



Scoliosis Analysis Using Attention U-Net Segmentation and Curvature Estimation

**Capstone Project Phase B
Group number: 25-1-R-10**

Students: Dvir Hayat, Rafael Abbassov

Advisor: Prof. Miri Weiss Cohen



Table of Contents

| | | |
|-----|----------------------------------|----|
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 3 | Methodology | 6 |
| 3.1 | Implementation | 6 |
| 4 | Research Proposed Approach | 10 |
| 4.1 | YOLOv8 | 13 |
| 4.2 | Attention U-Net | 14 |
| 4.3 | Linear Function Pipeline | 16 |
| 5 | Discussion | 18 |
| 5.1 | Future Work | 18 |
| 6 | Conclusion | 20 |
| 7 | User Documentation | 20 |
| 7.1 | GUI | 20 |
| 7.2 | User Manuel | 21 |
| 7.3 | Maintainer Manual | 22 |

1 Abstract

Scoliosis is a medical condition characterized by an abnormal curvature of the spine, commonly measured by the Cobb Angle. Accurate measurement of this angle and a comprehensive understanding of the spine's curvature are essential for precise treatment, necessitating high-precision segmentation and detailed description.

The project proposes a three-stage approach using spinal imaging scans: first, the vertebrae are detected using YOLOv8, then they are segmented using Attention U-Net, and finally, they are represented using a corresponding B-Spline. In spinal imaging, YOLOv8 is a detection that is critical to identifying vertebrae accurately. Also, YOLOv8 is capable of handling complex scenes, ensuring reliable detection in even the most challenging of medical imaging environments. The Attention U-Net enhances segmentation by incorporating attention mechanisms that focus on the most relevant features of spinal images. This allows the model to effectively capture fine details and differentiate between adjacent vertebrae. Using B-Spline representations of segmented vertebrae, it is possible to model the spinal shape with great precision, even capturing subtle deviations. As a result of this detailed representation, effective treatment strategies can be devised and the progression of scoliosis can be monitored.

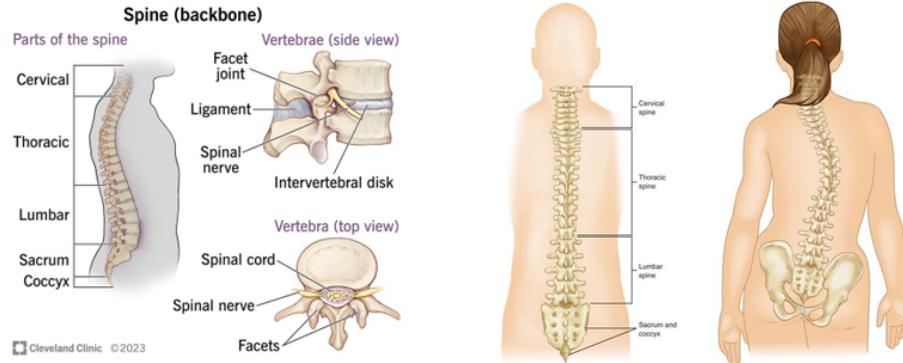
Keywords: Scoliosis, Spine Segmentation, B-Splines, Attention U-Net, YOLOv8

2 Introduction

The back bones form the vertebral column, which is divided into five sections: the cervical, thoracic, lumbar vertebrae, the sacrum, and the coccyx. Together, these 33 vertebrae support movement and posture while also providing protection and structural support [1].

We will elaborate about the three largest sections of the spine. The cervical, thoracic, and lumbar spine sections work together to protect the spinal cord, support the body, and enable movement [7]. The cervical spine, located in the neck region, supports and cushions the head and neck. It allows for a wide range of motion, including rotation, while simultaneously safeguarding the spinal cord from injury [7]. The thoracic spine, situated in the mid-back, is primarily responsible for bearing significant loads from the upper body. It contributes to maintaining posture and stability in the trunk. Additionally, it is connected to the rib cage, providing protection for the vital organs within the chest, such as the heart and lungs, and plays a major role in maintaining the body's overall safety and function.

The lumbar spine, located in the lower back, is designed to provide maximum stability, supporting the heavy loads carried by the upper body while allowing mobility of the trunk relative to the hips/pelvis. It also plays a major part in bending and twisting movements.



