth. If only the L1 loss was used, then the synthetic images were very blurry. When combining these 2 loss functions, the result seemed to be close to ground truth. This conclusion held true in our projects as proved by our experiments. More details about those experiments will be covered in section 5.2.

In our project,  $\lambda$  was set to be 10.0.

#### 4.3.8. Performance Metrics

It was really hard to find a good numerical metric to judge the performance of our model. The noble Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) were usually ideal choices to measure how close the synthetic image was to the ground truth image. However, in this project, these 2 metrics were not always reliable. For example, some examples of the situations where SSIM and PSNR performed poorly were shown in table 4 below.

	Ground Truth Images	Sample Synthetic Images 1	Sample Synthetic Images 2
Images			
SSIM ↑		0.360	0.153
PSNR ↑		28.128	28.227

	Ground Truth Images	Sample Synthetic Images 1	Sample Synthetic Images 2
Images			
SSIM ↑		0.313	0.231
PSNR ↑		28.295	28.220

*Table 4: Cases where SSIM and PSNR failed to measure the performance of different models in this project.*

From table 4 above, SSIM tended to prefer the blurry images as the blurry images had higher SSIM than the clear images. PSNR had little difference between blurry images and clear images, which was not suitable to measure the performance of the model. Therefore, we abandoned SSIM and PSNR. Instead, 2 other metrics were chosen.

The shape of the spine curve and the clarity of the synthetic images were the main concern in this project. Therefore, we used 2 metrics to measure these 2 factors respectively.

Histogram intersection was used to measure the similarity between the synthetic X-ray images and ground truth X-ray images. This metric calculated the intersection between the histograms of the synthetic X-ray images and ground truth X-ray images. The range of this metric was [0,1], the larger this metric, the more similar the synthetic image was to the ground truth image. This metric was good at measuring the clarity of the X-ray images.

On top of that, we used Image hashing value to measure the structural similarity of 2 images in binary level. Image hashing was a value no less than 0. The smaller the image hashing value was, the closer these 2 images were.

#### **4.3.9. Result**

Because of the limited number of data samples and possible alignment error, the performance of our model was not perfect. However, more than half of the synthesized images had a similar spine curve as their ground truth images and nearly all of the synthesized images were relatively clear for human beings to study. For a numeric measurement, our model had 0.9 in histogram intersection and in 5.675 image hashing.

The synthesized X-ray images will be presented in table 22 in the appendix.

### **5. Experiments**

In this section, we will name experiments with a capital letter and a number. For landmark detection, the experiments will be in the form of  $Ax$ , where  $x$  is an integer. For X-ray synthesis, the experiments will be in the form of  $Bx$  where  $x$  is an integer.

Experiments that were trained using exactly the same methods as described in section 4.2 and 4.3 will be named as A0 and B0. A0 and B0 had the models with relatively the best performances in their stage respectively, which was the reason why we presented the methodologies of these 2 models in section 4.

All the experiments followed the single variable principle, that was, every time we compared 2 experiments, only 1 variable in these 2 experiments would differ. All other variables would be kept the same.

#### **5.1. Landmark Detection**

In this section, the loss curve figures of training process will be shown on the left while the loss curve for validation process will be shown on the right.

The independent variable for training process is the number of iterations while the independent variable for validation process is the number of epochs. As we have 520 training samples and the batch size per iteration was 8, one epoch will have 65 iterations.

##### **5.1.1. Input Data Composition**

###### **5.1.1.1. Motivation**

It seemed intuitive that the depth image contained essential surface geometry of the back of the patients hence could provide sufficient information to the model to predict the locations of 6 anatomical landmarks. However, this intuition still needed to be proved.

Moreover, the RGB images, which seemed not to contain surface geometry might or might not help to detect the 6 anatomical landmarks. An experiment was also needed to verify the validity of using RGB images in the input data.

As a result, we carried out several experiments to find the best composition of the input data.

### 5.1.1.2. Experiment Design

We had RGB images and depth images. Hence, there were 3 ways to compose our input data – pure RGB images, pure depth images, and combined RGB-D images. Then we conducted 3 experiments respectively using pure RGB images, pure depth images and combined RGB-D images as input. We then compared and analyzed the performances of these 3 models in training, validation, and testing process.

We controlled all other parameters, only changed the input composition.

### 5.1.1.3. Result and Analysis

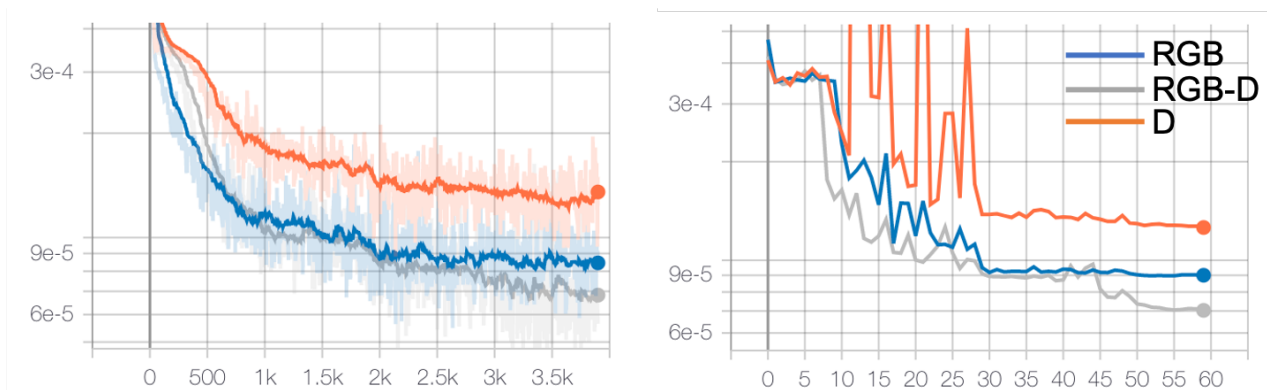


Figure 17: The loss curves for training (left) and validation (right) with 3 data compositions.

Experiment Name	Input Composition	Number of Channels	Mean MSELoss ( $10^{-5}$ )
A1	RGB	3	6.285
A2	D	1	5.816
A0	RGB-D	4	4.747

Table 5: Results of 3 data compositions on testing dataset.

As shown in figure 17, the gray curve which stood for A0, had superior performances compared with the blue curve and orange curve in both training and validation processes. While blue curve which stood for pure RGB images came the second place. In the training loss curve on the left, the gray curve converged to the lowest loss value compared to other 2 curves, showing that using RGB-D images as input helped more for the model to fit the training data than using pure RGB images or pure depth images. Moreover, RGB images could helped more for the model to fit the training dataset than pure depth images.

Coming to the validation loss figure on the right, none of these 3 models suffered the overfit problem because the validation loss was overall slightly less than the training loss at the same time, therefore, the validation curve could be a good measurement of the performance of the model.

From the validation loss curve figure, the gray curve had the lowest final MSELoss on validation dataset compared to other 2 curves and the orange curve came second. The result was the same as the training loss curve. Then theoretically, using RGB-D images would achieve the lowest mean MSELoss on the testing dataset, showing it superiority over 2 other data compositions, which was the truth after we tested these 3 models on our testing dataset.

From table 5, A0 using RGB-D images as input achieved an incredible mean MSELoss on the testing dataset which was lower than A2 and the lower than A1. The result on testing processes was consistent with the training and validation processes.

Therefore, using RGB-D images as input data was the best option to detection the landmarks. RGB images and depth images both contained some information about the surface geometry. But neither of them could provide sufficient information for the model alone. Hence, combining them was a superior choice.

## **5.1.2. Validity of Data Augmentation**

### **5.1.2.1. Motivation**

In this stage, we applied lots of data augmentation. To prove that our data augmentation actually helped to make the model more robust and help the model to avoid overfit problem, we conducted the experiment.

### **5.1.2.2. Experiment Design**

We trained a model without data augmentation policies as described in section 4.2 with all other configurations unchanged. And then we compared the performance of this model with the model trained in A0.

This time, it was not sufficient to only compare the performances of 2 models using numerical metrics (mean MSELoss). Some actual visualized results (i.e. figures with the original image and predicted landmarks) were necessary for us to truly prove the validity of data augmentation. However, the testing dataset were all the normal images with a person standing straight at the center of the images. It was hard for us to tell the difference between the predicted landmarks of these 2 models. Hence, we applied data augmentation to the testing dataset to make the testing data samples more “difficult” only for this experiment, which was not a common practice.

### **5.1.2.3. Result and Analysis**

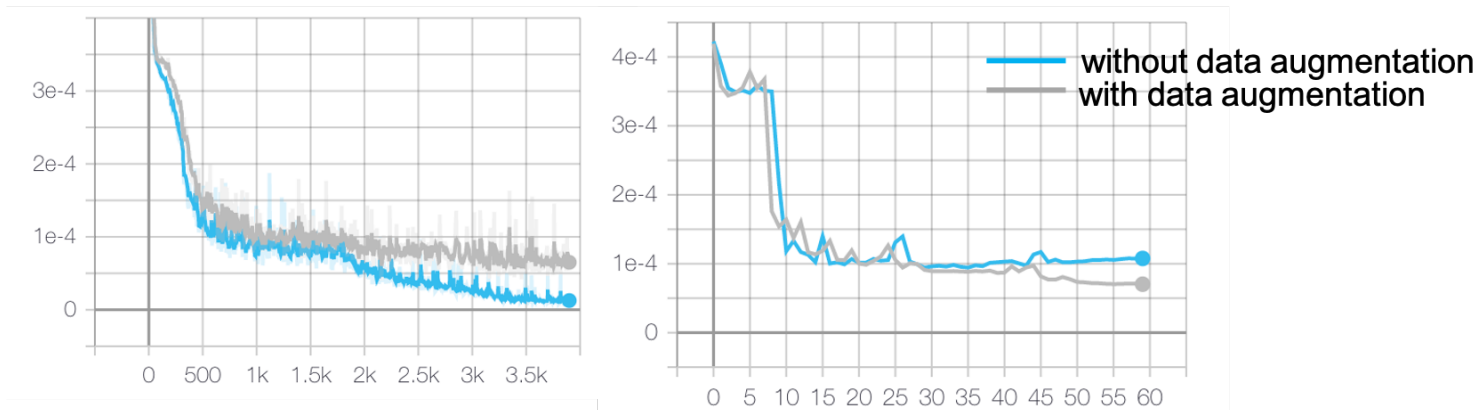


Figure 18: The loss curves for training (left) and validation (right) with or without data augmentations.

