Automating Cobb Angle Measurement for Adolescent Idiopathic Scoliosis with Segmentation Based Methods

Chaojun Chen
chaojun.chen@mail.utoronto.ca

Luke Sun
lukej.sun@mail.utoronto.ca

Yujie Wu
yujie.wu@mail.utoronto.ca

Justin Zheng
justinzeyong.zheng@mail.utoronto.ca

Abstract

The Cobb angle is the standard for Adolescent Idiopathic Scoliosis (AIS) evaluation. Correct measurement of Cobb angles helps doctors determine the disease severity and optimal treatment plans for patients. The traditional method of measuring and assessing the Cobb angle involves manual procedures, which are labor-intensive and can lead to inconsistent results from observer to observer or from one assessment to another. Although there are automatic measurement methods using machine learning-based vertebrae segmentation, these methods have limited accuracy due to challenges posed by low-contrast X-ray imaging and ambiguous local morphology. As such, we explored and evaluated various existing segmentation architectures and techniques as inspiration to develop our own novel models with different data and loss function variations. Our best model achieved a Symmetric Mean Absolute Percentage Error (SMAPE) of 10.30%, outperforming previous researches.

1 Introduction

Adolescent Idiopathic Scoliosis (AIS) refers to the abnormal structural curvature of the spine that often appears during adolescence. AIS can cause chronic back pain and in severe cases, respiratory failure. Hence, early detection and diagnosis are necessary to prevent the effects of AIS from exacerbating. The measurement of the Cobb angle using anterior-posterior (AP) radiography (X-ray) is the standard clinical technique for scoliosis detection. However, due to the ambiguity and variability of X-ray images, manually measuring the Cobb Angle can be time-consuming and challenging. Given these challenges, it is imperative to develop automatic methods to precisely and efficiently measure the Cobb Angle in AP X-ray images. Such methods must (1) correctly locate the vertebrae, (2) identify crucial landmarks in the vertebrae and (3) precisely calculate the Cobb Angles. In this paper, we evaluated the performance of different segmentation models and propose modifications to existing methods. Using different models (UNet, Nested UNet, Attention UNet, SalsaNext) across different loss functions (Binary Cross Entropy, Mean Squared Error, Boundary Loss, Modified Boundary Loss) and different data variations (large data augmentation vs. small data augmentation), we compared and evaluated the overall performance of these models with respect to Cobb angle estimation. The top-performing models we have attained produced SMAPE scores of 10.30% and 10.77%, surpassing the results of previous studies.

2 Related works

The majority of the current computer-aided automatic Cobb Angle estimation methods are eventually the task of localizing the landmarks in the thoracic and lumbar region. All of them share a similar pipeline, which computes angles from landmark coordinates. Common approaches to localizing landmarks include regression and segmentation of the vertebra locations. The regression-based

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

methods estimate the landmark locations and Cobb angles directly. Yi et al. achieved a SMAPE score of 10.81% [1] using a ResNet-based novel model that detect the center of each vertebra then trace the four corners through the learned corner offset. Segmentation-based approaches to the vertebra landmarks approximation consist of classifying individual pixels into foreground and background, followed by landmarks extractions from the segmentation. Yen et al. proposed to use standard Deep Learning techniques such as Convolutional Neural Networks (CNN) for vertebra detection [2] and achieved SMAPE score of 23.98% using DenseNet 121.

Many of the previous researchs lack common benchmarks and use different evaluation methods and dataset, making them difficult to compare. Moreover, there exists no research that evaluates the impact of model complexity, loss functions variations, and data augmentation on the accuracy of cobb angle measurement. Our project will fill this research gap.

3 Segmentation Based Methods and Experiments

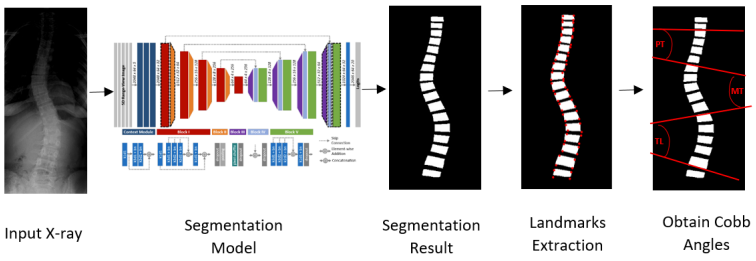

Figure 1: Pipeline: X-ray images are processed by a segmentation model. Landmarks are extracted from minimum bounding boxes around vertebrae segmentation result. 3 angles are then calculated in proximal thoracic (PT) section, main thoracic (MT) section, and thoracolumbar (TL) section.