(a) Spine structure side view with vertebrae top and side view.[3]

(b) Normal spine anatomy and Idiopathic scoliosis in children.[9]

Scoliosis is a spine deformity medical condition which is diagnosed through measuring spinal deviation on standing coronal plane radiographs [14]. Diagnosis usually measured through measurement of the Cobb angle. The Cobb angle is measured when 2 vertebrae are selected as the ones whose endplates are most tilted towards each other. Then, lines are then drawn along the endplates, and the angle between the two lines, where they intersect, is measured. Scoliosis is

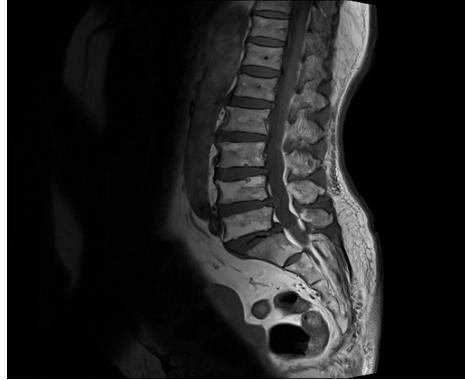


Fig. 2: Example CT scan of the spine. Such images are used for detecting and analyzing spinal

Cobb angle

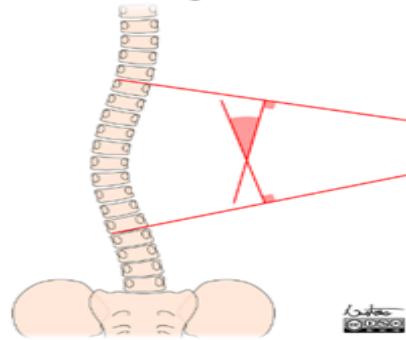


Fig. 3: Cobb angle measurement method for scoliosis[10] defined as a lateral spinal curvature with a Cobb angle of $> 10^\circ$ [6][10]. There are three types of Scoliosis: The most common is “Idiopathic scoliosis”, where the main reason is uncertain; however, researchers believe that the condition probably stems from a variety of genetic and environmental factors, including abnormal muscle growth, hormonal issues, genetic influence and sometimes as a symptom of a problematic nervous system [16][4]. The second one is “Congenital scoliosis” which is rarer and often associated with pregnancy during the 5th and the 8th week, when the spine of the embryo is being developed and the bones don’t form as they should. This type of scoliosis is mostly detected at the time of parental ultrasound [4][15].

The third one is “Neuromuscular scoliosis,” and it occurs due to problems in the muscles and nerves supporting the spine, which can be caused by trauma or muscle disease. It commonly affects posture, mobility, and can lead to respiratory or cardiovascular complications as the curve progresses [4]. Understanding the patient’s structure and status of the vertebral column is crucial for diagnosing and treating spine-related disorders, particularly scoliosis, which remains a significant global health concern. Our goal is to improve the current segmentation existing abilities through deep learning methods and achieve higher accuracy that will

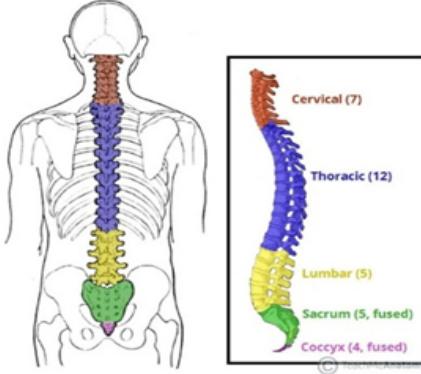


Fig. 4: The spine divided into cervical, thoracic, lumbar, sacrum, and coccyx regions.[15] enable medical personnel to treat patients with more specific treatment that will hopefully eventually lead to better recovery.

3 Methodology

As decided during the research process in the first stage of this project, we based our project on a three-stage deep learning pipeline for scoliosis analysis using CT scans. The first stage employs YOLOv8 to accurately detect and localize each vertebra within the scan, providing bounding boxes for further analysis. In the second stage, each detected vertebra is segmented using an Attention U-Net, which leverages attention mechanisms to enhance segmentation accuracy.

For modeling the curvature of the spine, our initial plan included the use of B-spline curves to represent the spatial trajectory of the vertebral centroids. However, during development, we found that a simpler linear interpolation of vertebral centers was sufficient for reliable curvature estimation. This approach provided the results with less computational overhead, while still capturing the clinically relevant features needed for scoliosis evaluation.

3.1 Implementation

In this chapter, we provide a detailed overview of our model’s components, describing each part of the system.