Experiment Name	Presence of Data Augmentations on Training Dataset	Presence of Data Augmentations on Testing Dataset	Mean MSELoss ( $10^{-5}$ )
A0	TRUE	FALSE	4.747
A3	FALSE	FALSE	5.798
A0	TRUE	TRUE	5.247
A3	FALSE	TRUE	25.356

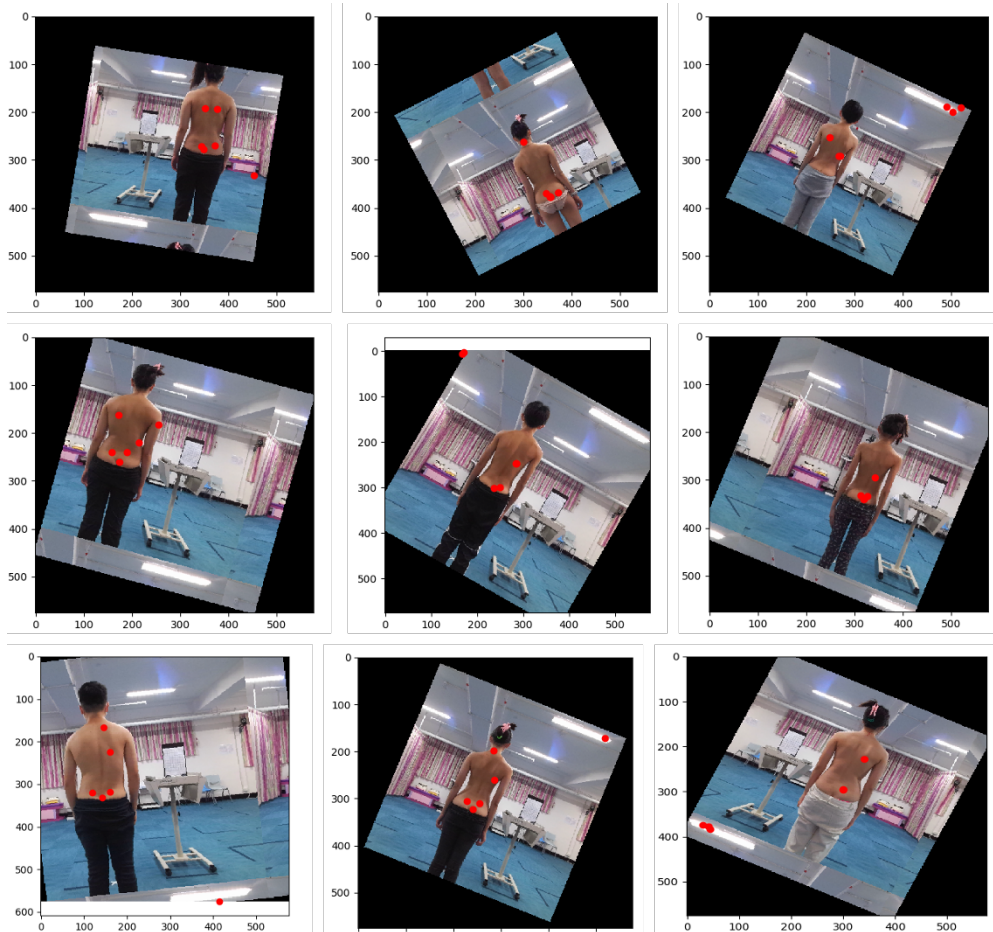
Table 6: Results of model with or without data augmentations on testing dataset.

As shown in figure 18, the blue curve which stood for absence of data augmentations showed a seemingly better fit than the gray curve which stood for presence of data augmentations, which was true. However, starting from the 28th epoch or the 1820th iteration, in A3, the validation loss started to increase while the training loss started to decrease substantially. The validation loss was much higher than the training loss, which was a typical characteristic of overfit problem. In A3, the model fitted the training dataset too much, leading to model to collapse on the validation dataset. Meanwhile, in A0, the validation loss was overall slightly higher than the training loss, meaning that it was unlikely for A0 to suffer the overfit problem. So

far, data augmentation had been proved to be effective in helping the model to avoid the overfit problem.

The testing process shown in table 6 led to a consistent conclusion. A0 with data augmentation showed absolute superiority over A3 without data augmentation. Before applying data augmentation to the testing dataset, because the testing samples were relatively easy, testing loss in both experiments were closed. However, after applying data augmentation to the testing dataset, the testing loss in A3 boosted to an incredibly high level – 25.356, which was almost 5 times the testing loss in A0, indicating that the model in A3 had no power to handle abnormal samples where the patients were not in the normal position.

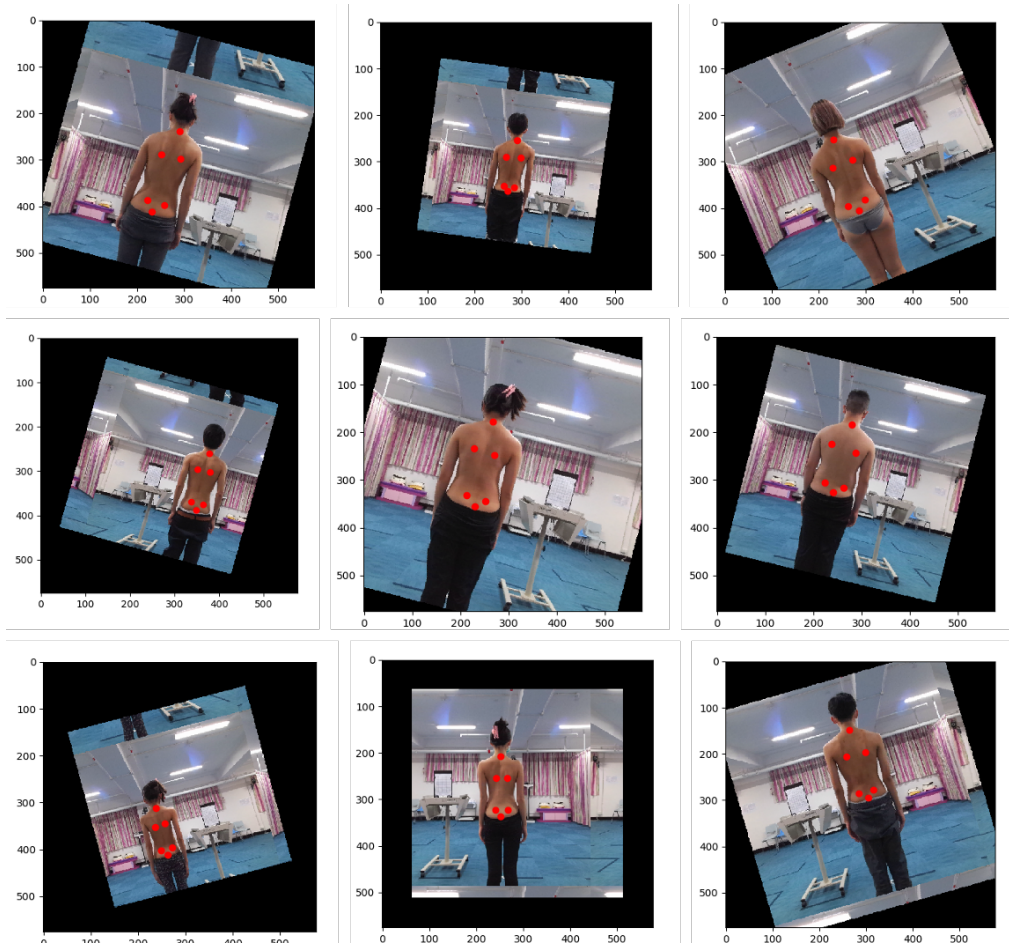
This lemma was showed clearly in the visualized images and predicted landmarks below. The model in A3 couldn't handle the augmented samples very well.





*Figure 19: 9 randomly picked results images and landmarks in A3.*

On the contrary, the model in A0 could handle abnormal samples very well, which is shown in figure 20 below.



*Figure 20: 9 randomly picked results images and landmarks in A0.*

Therefore, application of data augmentations was proved to be a good technique to make the model more robust and avoid overfit problem.

### **5.1.3. Method for Data Normalization**

#### **5.1.3.1. Motivation**

It was a big problem since we first worked on the project. As shown in section 4.1.2, the RGB images and depth images differed greatly in their data distribution. The average value of pixel on depth images was much larger than that on RGB images and so as the standard deviation. Obviously, it was not a wise option to directly train the model using the raw RGB-D images. Data standardization was needed before we input the RGB-D images to the model.