We implemented 4 different models for this segmentation task: UNet, Nested UNet, Attention UNet, and SalsaNext. In this section, we will go over the two models that achieve the best performance: Attention UNet and SalsaNext. Detailed explanation of UNet and Nested UNet is in Appendix B.

Attention UNet

Attention UNet is a neural network that combines UNet architecture with an attention mechanism to improve segmentation accuracy [3]. The main difference between UNet and Attention UNet is in the decoder section, Attention UNet integrates attention gates into the skip connections to capture pertinent regions. The attention gate inside the network takes 2 inputs, X and G. X comes from the early layers through skip connections which has better spatial information. G is the gating signal that comes from the next lowest layer of the network which has better feature representation. The attention gate combines X and G using the following operations:

$$S = sigmoid(\psi(ReLU(\phi_x(X) + \phi_g(G))) \tag{1}$$

$$Y = XS \tag{2}$$

In equation (1), ϕ, ψ_x, ψ_g, are linear transformations implemented as 1×1 convolutions. Equation (2) is an element-wise multiplication between S and the original X vector. The multiplication scales information in X based on relevance, giving more weight to the feature of interest. The attention gate parameters and parameters in convolution layers are trained through backpropagation [3]. (See Figure 3 in Appendix B for model detail.)

SalsaNext

SalsaNext is a more sophisticated network compares to Attention UNet. The input image first get pass through a contextual module that aggregate the context information by employing a residual dilated convolution stack. This enables the model to capture more detailed spatial information while also taking into account the global context [4].

The next section of the network employs the conventional encoder-decoder architecture with a bottleneck. SalsaNext's encoder integrates dilated convolutions with receptive fields of 3, 5, and 7

in a novel manner. This technique increases the receptive field without significantly increasing the number of trainable parameters. The outputs of the dilated convolution layers are fused together and further processed to exploit additional information(Block I). Dropout and pooling layers are then applied to each of these new residual dilated convolution blocks (Block II). In Block III, a pixel-shuffle layer is introduced to reshape the feature map's elements to a form of $(Hr \times Wr \times C)$, where H, W, C, and r denote the height, width, channel number, and upscaling ratio, respectively. To enhance performance, the pixel-shuffle outputs are concatenated with the corresponding skip connection outputs (Block IV) before being fed into the decoder's dilated convolutional blocks (Block V). Following the decoder unit, a 1×1 convolution layer and a sigmoid function are used to compute the final segmentation [4]. (See Figure 4 in Appendix B for model detail.)

3.1 Loss Functions

In this section, we introduce the loss functions used. Loss functions help to optimize segmentation neural networks. We evaluated the performance of different loss functions such as MSE and BCE. We also implemented and modified a novel loss function *boundary loss* that address the difficulties caused by unbalanced segmentations.

Mean Squared Loss (MSE) and Binary Cross Entropy (BCE):

The MSE loss function and BCE loss function can be expressed as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \qquad \text{BCE} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{3}$$

where n is the number of samples, y_i is the true value of the i-th sample, and $\hat{y}_i$ is the estimation of the i-th sample. For BCE, y_i is either 0 or 1 and $\hat{y}_i$ is between 0 and 1. Reference Appendix C.

Modified Boundary Loss

We proposed a modification to the existing boundary loss function [5] (see Appendix C) to address its limitations. We took false negatives into account and computed a ratio between correctly predicted targets and incorrectly predicted targets. This approach enforces non-trivial outcomes and leads to more meaningful results. The new formula is given as follows:

$$\mathcal{L}_{\textit{modified}}(\theta) = \int_{\Omega} -\frac{\phi_{G_t}(q)s_\theta(q)}{\phi_{G_{bg}}(q)s_\theta(q) + [\phi_{G_t}(q) - \phi_{G_t}(q)s_\theta(q)] + \epsilon} dq \tag{4}$$

Ω represents the image domain, $s_\theta(q)$ represents the neural network's softmax probability output. $\phi_{G_t}(q)$ and $\phi_{G_{bg}}(q)$ encode distance information only for ground truth target and background pixels. Equation 4 incorporates False Negative regions and penalizes both False Negative and False Positive regions while rewarding True Positive regions. This sensitivity to weights and biases reduces the risk of being trapped in a local minimum.