YOLOv8

YOLO, by Joseph Redmon et al., was published at CVPR 2016 [11]. It presented a real-time end-to-end approach to object detection analysis. Ultralytics, the developers of YOLOv8 as well as YOLOv5, have created a robust computer vision model [5,12]. Unlike earlier versions, YOLOv8 uses an improved structure, providing more flexible bounding boxes, and maintains a high level of accuracy with

speed, making it versatile for many use cases [5]. Figure 5 presents a detailed description of the YOLOv8 architecture. YOLOv8 uses a special backbone called the C2f module, which is an improvement over the older CSPLayer. The C2f module (cross-stage partial bottleneck with two convolutions) combines high-level features with contextual information to improve detection accuracy [11]. YOLOv8 uses an anchor-free design, predicting the center of an object directly and removing the need for predefined anchor boxes, which simplifies the model. A decoupled head is used for classification and regression tasks independently, allowing each branch to specialize and improve overall accuracy. In the output layer, a sigmoid function is used as the activation function for the object detection score, representing the probability that a bounding box contains an object. The SoftMax function calculates the object's probabilities belonging to each possible class [11]. The YOLOv8 architecture features a modified CSP-

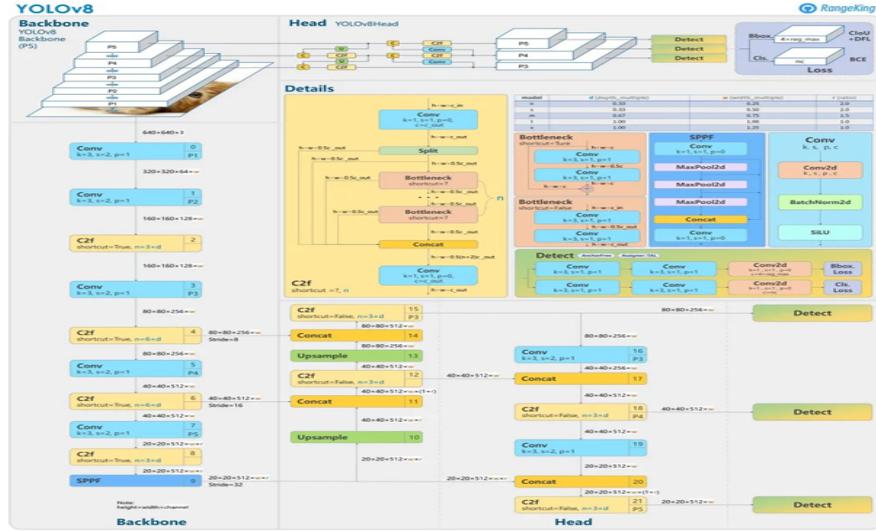


Fig. 5: YOLOv8 Architecture [2]

Darknet53 backbone (as in YOLOv5), a C2f module, and an SPPF layer for faster computation by pooling features into a fixed-size map. Convolution layers use batch normalization and SiLU activation. Its decoupled head processes objectness, classification, and regression tasks separately for improved accuracy. YOLOv8, which builds on the foundations of YOLOv5, presents several advancements compared to earlier versions. Evaluated on the MS COCO dataset test-dev 2017, YOLOv8x achieved an AP of 53.9% with an image size of 640 pixels (compared to 50.7% for YOLOv5 on the same input size), with a speed of 280 FPS on an NVIDIA A100 and TensorRT.

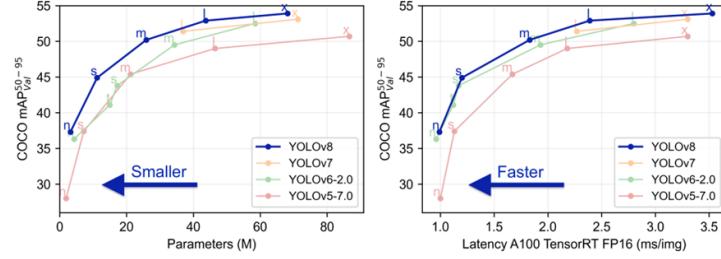


Fig. 6: Latency and parameters convergence comparison through different YOLO versions [5]

Attention U-Net

Attention U-Net, by Oktay et al., is a CNN-based model with modifications to the well-known U-Net [8], created to enhance segmentation (originally for the pancreas) by focusing on relevant areas within a CT scan. The network uses the encoder-decoder U-Net architecture as a base, with additional attention gates (AG) between decoder layers. The encoder uses down-sampling with max-pooling for feature extraction, and the decoder uses up-sampling layers. As the decoder begins (first up-sampling layer), attention gates are positioned along the skip connections.