To find the best method to do the data standardization, several experiments were conducted.

In the experiments conducted below, RGB images and depth images would be standardized separately using their own statistics.

### 5.1.3.2. Experiment Design

There are 2 common methods for data standardization. The first one is to scale the data directly on the minimum and maximum pixel values in the dataset (Min-Max). The formula is shown below.

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

The Min-Max standardization will linearly map all the pixel values to [0,1].

Another common method is to apply Z-Score function mentioned in section 4.2.3. The Z-Score standardization will map all the pixel values roughly to a standard Gaussian distribution centered at 0. Obviously, we needed to use experiments to prove which method was superior.

Another issue was that, the data distribution of depth images was too dispersive. As shown in figure 21 below, most of the values falls out of the range [1200,1800]. Then the mean and standard deviation would be greatly influenced by those pixel values. This indicated that applying Z-Score using the mean and standard deviation of the whole dataset might not be a good choice because the Z-Score would then map the pixel values on the back of the patients far from 0, which was the center of the new distribution after standardization.

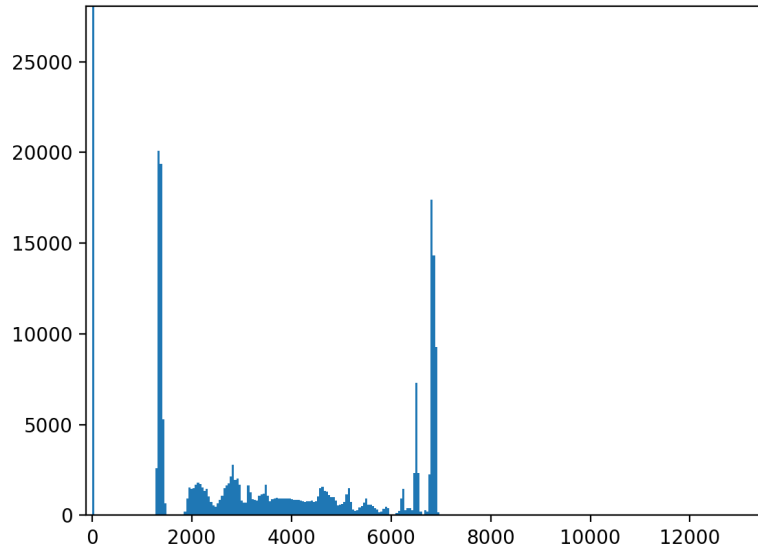


Figure 21: Distribution of the pixel values of the depth images (values that occurred rarely will have a very short bar so that could not be observed).

Combining these 2 issues, 4 experiments were conducted with 1 of which being A0. We set a blank reference A4, where we didn't do any standardization before training. In A5 and A6, we applied Min-Max standardization to RGB-D images while A6 used the minimum and maximum value of the whole dataset and A5 only used the minimum and maximum value on the back of the patients. In A7, we applied Z-Score standardization to RGB-D images with the mean and standard deviation only on the back of the patients.

### 5.1.3.3. Result and Analysis

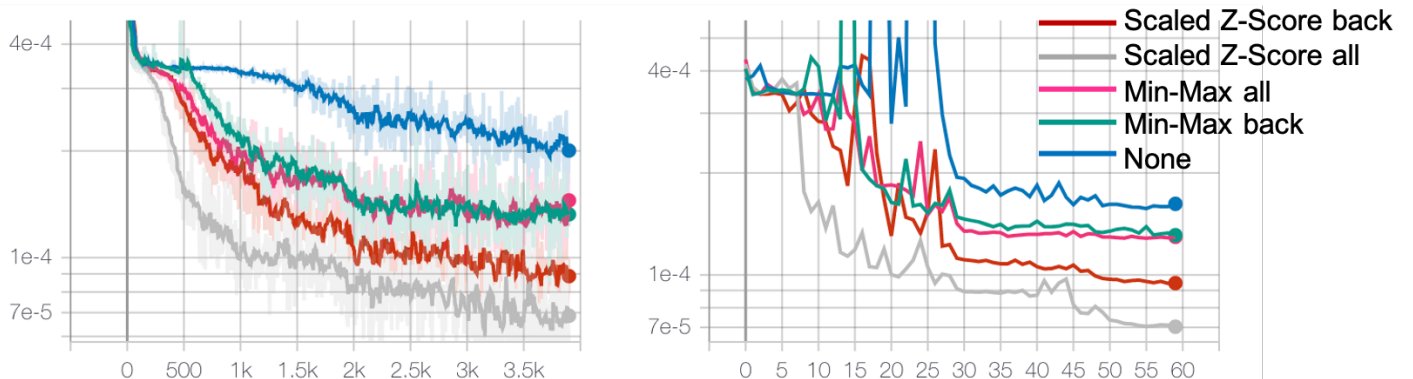


Figure 22: The loss curves for training (left) and validation (right) with different data standardization methods.

Experiment Name	Standardize Method	Mean MSELoss ( $10^{-5}$ )
A4	None	26.744
A5	Min-Max back	19.801
A6	Min-Max all	17.989
A7	Scaled Z-Score back	8.591
A0	Scaled Z-Score all	4.747

Table 7: Results of model with different data standardization methods on testing dataset.

From figure 22 and table 7, we could see clearly that applying Z-Score with mean and standard deviation for the whole dataset dominated all other alternatives in both training and validation processes.

#### 5.1.4. Comparison with Other Models

##### 5.1.4.1. Motivation

The HRNet was regarded as a state-of-the-art deep learning model for pixel-level classification tasks which did well in other projects. However, doing well in other projects did not necessarily mean good performance in this project. Therefore, to support our choice of the HRNet, we compared the performance of HRNet in this project with performances of other models in this projects.

##### 5.1.4.2. Experiment Design

We chose 3 models that were previously widely used in landmark detection tasks: ResNet-101, VGG-16, and Hourglass-104. Then we trained these 3 models on our dataset. These 3 experiments were named A6, A7, and A8.

The training and validation curves for different models differed hugely, which made it not so useful to reveal the performances of different models. Therefore, here we only show the final result measured in mean MSELoss on testing dataset.

### 5.1.4.3. Result and Analysis

Experiment Name	Model	Mean MSELoss ( $10^{-5}$ )
A0	HRNet	4.747
A6	ResNet-101	6.882
A7	VGG-16	21.960
A8	Hourglass-104	8.094

*Table 8: Results of different models on the testing dataset.*

From table 8, it was clear that the HRNet yielded the smallest mean MSELoss on the testing dataset. Therefore, to some extent, we could say the HRNet was the best choice for stage 2 of this project.

However, this analysis was not so robust that could show the superiority of the HRNet over all other existing models because the parameters in different models were very different. It was hard to say that we followed strictly the one variable principle in different experiments with different models.

## 5.2. X-ray Synthesis

### 5.2.1. X-ray Enhancement

#### 5.2.1.1. Motivation

As stated in section 4.3.4.1, the contrast and brightness of X-ray images varied greatly. We wanted the model to generate X-ray images with nearly the same, normal contrast and brightness. Therefore, we carried out contrast stretching on all the X-ray images before training to make them in normal contrast and brightness. However, there were multiple methods to normalize the contrast and brightness of images. Contrast stretching was not the latest and the

best one in usual cases. More advanced methods such as histogram equalization and adaptive equalization were the common choices.

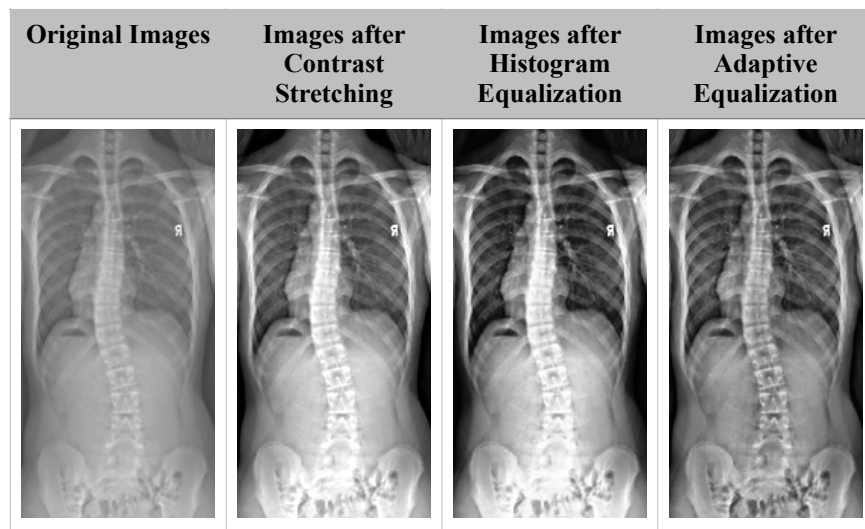
Therefore, we needed to carry out experiments to prove that contrast stretching worked best on our dataset.





### 5.2.1.2. Experiment Design

We enhanced all the X-ray images using contrast stretching, histogram equalization and adaptive equalization respectively to create 3 datasets. Then we compared the images in this 3 datasets after enhancement originated from the same image in the raw dataset to see if the X-ray images were in the nearly the same and normal brightness and contrast. We also plotted the distributions of the images in these 3 datasets to see if the range of intensities were normalized roughly to [0,255].

On top of that, we conducted an experiment B1 to compare with B0. In B1, we did not apply contrast stretching before training. With all other factors unchanged, comparison between the synthetic images in B1 and B0 could reveal the validity of contrast stretching.