3.2 Cobb Angle Calculation Pipeline

To calculate Cobb angles, a minimum bounding box is drawn around each vertebra segmentation. The 4 corners of the box are our landmarks. Using the extracted landmarks and simple trigonometry, we calculate the 3 angles in the proximal thoracic (PT), main thoracic (MT), and thoracolumbar (TL) section (as shown in figure 1). Appendix D contains detailed description of these steps.

3.3 Dataset

We use AASCE-MICCAI challenge 2019 dataset [6], which consists of 609 spinal Anterior-Posterior X-ray images, all showing signs of scoliosis. Each image is annotated with 68 landmark coordinates, corresponding to the four corners of 17 vertebrae, as labeled by radiologists. The dataset also provides ground truth Cobb angles assessed by radiologists. The dataset is split into 481 training images and 128 testing images. To evaluate the impact of data augmentation on model performance, we created two sets of augmented data using techniques such as mirroring, flipping, tilting, and gamma adjustment. The first set had 626 images, while the second set had 2886 images. We used 80% of the images for training and 20% for validation.

3.4 Implementation and Evaluation Metrics

We implement our method in PyTorch with GeForce RTX 4090, NVIDIA A10 and NVIDIA A100 GPUs. Input resolution of the images is fixed to 512×256 pixels. The networks are optimized using the AdamW with an initial learning rate 1×10^{-4}. Most networks were trained with 60 epochs and batch size of 16. The exceptions due to computation constraints are listed in Appendix E.

We choose Symmetric Mean Absolute Percentage Error (SMAPE) as our primary evaluation metric to reflect the accuracy of the predicted Cobb Angles.

$$SMAPE = \frac{1}{N} \sum_{N} \frac{SUM[|\text{ground truth angle} - \text{prediction angle}|]}{SUM[\text{ground truth angle} + \text{predictin angle}]} * 100\% \tag{5}$$

Where N stands for the number of samples in the test dataset. We will also calculate the mean absolute difference with N equals the total number of angles:

$$\text{Mean Absolute Difference} = \frac{1}{N} \sum_{i=1}^{N} |\text{ground truth angle}_i - \text{predicted angle}_i| \tag{6}$$

4 Result and Discussion

4.1 Qualitative Results

Comparing Figure 2 a) and b), it is evident that a larger training dataset results in better segmentation and less noise. This is because larger training set exposes the model to more diverse cases and make the model more robust. From Figure 2 a), we can also see that complex model such as SalsaNext has the best results and produces less corrupted segmentation when comparing with the other models.

We found that incorporating boundary or modified boundary loss is effective in solving the corrupted mask issue when using simpler models like UNet. This is because UNet model tends to produce overlapping masks when vertebra endplates are close to each other. Boundary loss can create clear boundaries to overcome this overlapping issue. However, a drawback of using boundary loss is that it may misclassify unconfident pixels as background, leading to smaller vertebrae segmentation and less accurate results. Additionally, in more complex models, the benefits of using boundary loss are diminished as segmentation becomes more precise.

Our findings indicate that in the most complex models (i.e. SalsaNext), using BCE and MSE are more effective. This may be because boundary loss requires more iterations before producing non-trivial results, and in certain computationally expensive settings only a limited number of gradient descent iterations can be performed. Despite the potential for solving overlapping issues, BCE and MSE produce more precise masks for the majority of important structures, resulting in overall better performance.

4.2 Quantitative Result

From Table 1, SalsaNext has the best SMAPE result of 10.30% and a mean absolute difference of $4.51°$. Attention UNet achieved an equally good SMAPE result of 10.77% and a mean absolute difference of $4.81°$.

SalsaNext achieves strong performance by utilizing dilated convolution blocks, which increase the receptive field of the model and allow it to capture more descriptive spatial features. We also noticed that Attention UNet has better performance than the traditional UNet. This is because Attention UNet actively suppresses activations at irrelevant regions through its attention gate, which directs the model's focus to salient regions. This results in a significant improvement in the model's ability to represent information without increasing computational cost or model parameters. Meanwhile, UNet does not have attention mechanism so its skip connections will bring along poor feature representation from the initial layers. Comparing the qualitative and quantitative results of Nested UNet and UNet, we see that Nested UNet has more consistent results. This is beacuase, Nested UNet can handle multi-scale features more effectively.