The base U-Net structure consists of convolutional layers, max-pooling, up-convolutional layers, and skip connections. The network applies two 3×3 un-padded convolutions followed by ReLU. In the encoder, convolutions are followed by 2×2 max-pooling for down-sampling. In the decoder, after convolutional layers, the network applies up-sampling. To minimize information loss, up-sampled feature maps are combined with corresponding encoder feature maps at the same level, preserving contextual information. Attention gates were initially explored

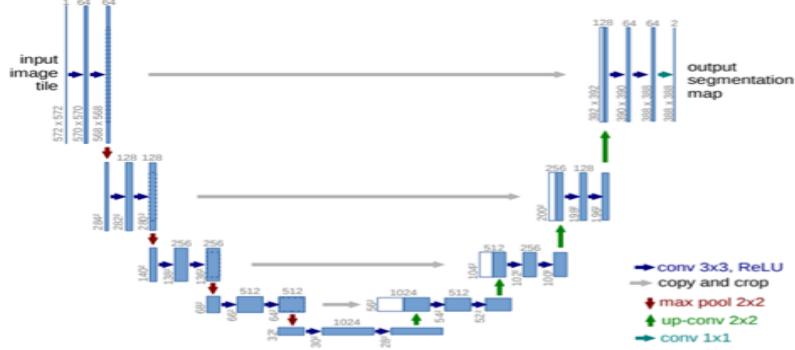


Fig. 7: Original U-Net Architecture [13]

by interpreting gradients of output class scores with respect to the input image, and were inspired by research in natural image analysis, knowledge graphs, and language processing. There are two main types of attention mechanisms: hard and soft. Hard attention is non-differentiable and may require reinforcement learning for parameter updates, while soft attention is differentiable and trained via backpropagation. The attention modules in this model use soft attention, improving accuracy by reducing false positives through the suppression of irrelevant background features, without scaling down the region of interest. Attention coefficients $\alpha_i \in [0, 1]$ identify salient regions and filter out irrelevant feature responses, keeping only the relevant areas for the specific task. The output of the attention gate is the element-wise multiplication of the input feature maps and the attention coefficients: $\hat{x}_{i,c}^l = x_{i,c}^l \cdot \alpha_i^l$. For each pixel $x_i^l \in \mathbb{R}^{F_l}$, a scalar or vector attention value is calculated, allowing each AG to focus on a different section of the feature map. The gating vector $g_i^l \in \mathbb{R}^{F_g}$ is used for each pixel to determine focus regions and prune less relevant features. Additive attention is used, formulated as follows:

$$q_{\text{att}}^l = \psi^T (\sigma_1 (W_x^T x_i^l + W_g^T g_i + b_g)) + b_\psi, \quad (1)$$

$$\alpha_i^l = \sigma_2 (q_{\text{att}}^l (x_i^l, g_i; \Theta_{\text{att}})), \quad (2)$$

where σ_2 is the sigmoid activation. AG is characterized by parameters Θ_{att} with linear transformations $W_x \in \mathbb{R}^{F_l \times F_{int}}$, $W_g \in \mathbb{R}^{F_l \times F_{int}}$, $\psi \in \mathbb{R}^{F_l \times 1}$, and bias terms $b_\psi \in \mathbb{R}$, $b_g \in \mathbb{R}^{F_{int}}$. These transformations are computed using channel-wise $1 \times 1 \times 1$ convolutions. In Attention U-Net, the attention gates are incorpo-

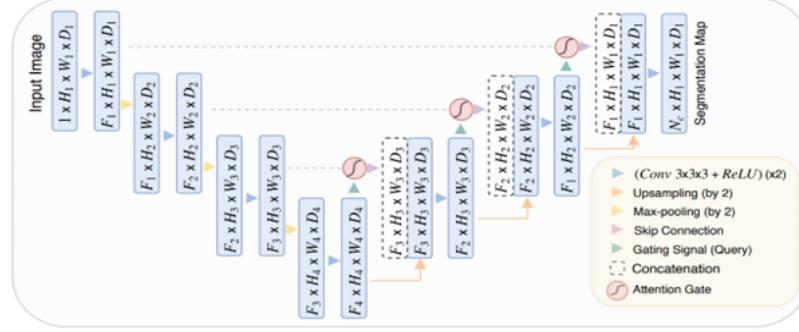


Fig. 8: Attention U-Net model architecture

rated through skip connections and the up-sampling stage at each decoder level. Each attention block receives two inputs: the gating signal (from the previous decoder stage) and spatial information (from the encoder via skip connection). Both inputs are convolved, with one using stride (1,1) and the other (2,2) to match dimensions. The resulting tensors are summed (aligned weights increase, unaligned decrease) and passed through a ReLU activation. A further operation

generates a matrix of size $L \times W$. The resulting weights matrix is normalized and multiplied with X , followed by a sigmoid function and up-sampling. The output vector is relevance-weighted and handled as in a standard U-Net.