### 5.2.1.3. Result and Analysis



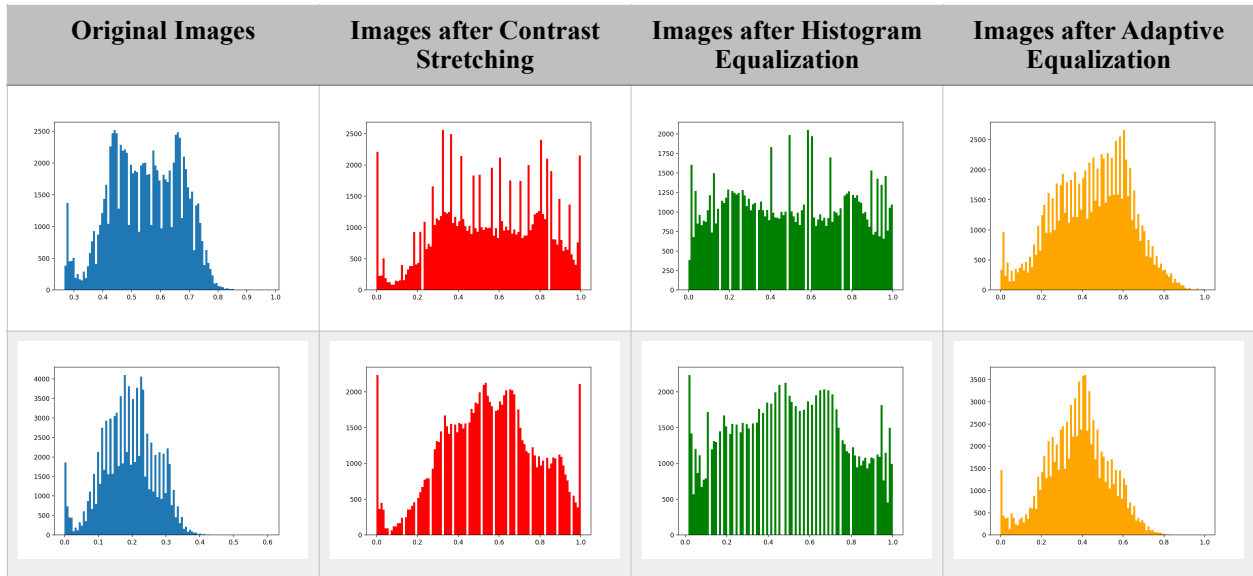
Original Images	Images after Contrast Stretching	Images after Histogram Equalization	Images after Adaptive Equalization
			

*Table 9: X-ray images enhanced by 3 algorithms.*

As shown in table 9 above, all of these 3 algorithm could cause a big improvement on the original X-ray images on contrast and brightness. Images after enhancement were much clearer than the original images for medical staff to study.

However, the contrast of the images generated by histogram equalization was so high that the images looked unnatural. The comparison between the white and black pixels were too sharp which was not an ideal condition for medical staff to study. Therefore, histogram equalization was not suitable for our project.

For contrast stretching and adaptive equalization, the enhanced images looked milder and still good in brightness and contrast. However, a clear difference of brightness could be observed among the images enhanced by adaptive equalization. To explain this phenomenon, we plotted the data distributions of the images. The intensities were scaled with 255 before plotting.



*Table 10: Distributions of X-ray images shown in table enhanced by 3 algorithms.*

From table 10 above, it was clear that adaptive equalization could not normalize the intensities of all X-ray images roughly in the range [0,255]. The upper image had overall higher intensities, making it look brighter while the lower image had overall lower intensities, making it look darker. However, after adaptive equalization, the range occupied by the upper image was still larger than the range occupied by the lower image. The upper image had some intensities approaching 1.0 which was 255 before scaling while the lower image tended not to have intensities over 0.8 which was about 204 before scaling. Therefore, it was natural for the images enhanced by adaptive equalization to have apparent difference in brightness.

Combined these 2 tables above, we could conclude that contrast stretching combined the advantages of histogram equalization and adaptive equalization. After enhancement by contrast stretching, the images were nice looking and not too sharp. Moreover, the images enhanced by contrast stretching did not show apparent difference in brightness.

We did not stop at the input level. B1 without contrast stretching was trained for comparison. Part of the results of B0 and B1 are shown in the table below.















	Synthetic Images in B1	Ground Truth in B1	Synthetic Images in B0	Ground Truth in B0
Sample 1				
Sample 2				
Sample 3				
Mean Histogram Intersection	0.816		0.9	
Mean Image Hashing	6.875		5.675	

Table 11: Synthetic X-ray images and the corresponding ground truth in B0 and B1.

From table 11, we could clearly observe that the contrast and brightness of the synthetic images in B1 varied and did not always match the contrast and brightness of the ground truth

images. On the other side, the images synthesized in B0 had good contrast and brightness which was easy for medical people to clearly study. The synthetic images also matched the ground truth images in B0 in contrast and brightness. This showed that contrast stretching helped to standardize output images.

Numerically speaking, B0 and B1 did not differ greatly in mean L1 distance, indicating that B0 did not perform better than B1 regarding spine curve. However, the difference of mean histogram intersection showed that adding contrast stretching helped a lot to generate clear X-ray images.

## 5.2.2. Avoid Overfit

### 5.2.2.1. Motivation

GAN-based projects tended to need huge amount of data to achieve a good result [6]. However, there were only 520 data samples in our training dataset when we started to write this report. Therefore, theoretically, it was easy for our model to suffer overfit problem on such a small dataset. To avoid this problem, we adopted both data augmentation and dropout layers. However, whether these techniques really helped to avoid the overfit problem needed to be proved by experiments.

### 5.2.2.2. Experiment Design

We carried out 3 experiments B2, B3, and B4 to compare with B0. In B2, only dropout layers technique was used during training. In B3, only data augmentation was used during training. In B4, neither of dropout layers technique nor data augmentation was used during the training process.

### 5.2.2.3. Result and Analysis

Experiments Name	Techniques to Avoid Overfit	Mean Histogram Intersection	Mean Image Hashing
B0	Dropout and Data Augmentation	0.9	5.675
B2	Dropout	0.883	7.125

Experiments Name	Techniques to Avoid Overfit	Mean Histogram Intersection	Mean Image Hashing
B3	Data Augmentation	0.862	9.825
B4	None	0.771	14.375

Table 12: Mean histogram intersection and mean image hashing in B0, B2, B3, B4.

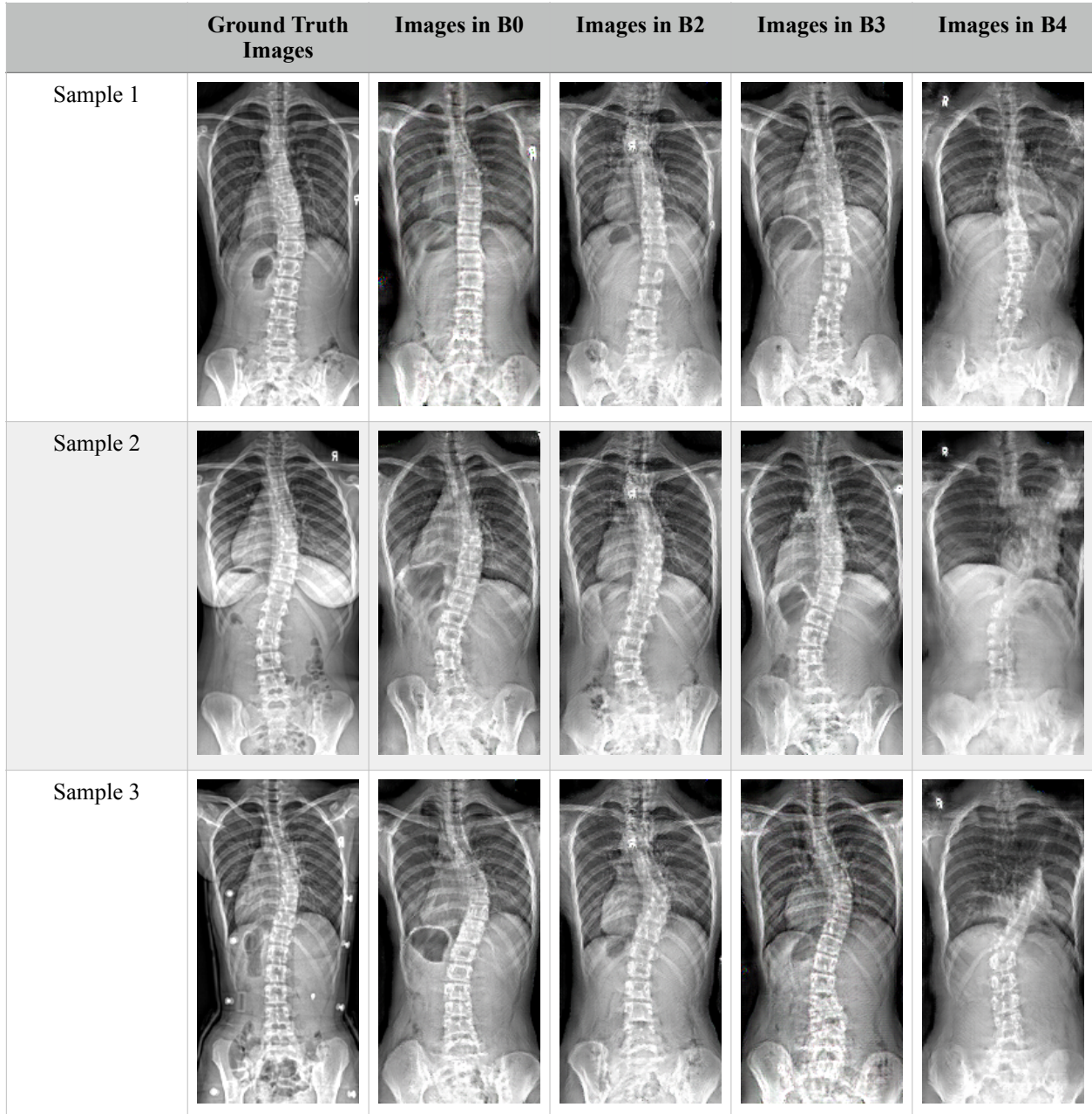


Table 13: Synthetic X-ray images and the corresponding ground truth in B0, B2, B3, B4.