Looking at frames per second (FPS), we can conclude that Attention UNet is more efficient than SalsaNext. Attention UNet reduces the computational resources wasted on irrelevant activations and

provides better generalization of the network. Meanwhile, SalsaNext uses multiple techniques such as dilated convolution blocks and pixel shuffle to enhance network performance which makes the model larger and slows it down drastically.

5 Conclusion

This paper focuses on automating the Cobb angle measurement process with segmentation based approaches. We evaluated the effect of model complexity, loss function variations and dataset size on Cobb angle calculation accuracy. We conclude that Attention UNet is the overall best model with a low SMAPE score (10.77%), a low mean absolution difference (4.81°), and a high FPS (107), demonstrating the model's ability in both vertebra localization and Cobb angle measurement.

Previous works on automatic scoliosis detection have shown that the ResNet-based regression architecture achieved the best performance so far, with a SMAPE of 10.81% [1]. Our best method outperformed this approach with a leading SMAPE of 10.30%. For future works, we intend to replace the existing non-learning Cobb angle calculation technique with a regression network to minimize Cobb angle calculation error.

In conclusion, this paper represents a pioneering effort to investigate the impact of model complexity, variations in loss functions, and data augmentation on the accuracy of cobb angle measurement. The results of this study have successfully filled a gap in previous research, providing valuable insights into these important factors and their effects on measurement accuracy.

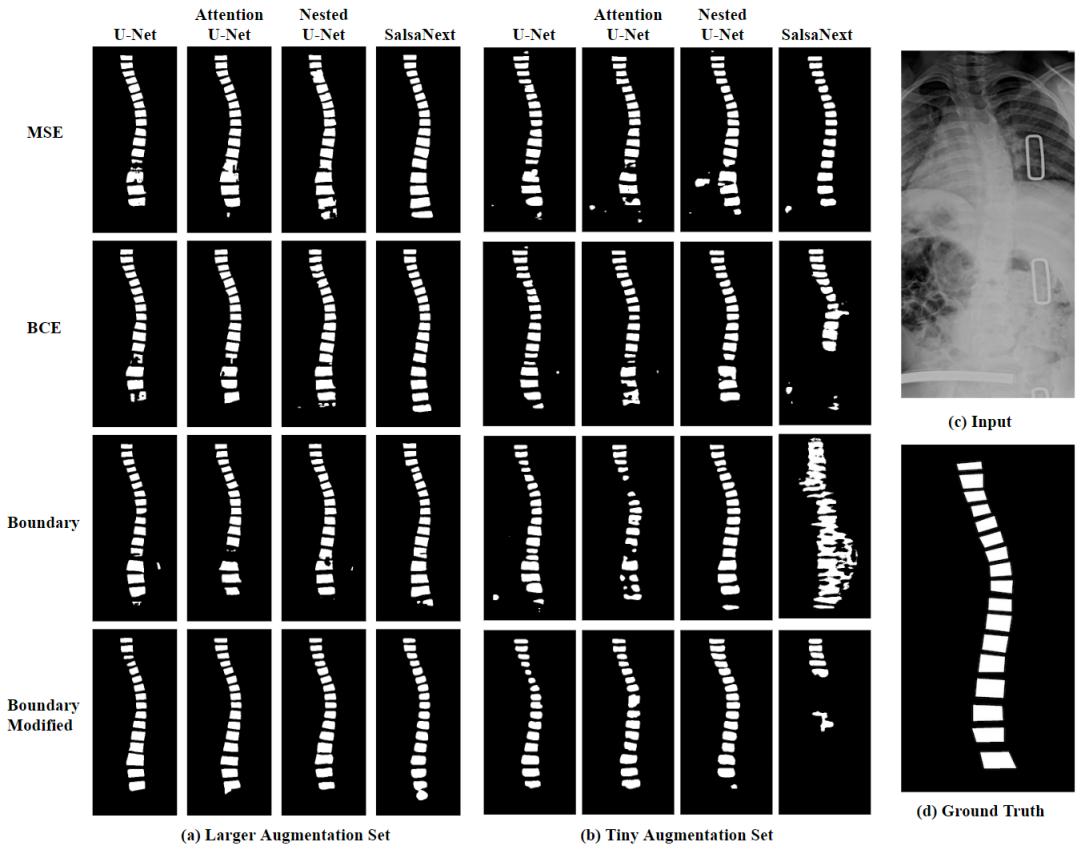

Figure 2: a) Test set results when training models using Large Data Augmentation Set (2886 training images); b) Test set results when training models using Tiny Data Augmentation Set (626 training images); c) Input X-ray; d) Ground truth segmentation provided by radiologists