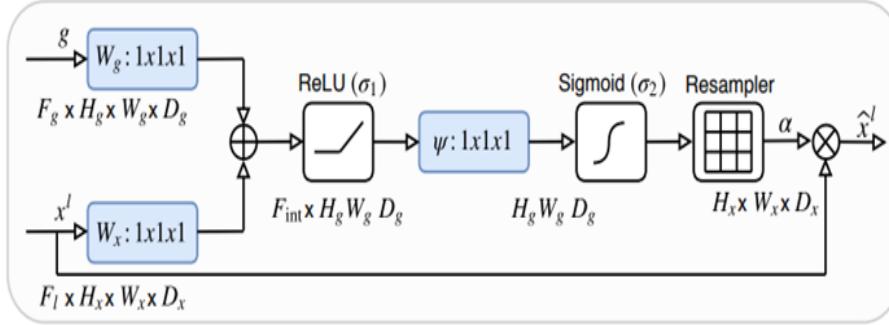


Fig. 9: Attention gate scheme within the Attention U-Net

4 Research Proposed Approach

The proposed framework leverages a two-stage deep learning pipeline that combines the capabilities of YOLOv8 for object detection and Attention U-Net for semantic segmentation to achieve accurate vertebrae segmentation from CT spine scans. In the first stage, a CT scan is provided as input to the YOLOv8 model. YOLOv8 detects and localizes each vertebral body by generating bounding boxes around them within the scan. These detected regions of interest are then extracted and serve as input for the next stage.

In the second stage, each cropped vertebral region is processed by an Attention U-Net. Using attention mechanisms after getting focus from the bounding box enables precise delineation of vertebral boundaries, significantly improving segmentation quality over single-stage approaches.

Following segmentation, the coordinates of each vertebra's center—denoted as $(x_0, y_0), \dots, (x_n, y_n)$ —are extracted for further analysis, such as measuring spinal curvature for scoliosis assessment.

To maintain consistency and optimize model performance, standard preprocessing techniques such as resizing and normalization are applied to all input data. The seamless integration of detection and segmentation within this architecture reduces computational complexity, increases segmentation accuracy, and efficiency, fully automated workflow for robust vertebrae analysis.

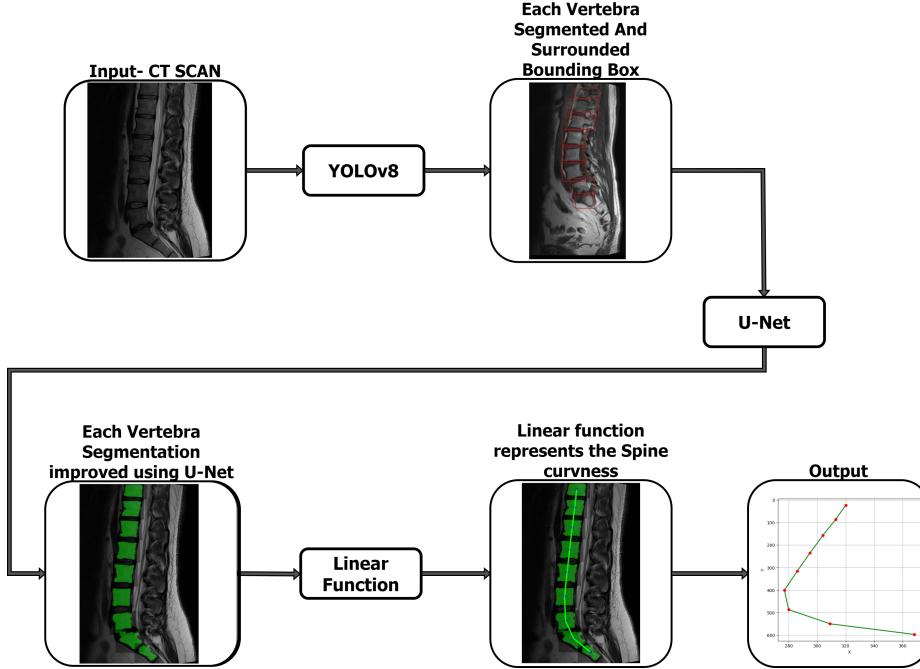


Fig. 10: BAYU Net Architecture

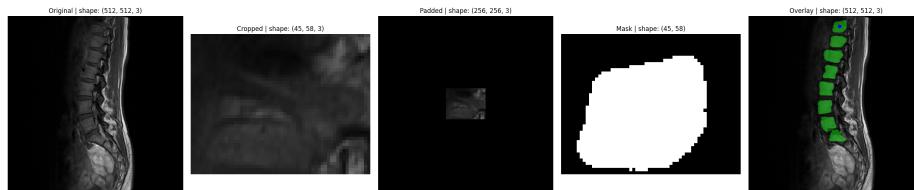


Fig. 11: Overview of the data processing pipeline. The figure illustrates the stages from the original spinal scan (left) to cropping, padding, and mask generation, concluding with the final segmentation overlay (right).