All the training samples in these 4 experiments had good performance where the synthesized images were very closed to the ground truth, meaning that in these 4 experiments, the model fitted the training dataset very well. However, it could be observed clearly that some models, which had good performance on training dataset, performed poorly on testing dataset.

As shown in the table 13, if neither data augmentation nor dropout was used in training, like in B4, the model crashed on the testing dataset. The spine curves completely lost their shape in B4. It could be speculated that B4 suffered great overfit problem as it performed very well on the training dataset.

Similarly, dropout in B2 and data augmentation in B3 helped the model to avoid overfit problem to some extent. When they were combined in B0, the performance seemed to be the best. The numerical metric shown in table also proved that.

Therefore, data augmentation and dropout layer did help the model to avoid overfit problem.

### **5.2.3. Data Alignment**

#### **5.2.3.1. Motivation**

Alignment was an extremely important step in our project. Especially, the X-ray images, before alignment, varied in all kinds of aspects. Before we got the landmarks to align the X-ray images, we also did several experiments using the unaligned X-ray dataset. To show that alignment essentially helped the model to generate better images, we carried out this experiment.

#### **5.2.3.2. Experiment Design**

We used unaligned X-ray images and RGB-D images to training our model in B5. And we compared B5 to B0. We did not calculate the numerical metrics for the contrast between B0 and B5 were too obvious. Instead, we compared the quality and similarity of the synthesized images to see which experiments produced more high-quality X-ray images with correct spine curves and good clarity. Analysis from medical people were also a great reference in this experiment.

### 5.2.3.3. Result and Analysis













	Synthetic Images in B5	Ground Truth in B5	Synthetic Images in B0	Ground Truth in B0
Sample 1				
Sample 2				
Sample 3				

Table 14: Synthetic X-ray images and the corresponding ground truth in B0 and B5.

From the result shown in table 14, it was so clear that, the unaligned X-ray dataset performed poorly on the testing dataset. As shown in the samples of B5, some of the synthetic images did not have a complete shape of person, some of the synthetic images did not have a clear spine curve. While on the other side, after alignment, in B0, the X-ray images synthesized were much more clearer. Moreover, the shape of spine curve had a higher probability to match the shape of spine curves in ground truth X-ray images.

Therefore, after alignment, the performance of our model improved a lot.

#### **5.2.4. Backbone for Generator and Image Resolution**

##### **5.2.4.1. Motivation**

U-Net was the backbone chosen in the original *pix2pix* paper which yielded the best performance [7]. However, this network only support square images. In our project, after we aligned the X-ray images, the resolution was  $128 \times 256$ . However, we could add 2  $64 \times 256$  paddings to the left and right of a rectangle image to transform its resolution into  $256 \times 256$ . Then we could instead use U-Net as the backbone of the generator.

Hence, we conducted several experiments to find the best backbone for the generator in our project.

Moreover, it was argued by the author of the *pix2pix* paper that *pix2pix* model was not good at handling images with large resolution. The default resolution for input and output images were  $256 \times 256$  if square images were used. However, we observed that when the X-ray images were scaled to  $128 \times 256$ , the ground truth images started to become pixelated, for the edges of square pixels could be seen clearly. We were concerned this phenomenon might affect the performance of our model. Therefore, to find the best resolution of input and output images for the model, several experiments with larger resolution input and output images were carried out.

##### **5.2.4.2. Experiment Design**

We denoted the dataset with  $128 \times 256$  images as “Rectangle”, and the dataset with  $256 \times 256$  images after padding from “Rectangle” by “Square”. Further, we created another dataset named “RectangleHD” used the same procedures to crop and align the raw X-ray images

and RGB-D images except that all the images in “RectangleHD” were resized using anti-alias algorithm to  $256 \times 512$ .









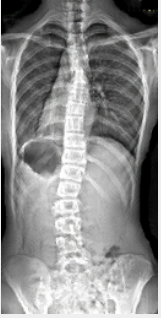







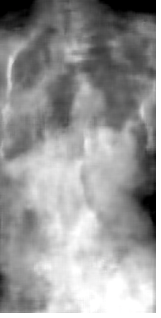













We then trained the model using U-Net 128 and U-Net 256 on “Square” dataset. We trained the model using ResNet with 9 blocks and ResNet with 6 blocks on “Rectangle” and “RectangleHD” datasets respectively.

Again, to measure the performance of our model in each experiment, mean histogram intersection and mean image hashing were used. However, the resolutions of images in different datasets were different, which would affect and introduce bias to the metrics we used. Therefore, to make the measurement unbiased, before measuring the metrics, we resized synthetic images in “RectangleHD” to  $128 \times 256$  and cropped (remove paddings on both sides) synthetic images in “Square” to  $128 \times 256$ . Then all the synthetic images were in  $128 \times 256$ .

#### 5.2.4.3. Result and Analysis

Experiment Name	Dataset	Backbone	Mean Histogram Intersection	Mean Image Hashing
B6	Rectangle	ResNet 6 blocks	0.871	6.0
B0	Rectangle	ResNet 9 blocks	0.9	5.675
B7	RectangleHD	ResNet 6 blocks	0.323	8.125
B8	RectangleHD	ResNet 9 blocks	0.817	9.250
B9	Square	U-Net 128	0.497	6.500
B10	Square	U-Net 256	0.483	6.150

Table 15: Mean histogram intersection and mean image hashing in B0, B6, B7, B8, B9, and B10.

Experiment Name	Synthetic Images, Sample 1	Ground Truth Images, Sample 2	Synthetic Images, Sample 2	Ground Truth Images, Sample 2	Synthetic Images, Sample 3	Ground Truth Images, Sample 3
B6						
B0						
B7						
B8						
B9						





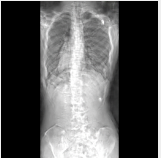

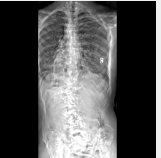

Experiment Name	Synthetic Images, Sample 1	Ground Truth Images, Sample 2	Synthetic Images, Sample 2	Ground Truth Images, Sample 2	Synthetic Images, Sample 3	Ground Truth Images, Sample 3
B10						

Table 16: Synthetic X-ray images and the corresponding ground truth in B0, B6, B7, B8, B9, and B10.

From the result shown in table 15 and 16, it could be observed that U-Net family and “Square” dataset could not produce X-ray images with high quality. There were lots of noise points in the synthetic images in B9 and B10. It could be potentially explained by the intuition that the paddings occupied too much area on the image.

On the contrary, rectangle datasets and ResNet family performed much better. Although in B7, ResNet with 6 blocks showed extremely poor adaptation on the “RectangleHD” dataset, B0, B6 and B8 had relatively good performance both in the clarity of synthetic X-ray images and the shape of the spine curve.

It could be further observed that, the “Rectangle” dataset had an overall better performance than “RectangleHD” using ResNet family as the backbone of the generator. In B8, the synthetic X-ray images were clear though, the shape of the spine curve struggled to match the ground truth compared to B0 and B6. While in B0 and B6, the shape of the spine curve roughly matched with sound truth. It was easy to tell that B0 and B6 were better than B8. However, B0 and B6 had very closed performances so that it was hard to tell which one was better only from the synthetic images.

Nevertheless, from the 2 numerical metrics, it could still indicate that B0 involving ResNet with 9 blocks and the “Rectangle” dataset had best performance both in the clarity of synthetic X-ray images and the shape of the spine curve.

### 5.2.5. Composition of Input Data

### 5.2.5.1. Motivation

Like in stage 2, we needed to find the best data composition of the input. Nevertheless, things became more complicated this time. Our original proposal was to synthesize the X-ray images from the RGB-D images and the anatomical landmarks. However, was it possible that using only the RGB images or depth images would yield a similar or even better result than using all of them? To prove we actually needed the RGB-D images and anatomical landmarks as the input data, we conducted experiments.

### 5.2.5.2. Experiment Design









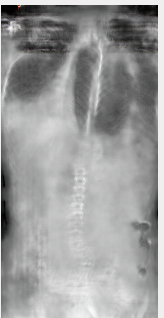















We had RGB images, depth images, and anatomical landmarks as input data. Therefore, there should be 7 ways to compose the input data. As using pure anatomical landmarks as input data made no sense because the shape of the human body was not given to the model, we omitted that experiment.

Therefore, we conducted 5 additional experiments from B11 to B15 to compare with B0. B11 used pure RGB images, B12 used pure depth images, B13 used RGB-D images, B14 used RGB images and anatomical landmarks, B15 used depth images and anatomical landmarks.