Architecture	Augmentation Scale		Loss Function								FPS
			MSE		BCE		Boundary		$\text{Boundary}_{modified}$		
	130% (626)	400% (2886)	SMAPE(%)	$\lvert\Delta^\circ\rvert$	SMAPE(%)	$\lvert\Delta^\circ\rvert$	SMAPE(%)	$\lvert\Delta^\circ\rvert$	SMAPE(%)	$\lvert\Delta^\circ\rvert$	
UNet	✓		13.34	6.11	14.00	6.37	16.32	8.13	18.61	8.49	105
		✓	12.59	5.94	12.86	6.19	12.39	6.03	12.35	5.57	
Attention UNet	✓		14.81	8.68	14.79	7.36	18.46	8.95	15.05	6.64	97
		✓	11.87	5.51	**10.77**	**4.81**	13.48	6.48	12.97	5.96	
Nested UNet	✓		15.36	7.31	16.03	7.62	15.58	7.04	16.54	7.76	97
		✓	12.02	5.36	12.02	5.47	12.93	5.80	13.14	6.01	
SalsaNext	✓		16.54	7.77	19.69	9.39	39.15	25.63	23.16	11.18	8
		✓	**10.30**	**4.51**	12.32	5.78	18.83	14.45	15.25	7.09	

Table 1: Comparison of various segmentation architectures in spinal vertebrae segmentation tasks. **Augmentation Scale**: Size of augmented dataset compared with the original dataset with 481 training images, $\lvert\Delta^\circ\rvert$: Mean absolute difference. **FPS**: Frames per second measured using GeForce RTX 4090.

References

[1] Jingru Yi, Pengxiang Wu, Qiaoying Huang, Hui Qu, and Dimitris N Metaxas. Vertebra-focused landmark detection for scoliosis assessment. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 736–740. IEEE, 2020.

[2] Yen Hoang Nguyen. Scoliosis detection using deep neural network. *arXiv e-prints*, pages arXiv–2210, 2022.

[3] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, and et al. Attention u-net: Learning where to look for the pancreas. *arXiv.org*, May 2018.

[4] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. *Advances in Visual Computing*, page 207–222, 2020.

[5] Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger, Jose Dolz, and Ismail Ben Ayed. Boundary loss for highly unbalanced segmentation. In *International conference on medical imaging with deep learning*, pages 285–296. PMLR, 2019.

[6] Hongbo Wu, Chris Bailey, Parham Rasoulinejad, and Shuo Li. Automatic landmark estimation for adolescent idiopathic scoliosis assessment using boostnet. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20*, pages 127–135. Springer, 2017.

[7] Haoliang Sun, Xiantong Zhen, Chris Bailey, Parham Rasoulinejad, Yilong Yin, and Shuo Li. Direct estimation of spinal cobb angles by structured multi-output regression. In *Information Processing in Medical Imaging: 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, pages 529–540. Springer, 2017.

[8] Bidur Khanal, Lavsen Dahal, Prashant Adhikari, and Bishesh Khanal. Automatic cobb angle detection using vertebra detector and vertebra corners regression. In *Computational Methods and Clinical Applications for Spine Imaging: 6th International Workshop and Challenge, CSI 2019, Shenzhen, China, October 17, 2019, Proceedings 6*, pages 81–87. Springer, 2020.

[9] Z Zhou, Rahman Siddiquee M. M, N Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, page 3–11, 2018.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[11] Jingru Yi, Pengxiang Wu, Menglin Jiang, Qiaoying Huang, Daniel J Hoeppner, and Dimitris N Metaxas. Attentive neural cell instance segmentation. *Medical image analysis*, 55:228–240, 2019.

[12] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[13] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[14] Debasis Chaudhuri and Ashok Samal. A simple method for fitting of bounding rectangle to closed regions. *Pattern recognition*, 40(7):1981–1989, 2007.

[15] Hana Kim, Hak Sun Kim, Eun Su Moon, Choon-Sik Yoon, Tae-Sub Chung, Ho-Taek Song, Jin-Suck Suh, Young Han Lee, and Sungjun Kim. Scoliosis imaging: what radiologists should know. *Radiographics*, 30(7):1823–1842, 2010.