Dataset

The dataset is based on the Kaggle Lumbar Spine 2024 dataset, which contains a large collection of lumbar spine CT scans (approximately 147,000 images). As the original dataset did not include vertebra-level annotations required for our models, we manually labeled the vertebrae using the Roboflow platform. This process involved identifying and annotating individual vertebrae in each scan,

resulting in a custom-labeled dataset specifically suited for vertebra detection and segmentation tasks. The dataset is based on the Kaggle Lumbar Spine 2024 dataset, which contains a large collection of lumbar spine CT scans (approximately 147,000 images). As the original dataset did not include vertebra-level annotations required for our models, we manually labeled the vertebrae using the Roboflow platform. This process involved identifying and annotating individual vertebrae in each scan, resulting in a custom-labeled dataset specifically suited for vertebra detection and segmentation tasks. Additionally, the original dataset contained a significant number of images that were not relevant for our task, such as non-sagittal or frontal planes. To improve training efficiency and model performance, we filtered out these irrelevant images. Small proportion of less relevant scans were intentionally retained in the dataset. This decision was made to improve the robustness and generalization ability of our models by exposing them to a broader variety of real-world imaging scenarios.

The final dataset consists of cropped and labeled images, tailored for both detection and segmentation(YOLO costum dataset, UNet costum dataset). Despite the smaller size (approximately 150 images) resulting from manual labeling and filtering, the dataset provided a sufficient foundation for training and evaluating both the YOLOv8 detection model and the Attention U-Net segmentation model.

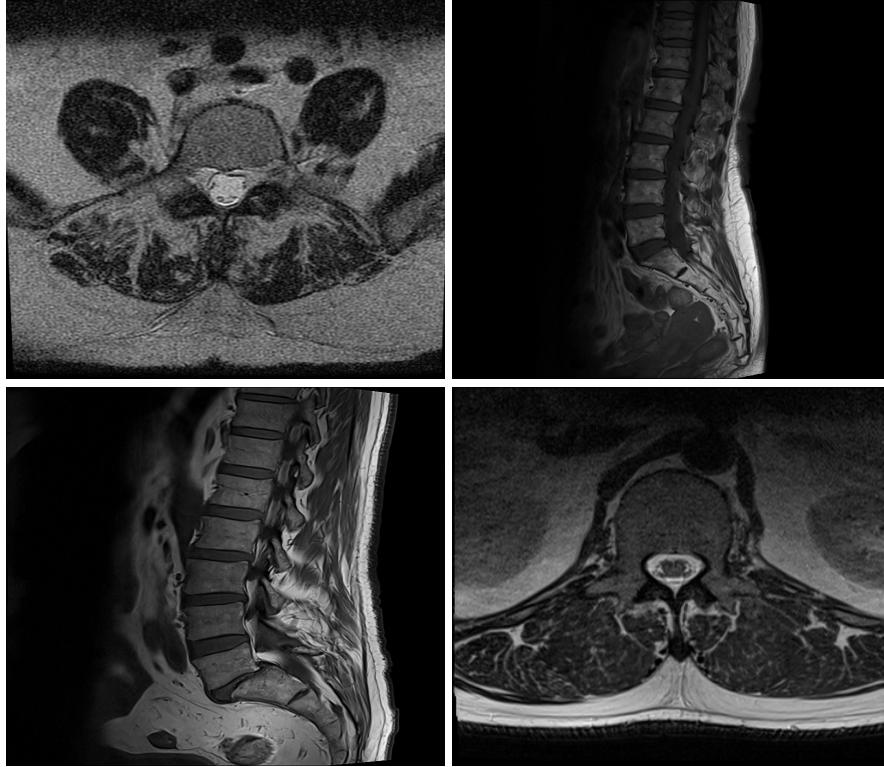


Fig. 12: Samples from RSNA 2024 Lumbar Spine

4.1 YOLOv8

Training Methods

The YOLOv8 model was trained using the Ultralytics training pipeline for vertebra detection, utilizing the custom-labeled CT scan dataset described above. Prior to training, the dataset underwent data augmentation including random horizontal flipping, Gaussian blur (with a maximum intensity of 0.5 pixels), and the application of random noise to up to 0.3.