### 5.2.5.3. Result and Analysis

Experiment Name	Input Data Composition	Number of Channels	Mean Histogram Intersection	Mean Image Hashing
B0	RGB, D, L	10	0.9	5.675
B11	RGB	3	0.775	11.15
B12	D	1	0.836	10.2
B13	RGB, D	4	0.813	9.425
B14	RGB, L	9	0.826	10.15
B15	D, L	7	0.847	9.4

Table 17: Mean histogram intersection and mean image hashing in B0, B11, B12, B13, B14, and B15 (L stood for anatomical landmarks).

Experiment Name	Synthetic Images, Sample 1	Ground Truth Images, Sample 2	Synthetic Images, Sample 2	Ground Truth Images, Sample 2	Synthetic Images, Sample 3	Ground Truth Images, Sample 3
B0						
B11						
B12						
B13						













Experiment Name	Synthetic Images, Sample 1	Ground Truth Images, Sample 2	Synthetic Images, Sample 2	Ground Truth Images, Sample 2	Synthetic Images, Sample 3	Ground Truth Images, Sample 3
B14						
B15						

Table 18: Synthetic X-ray images and the corresponding ground truth in B0, B11, B12, B13, B14, and B15.

As shown in table 17 and 18 above, only using the RGB-D image plus anatomical landmarks could yield the best performance. None of the remaining experiments could have better synthetic images than B0.

## 5.2.6. Weight of L1 Loss

### 5.2.6.1. Motivation

One of the big innovation in the *pix2pix* model was to combine the cLSGAN loss and L1 loss [7]. As argued by the author, pure cLSGAN loss seemed to produce more clear images with artifacts, while pure L1 loss seemed to produce blurry images with roughly correct color on different regions without artifacts. Hence, combining these 2 losses functions with a weight parameter  $\lambda$  yielded a good improvement on the original model.

In this project, we also need to find the best way to combine these 2 losses functions. It was a really important parameter to tune. Before we found the right value for  $\lambda$ , no clear images could be generated.

### 5.2.6.2. Experiment Design

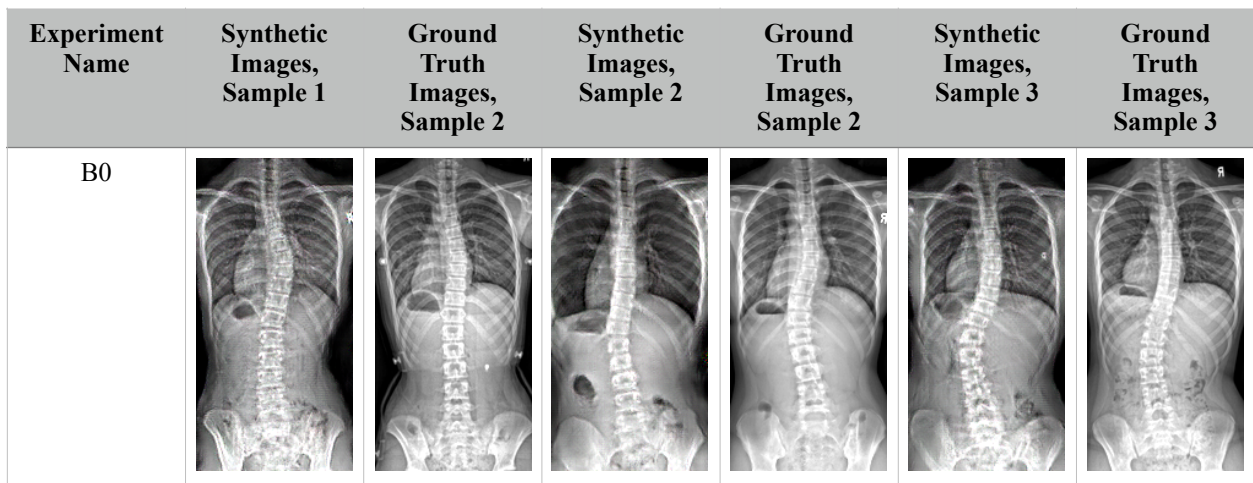
We set 4 additional  $\lambda$  levels: 2.5, 5.0, 15.0, 30.0 for 4 additional experiments B16, B17, B18, B19 to compare with B0 with  $\lambda$  being 10.0.

By setting these  $\lambda$  levels, we could detect roughly the range we should set  $\lambda$  to.

### 5.2.6.3. Result and Analysis

Experiment Name	$\lambda$	Mean Histogram Intersection	Mean Image Hashing
B0	10.0	0.9	5.675
B16	2.5	0.878	9.875
B17	5.0	0.834	9.775
B18	15.0	0.772	6.225
B19	30.0	0.367	6.375

Table 19: Mean histogram intersection and mean image hashing in B0, B16, B17, B18, B19.





















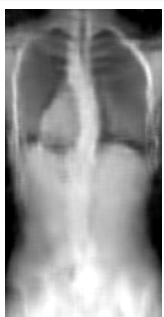

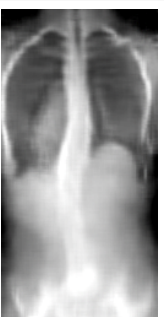

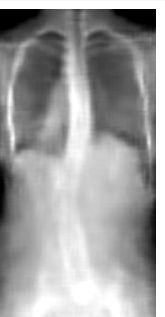

Experiment Name	Synthetic Images, Sample 1	Ground Truth Images, Sample 2	Synthetic Images, Sample 2	Ground Truth Images, Sample 2	Synthetic Images, Sample 3	Ground Truth Images, Sample 3
B16						
B17						
B18						
B19						

Table 20: Synthetic X-ray images and the corresponding ground truth in B0, B16, B17, B18, and B19.

Although it was not clear in numerical metrics shown in table 19, we could see that a high value of  $\lambda$  could lead to blurry images as shown in B19 as shown in table 20, which was consistent to the argument of the author of the *pix2pix* model. However, in B19, the shape of the spine curve was roughly the same as the ground truth, which was better than other experiments. Moreover, while small  $\lambda$  led to good clarity, they tended to fail in the shape of spine curve.

Therefore,  $\lambda = 10$  was roughly an optimal value for  $\lambda$ . Further research could still be done to evaluate the effect of  $\lambda$  on the model.

## 6. Future Works

In this section, we will present several items that we did not handle very well during the project. Therefore, if our project is going to be elaborated in the future, this part will be an important reference.

Firstly, the number of data samples were not enough compared to other similar projects. Actually, when we nearly finished this report, totally over 700 data samples had been collected. However, at that time, all of our experiments were conducted using 520 data samples. To keep the experiment reliable, we decided not to add about 200 additional data samples to our projects. Hence, this is an important aspect that this project can be elaborated in the future. It can be observed in table 22 in the appendix that, so far, the synthetic images are far from perfect. Some of the synthetic images still have wrong shape of spine curves compared to the ground truth. This problem can be potentially explained by the insufficiency of the data samples.

Secondly, the X-ray images were not strictly aligned. We were supposed to have 6 anatomical landmarks on each X-ray images. However, due to the medical condition, time was not sufficient for the medical staff to label all 6 landmarks before the deadline of this project. Instead, they labeled 2 of them for us. Therefore, we could only align all the X-ray images roughly, which would contribute to the error of the model. Moreover, medically speaking, the body positioning of a patient when taking RGB-D images is definitely different from the body positioning when taking X-ray images as the arms will lift up in the latter case while the arms will lower down in the former case. Therefore, it was theoretically not possible to fully align X-ray images and RGB-D images pixel by pixel. Therefore, if future elaboration is going to be

conducted, designing a deep learning-based model for image-to-image translate that do not require to align the input and output images will be a good direction.

Thirdly, we struggled to find reliable metrics to measure the performance of our model in stage 3. As we mainly cared about the clarity of the synthetic images and the correctness of the synthetic spine curves in the project, traditional metrics were not chosen by us. However, even the 2 metrics chosen by us, frankly speaking, could not fully reflect the performance of our model. We still found something experiments with 2 metrics that we could not explain. Hence, to find or even design more reliable metrics can be a good future work. In fact, we planned to use indirect methods to measure the quality of synthetic X-ray images after the final discussion with our supervisor. These indirect methods included doing landmark detection or apply the Cobb Angle estimator on the synthetic X-ray images to see if those models could yield good results on our synthetic X-ray images. However, the medical staff seemed not to have time with these indirect measurements. Although we sent the best batch of synthetic X-ray images to them, they did not give us the results at the end of this project. Therefore, if future work is permitted, these indirect measurements can also be a choice.

Finally, due to the medical condition, we received the 2 landmarks on X-ray images 15 days before the deadline of this projects. Therefore, not every parameters or hyper-parameters in the *pix2pix* mode had been fully tuned by us. In the future, more research can be done to find the best configuration of the model.

## **7. Conclusion**

In this project, we mainly implemented and trained 2 deep learning-base models to do landmark detection and X-ray synthesis in stage 2 and 3 respectively. We were relatively pleased with the model we trained in stage 2, which was relatively robust and accurate. We were not so pleased with the model we trained in stage 3 for the error in the synthetic shapes spine curves. More efforts can still be put into this project in the future to elaborate it.

However, the proof of concept is overall positive. The depth images could indeed provide essential information for the surface geometry of the back of the patients. Thus using RGB-D



images to finally synthesize the corresponding X-ray images has been proved to be a sensible concept so far.

In conclusion, using RGB-D images of the back of the patients to synthesize the X-ray images of the back of the patients is a possible solution to the side effect of X-ray machines. If a robust end-to-end application could be developed to synthesize accurate and clear X-ray images, radiation would not be a necessary by-product to diagnose scoliosis in children. If such an application could be finished, risk of radiation on children in the process of diagnosis of scoliosis could be avoided, which would be a great invention.