6 Appendix A

6.1 Previous Works on Direct Estimation Approach

Direct estimation approaches to the vertebra landmarks approximation are adapted from various machine learning algorithms, such as Support Vector Regression and Convolutional Neural Networks. The Structured Multi-output Regression [7] takes the advantage of Support Vector Regression in handling nonlinear input-output relationships and further employs a structure matrix in front of the original Support Vector Regression to encode the correlations across outputs (landmarks and angles). The BoostNet [6] extracts image features using convolutional neural network and filters out outlier features with statistical outlier removal methods in the specially designed Boostlayer. Other Convolutional Neural Network related approaches include isolating each vertebra with Faster-RCNN followed by detecting landmarks within each individual vertebra with DenseNet [8].

7 Appendix B

7.1 UNet

UNet is a U-shaped encoder-decoder deep learning network. The encoder part of the UNet architecture is composed of a series of convolutional and pooling layers that progressively reduce the spatial dimensions of the input image, while the decoder part is composed of a series of upsampling and convolutional layers that increase the spatial dimensions of the input image back to its original size.

UNet also uses skip connections between the encoder and decoder to pass low-level features to the decoder, which helps to create a finer segmentation output. Due to its high accuracy and efficiency for a model with relatively small number of parameters, UNet is a popular architecture for semantic segmentation tasks and image analysis.

7.2 Nested UNet [9]

Nested UNet is an extension of the UNet architecture that was proposed to further improve the performance of semantic segmentation tasks. Nested UNet has 2 main modifications when compared to the UNet:

1. It integrates convolution layers on the skip connections that connect the encoder and decoder feature maps, addressing the semantic gap between them.
2. It uses deep supervision at hidden layers of the network which allows for model pruning. This method improves or maintains similar performance compared to using a single loss layer.

7.3 Attention UNet

See Figure 3.

7.4 SalsaNext

See Figure 4.

7.5 ResNet Based Regression Approach [1]

The architecture proposed by Yi et al. [1] utilizes ResNet34 (or ResNet50 or ResNet101, interchangeably) to extract high-level semantic features, followed by skip connections that combine deep and shallow features, similar to [10] [11]. The landmark localization is achieved by constructing the keypoint heatmap, center offset, and corner offset maps using convolutional layers at layer D2.

The keypoint heatmap relies on an unnormalized 2D Gaussian disk to represent the pixel-wise ground truth. The Gaussian disk is expressed as $exp(-\frac{x^2+y^2}{2\sigma^2})$ and the radius θ is determined by the size of the vertebrae [12]. In order to save computational resources, positions (x_s, y_s) on the input image are mapped to the location $(\lfloor \frac{x_s}{n} \rfloor, \lfloor \frac{y_s}{n} \rfloor)$ on the feature map, where n is the downsampling factor. The center points are then extracted from the downsized feature map and mapped back to the original

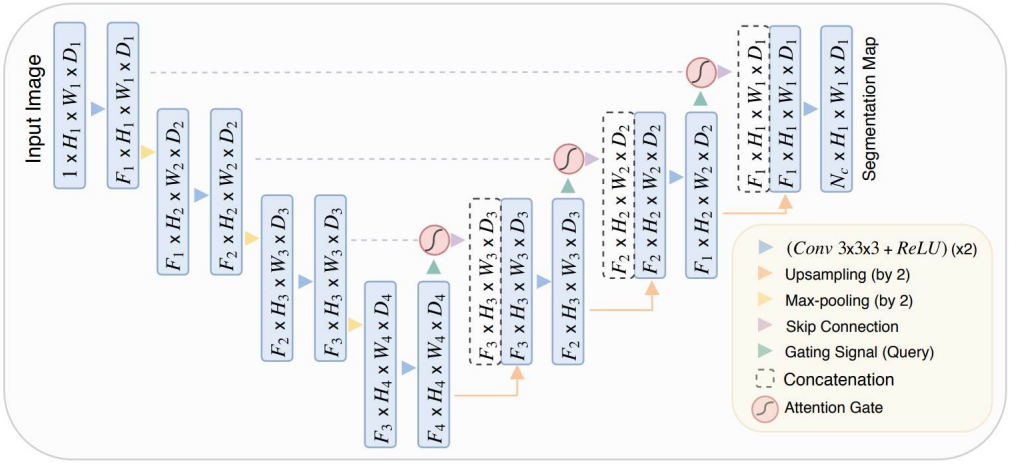

Figure 3: Attention UNet archetecture.