Hyper-parameters and results

The model demonstrated excellent performance across all key evaluation metrics. The training and validation loss curves decreased steadily and showed no signs of overfitting, indicating stable and effective learning.

Throughout training, both precision and recall rapidly converged to values near 1.0, with validation **mAP@0.5: 0.99** and **mAP@0.5:0.95 stabilizing around**

| Parameter / Result | Run 1 (50 epochs) | Run 2 (100 epochs) | Run 3 (200 epochs) |
|--|------------------------|----------------------|----------------------|
| Epochs | 50 | 100 | 200 |
| Batch Size | 8 | 8 | 8 |
| Initial Learning Rate (pg0) | 1.676×10^{-3} | 2.0×10^{-5} | 3.0×10^{-5} |
| Drop Rate (1 - mAP) (\downarrow) | 0.20087 | 0.22034 | 0.42038 |
| mAP@0.5:0.95 (\uparrow) | 0.77966 | 0.79913 | 0.57962 |

Table 1: Comparison of YOLOv8 training configurations and performance results.

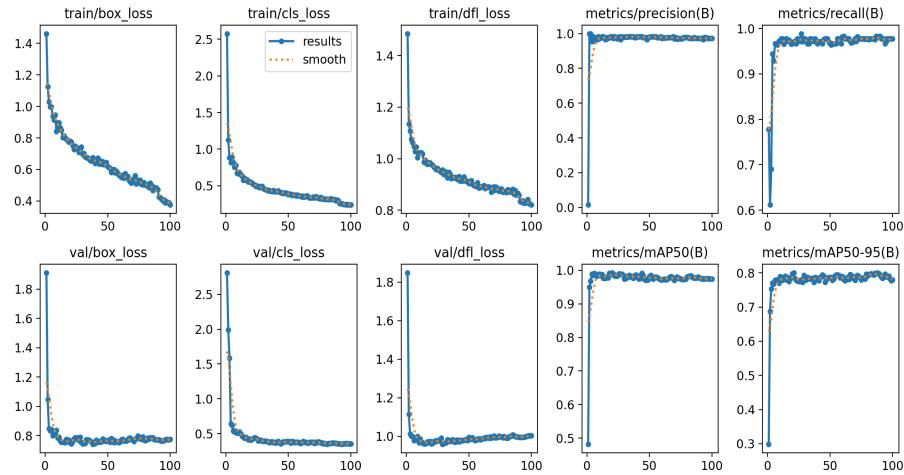


Fig. 13: Training Result For YOLOv8

0.79. These results indicate that the model reliably and accurately detects vertebrae with minimal false positives or false negatives on the validation dataset.

4.2 Attention U-Net

After detection, we used the YOLOv8 outputs to generate input samples for the Attention U-Net, which performs vertebra segmentation. Each detected vertebra region was cropped and resized to a fixed input size of 256×256 pixels. The bounding box of the vertebra was centered within this frame, and the surrounding area was padded with a black background as needed to preserve the input dimensions. This preprocessing step ensured consistent input formatting, with one vertebra per image. The U-Net was trained using the Adam optimizer, with each sample consisting of a vertebra image and its corresponding segmentation mask.

The staged training approach—first YOLOv8 for detection, then Attention U-Net for segmentation—allowed us to leverage accurate detection results to gen-

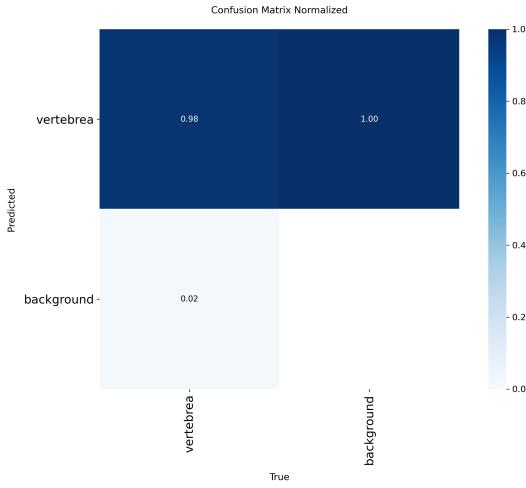


Fig. 14: YOLOv8 Training Confusion Matrix

erate high-quality segmentation masks. The dataset used for U-Net training underwent additional augmentation, including random vertical flipping, exposure adjustment between -10% and $+10\%$, and Gaussian blur with a maximum intensity of 0.7 pixels.