# Appendix

Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks

Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks

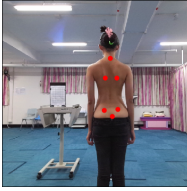
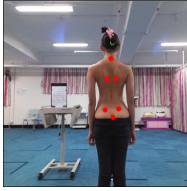
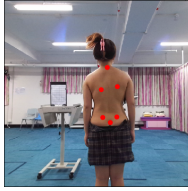
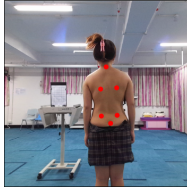
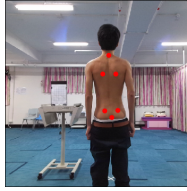
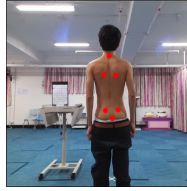

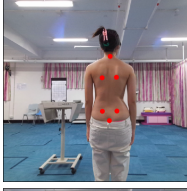
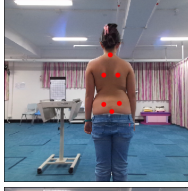
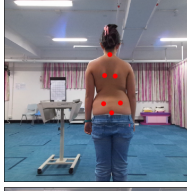
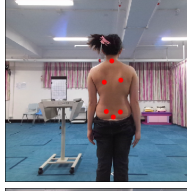
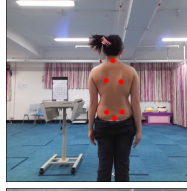
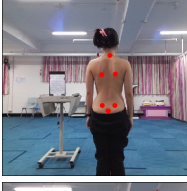

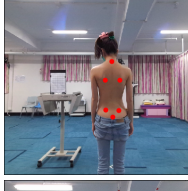
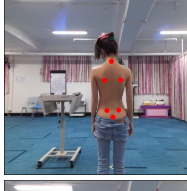
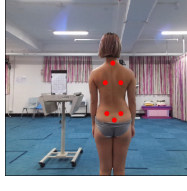
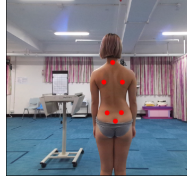
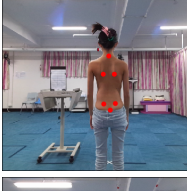
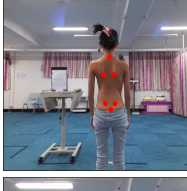
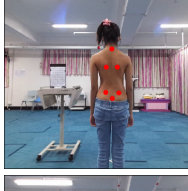
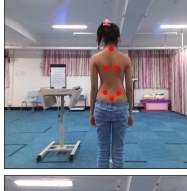


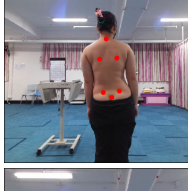
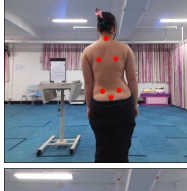


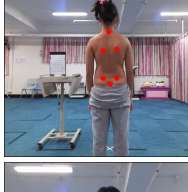
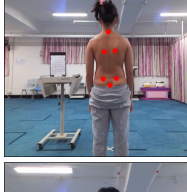
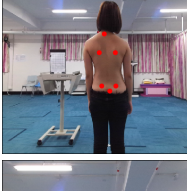
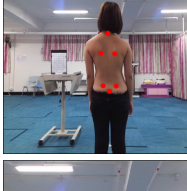
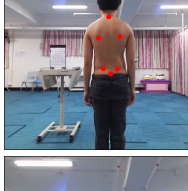
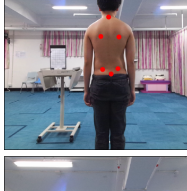