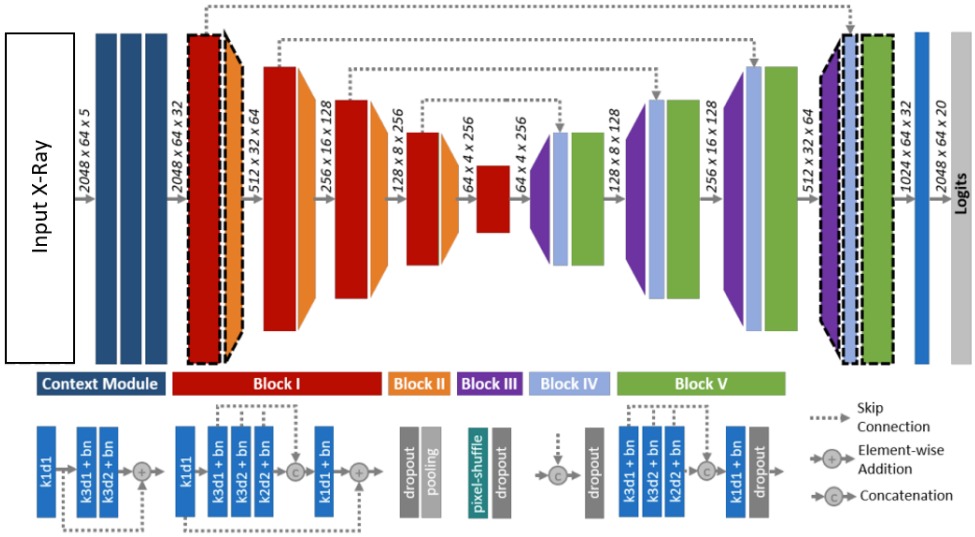

Figure 4: SalsaNext archetecture.

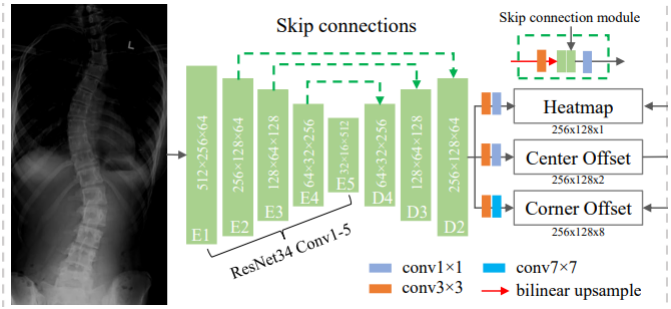

Figure 5: ResNet based regression acrhitecture for landmark detection [1]

input image using the center offset defined as $(\frac{x_s}{n} - \lfloor \frac{x_s}{n} \rfloor, \frac{y_s}{n} - \lfloor \frac{y_s}{n} \rfloor)$. Once the center points of each vertebra are localized, the 4 corner landmarks are traced using corner offsets, which are defined as vectors starting from the center and pointing to the vertebra corners, as shown in 5 [1].

8 Appendix C

Mean Squared Error

The mean squared error (MSE) is a statistical measure used to evaluate the accuracy of an estimator for estimating an unknown value. It calculates the average of the squared differences between the estimated values and the actual values. The MSE serves as a risk function and represents the expected value of the squared error loss. The MSE can be expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2 \tag{7}$$

where n is the number of samples, Y_i is the true value of the i-th sample, and $\hat{Y_i}$ is the estimated value of the i-th sample.

Binary Cross Entropy

Binary cross entropy (BCE) is a loss function commonly used in machine learning for binary classification problems. The goal of binary classification is to assign an input sample to one of two possible classes, typically represented as 0 or 1.

The binary cross entropy is a measure of the difference between the predicted probability distribution and the true probability distribution. In binary classification, the true probability distribution is a probability distribution that assigns a probability of 1 to the true class and a probability of 0 to the other class. The function assigns a score to each prediction reflecting how much it deviates from the expected value. In other words, it measures how close or far the predictions are from the truth. The Binary Cross Entropy can be expressed as :

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} y_i(log(\hat{y_i})) + (1 - y_i)log(1 - \hat{y_i}) \tag{8}$$

where N is the number of samples, y_i is the true label of the i-th sample (either 0 or 1), and $\hat{y}_i$ is the predicted probability of the i-th sample belonging to the positive class (between 0 and 1).

Boundary Loss [5]

The Boundary Loss as shown in Figure 6, is defined as the integral of the distances between pixels and their closest border, rather than the overlapping area between the prediction and the label, and can be expressed as follows:

$$\mathcal{L}_B(\theta) = \int_{\Omega} \phi_G(q) s_\theta(q) dq \tag{9}$$

In this context, Ω denotes the domain of the image. The function $\phi_G(q)$ returns the encoded distance information, where $\phi_G(q) = -D_G(q)$ if the pixel q lies within the target region of the ground truth, and $\phi_G(q) = D_G(q)$ if it is in the background ($D_G(q)$ is the distance map, which can be precomputed from the ground truth). On the other hand, $s_\theta(q)$ represents the softmax probability output of the neural network for the θ-th iteration, which falls within the range of [0, 1] [5].

From 9 we observe three potential cases depending on the location of the pixel:

- True Positive: where a pixel is in the target region of both the GT and current prediction. In this situation, $\phi_G(q) = -D_G(q)$ and $s_\theta(q) \approx 1$ and the pixel contributes negatively to the loss.
- False Positive: where a pixel is in the background region of the GT but classified as the target. In this situation, $\phi_G(q) = D_G(q)$ and $s_\theta(q) \approx 1$ and the pixel contributes positively to the loss.
- False Negative: where a pixel is in the target region of the GT but classified as the background. In this situation, $\phi_G(q) = -D_G(q)$ but $s_\theta(q) \approx 0$ and the pixel has no contribution to the loss.

The boundary loss function is designed to address the difficulties caused by unbalanced segmentations. Instead of using unbalanced integrals within regions, the loss function uses integrals over the boundary

between regions. Whereas, neglecting false negatives can result in a major issue where the loss function is susceptible to being trapped in a local minimum. This problem arises when a probability map is generated, classifying all pixels as background, resulting in a trivial solution with very low gradients. To prevent such a situation, the original boundary loss needs to be guided by a traditional loss function in early training steps [5].

8.1 Modified Boundary Loss

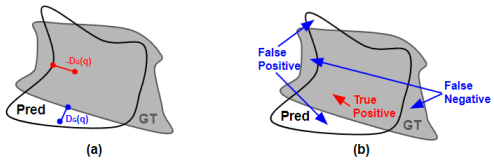

Figure 6: (a) the evaluation of the Boundary Loss [5] at different pixel locations. (b) three essential scenarios are taken into consideration by the Boundary Loss (and its modification).

9 Appendix D

9.1 Segmentation Masks to Landmark Coordinates

To initiate the vertebra extraction process, we employed the Border-following algorithm [13] on the segmentation results. This algorithm detects the borders of connected foreground objects, namely, the vertebrae, and generates coordinate collections. Within each vertebra contour, we implemented the minimum bounding rectangle (MBR) algorithm [14] to identify the tightest enclosing rectangle. The landmark coordinates were then obtained as the four corners of the minimum bounding rectangle.

9.2 Landmark Coordinates to Cobb Angles

To calculate the final Cobb angle, we utilized the extracted landmark coordinates. The following steps were undertaken:

1. calculating the slopes between each pair of corners,
2. determining the apex [15], which is the pair with the greatest deviation from the expected spine position,
3. identifying the most tilted landmark pair above the apex (i.e. α),
4. identifying the most tilted landmark pair below the apex (i.e. β),
5. using equation 1. to calculate three Cobb angles: proximal thoracic (PT), main thoracic (MT), and thoraco-lumbar (TL).

Finally, the equation for calculating the angles is

$$Arctan\theta = \frac{slope_1 - slope_2}{1 + slope_1 \cdot slope_2} \tag{10}$$

10 Appendix E

Due to computational limitations, all NestedUNet models were trained for 30 epochs with a batch size of 8. SalsaNext models were similarly trained for 30 epochs, using batch sizes of 8-16. Among these, the SalsaNext model with MSE loss outperformed the others, achieving a SMAPE of 10.38%. To further improve its performance, the model was trained for an additional 30 epochs, totaling 60 epochs, with a batch size of 16, slightly improving its SMAPE to 10.30%.