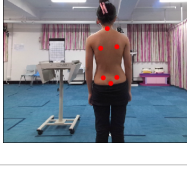
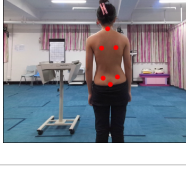
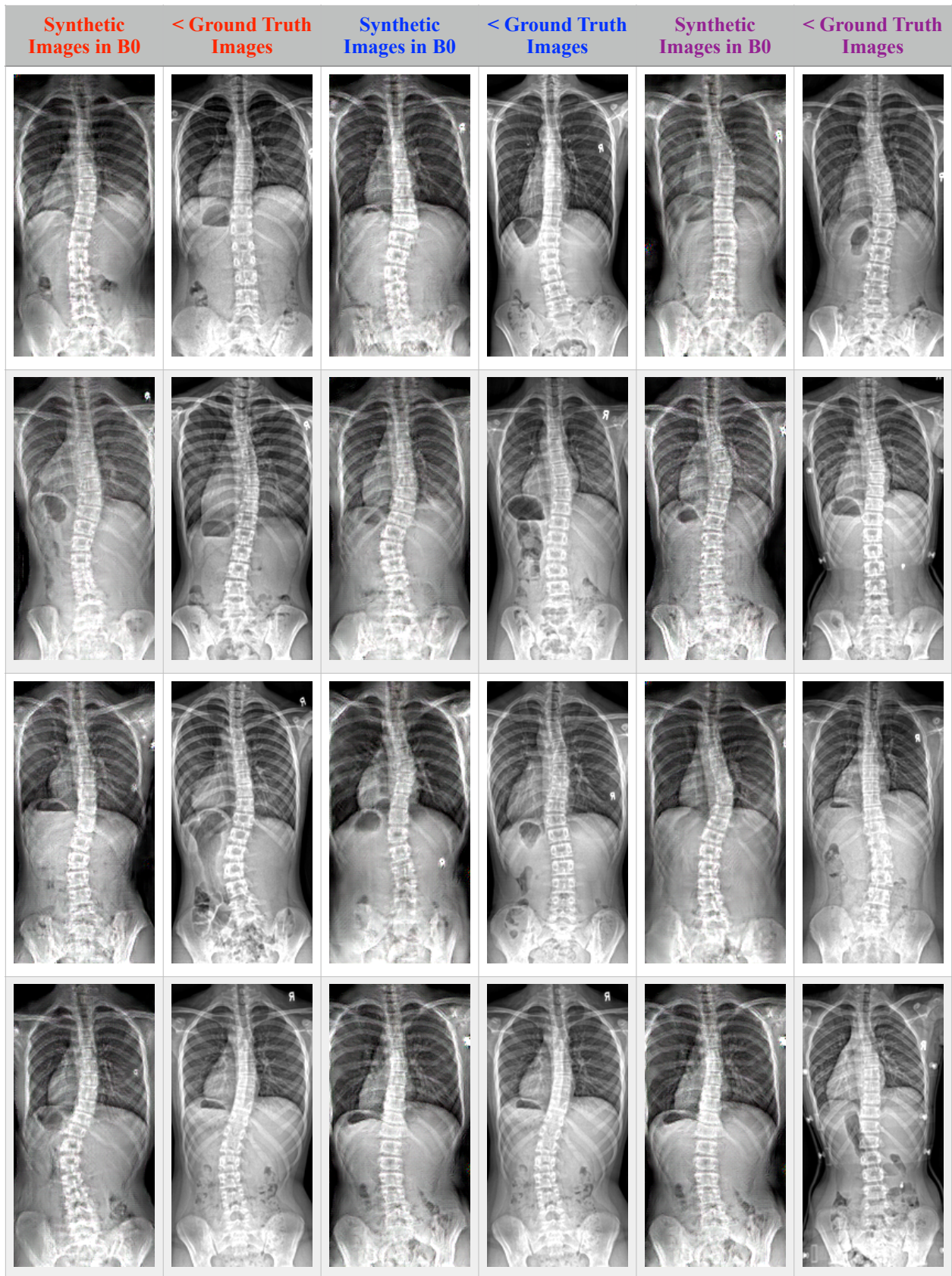
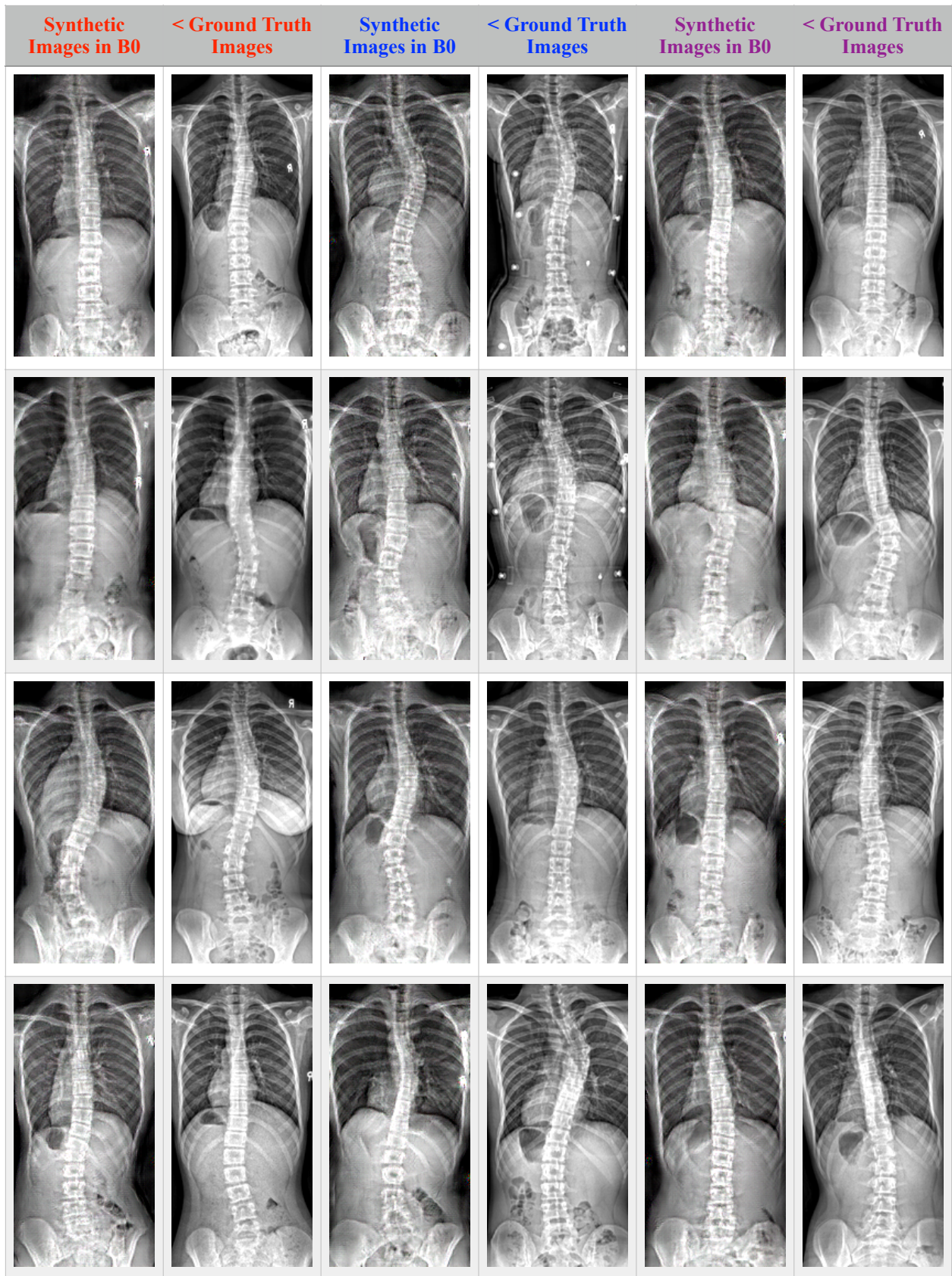
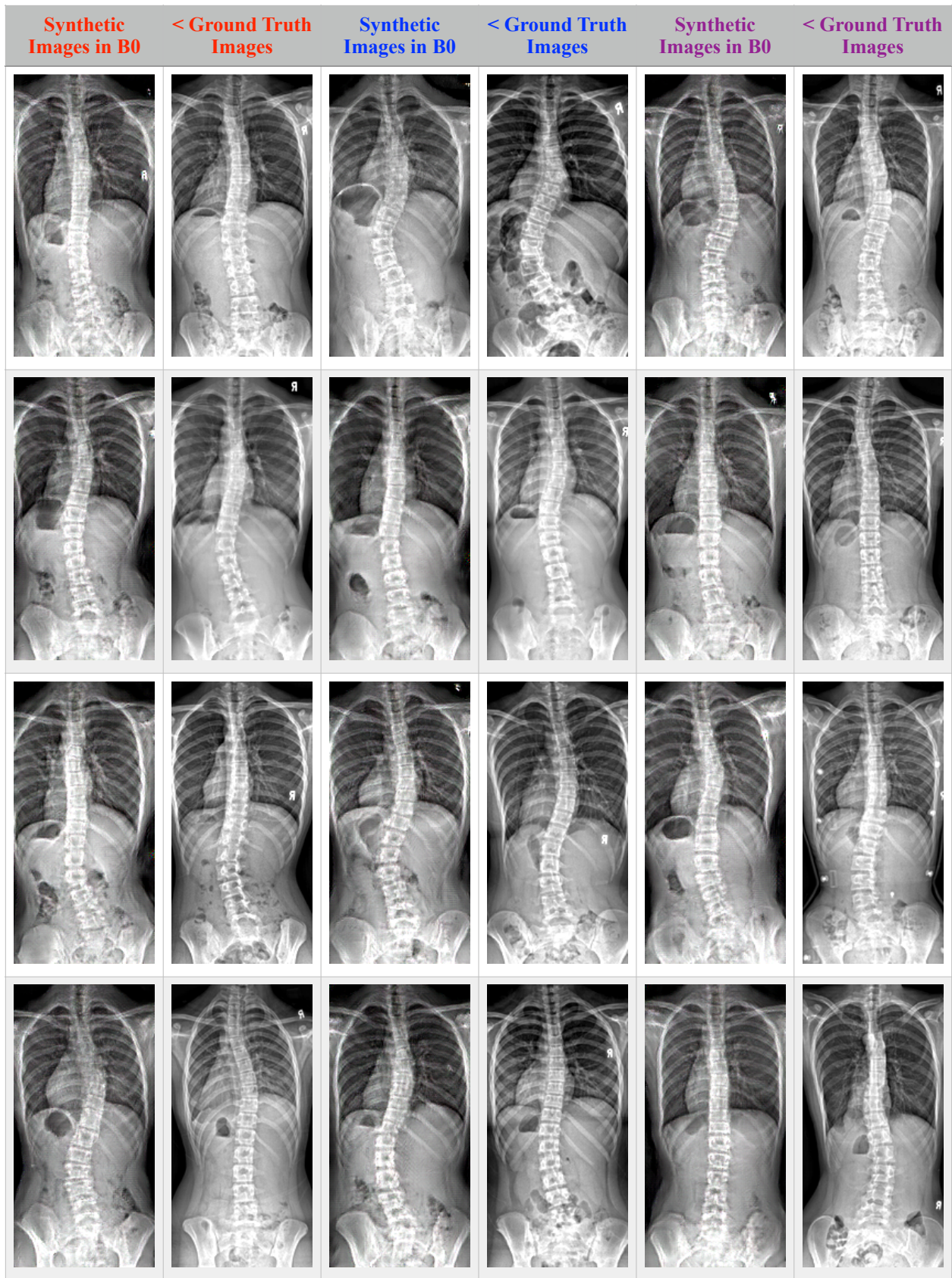
Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks	Predicted Landmarks	< Ground Truth Landmarks
					
					
					
					
					
					
					
					

Table 21: Final results of stage 2.









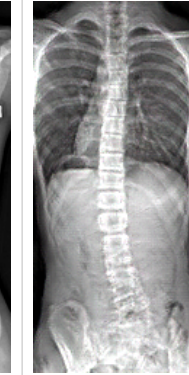
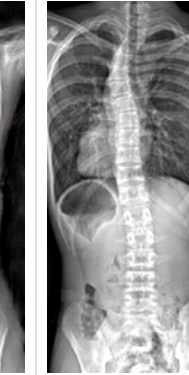
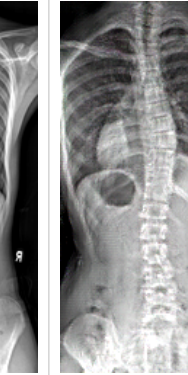
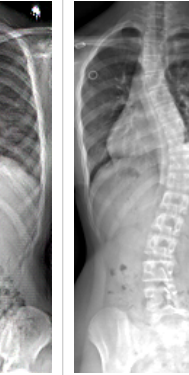


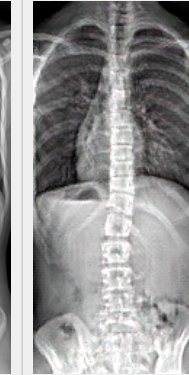
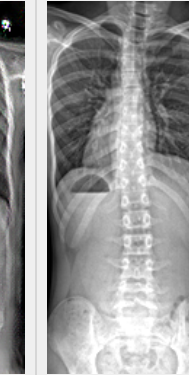
Synthetic Images in B0	< Ground Truth Images	Synthetic Images in B0	< Ground Truth Images	Synthetic Images in B0	< Ground Truth Images
					
					

Table 22: Final results of stage 3.



## References

- [1] H. Wu, C. Bailey, P. Rasoulinejad, and S. Li. Automatic Landmark Estimation for Adolescent Idiopathic Scoliosis Assessment Using BoostNet. In MICCAI, 2017.
  
- [2] H. Wu, C. Bailey, P. Rasoulinejad, and S. Li. Automatic Comprehensive Adolescent Idiopathic Scoliosis Assessment Using MVC-Net. In Medical Image Analysis, 2018.
  
- [3] R. Choi, K. Watanabe, H. Jinguji, N. Fujita, Y. Ogura, S. Demura, T. Kotani, K. Wada, M. Miyazaki, H. Shigematsu, and Y. Aoki. CNN-based Spine and Cobb Angle Estimator Using Moire Images. In IIEEJ, 2017.
  
- [4] M. Lootus, T. Kadir, and A. Zisserman. Vertebrae Detection and Labeling in Lumbar MR Images. In Computational Methods and Clinical Applications for Spine Imaging, 2014.
  
- [5] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. High-Resolution Representations for Labelling Pixels and Regions. In arXiv, 2019.
  
- [6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In arXiv: 1406.2661, 2014.
  
- [7] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In arXiv, 2018.
  
- [8] B. Teixeira, V. Singh, T. Chen, K. Ma, B. Tamersoy, Y. Wu, E. Balashova, and D. Comaniciu. Generating Synthetic X-ray Images of a Person from the Surface Geometry. In CVPR, 2018.
  
- [9] Microsoft Azure Kinect DK: Azure Kinect DK Documentation. <https://docs.microsoft.com/en-us/azure/kinect-dk>.