Hyper-parameters

| Parameter / Metric | Run 1 | Run 2 | Run 3 | Run 4 |
|--------------------|--------------------|--------------------|--------------------|--------------------|
| Epochs | 500 | 200 | 100 | 100 |
| Batch Size | 8 | 8 | 16 | 8 |
| Learning Rate | 5×10^{-6} | 1×10^{-5} | 1×10^{-4} | 1×10^{-5} |
| Dice Score | 0.97 | 0.97 | 0.91 | 0.95 |
| IoU Score | 0.94 | 0.80 | 0.95 | 0.80 |

Table 2: U-Net training configuration and performance comparison.

The Attention U-Net model was trained for 500 epochs with a batch size of 8 and a learning rate of 5×10^{-6} . The training performance was evaluated using the Dice Score and Intersection-over-Union (IoU) metrics, both computed for the training and validation sets across epochs.

As shown in Figure 17, the model exhibited steady and consistent improvement in both Dice and IoU scores throughout training. The training curves indicate a smooth learning trajectory with no signs of overfitting. Additionally, the training and validation loss curves (Figure 16) demonstrate a gradual decrease, further confirming the model’s convergence.

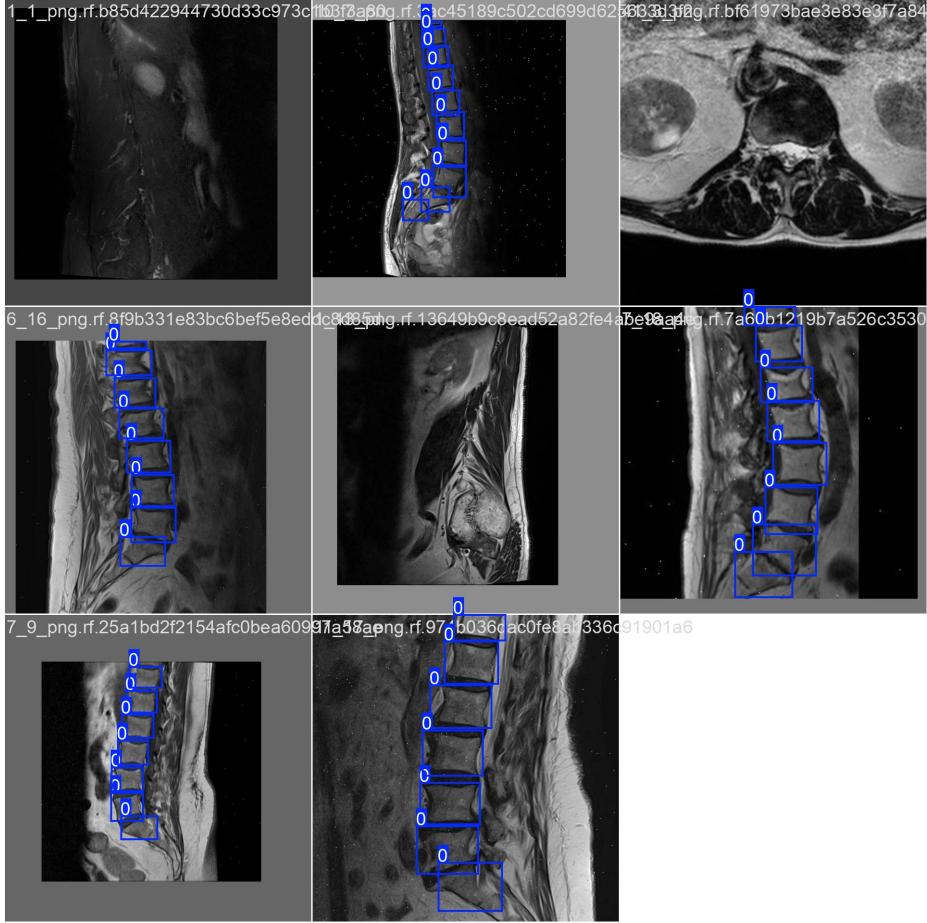


Fig. 15: Sample outputs from the YOLOv8 detection model applied to CT scans. The model successfully detects vertebrae across different patients and imaging angles, marking each vertebral body with a blue bounding box. Each box is labeled with the predicted class (vertebra index), demonstrating the robustness of the model in handling various anatomical configurations and image qualities.

4.3 Linear Function Pipeline

The final stage of BAYU-NET utilizes the segmentation results to create a model representing the segmented spine. The main idea is that with access to a large and diverse dataset (which we did not have), it would be possible to build a general model representing the average healthy spine. This model could then be used as a reference to compare with new segmented spines, allowing automatic detection of spinal deviations and helping medical professionals assess conditions like scoliosis. However, due to the limited size of our dataset, we instead provide

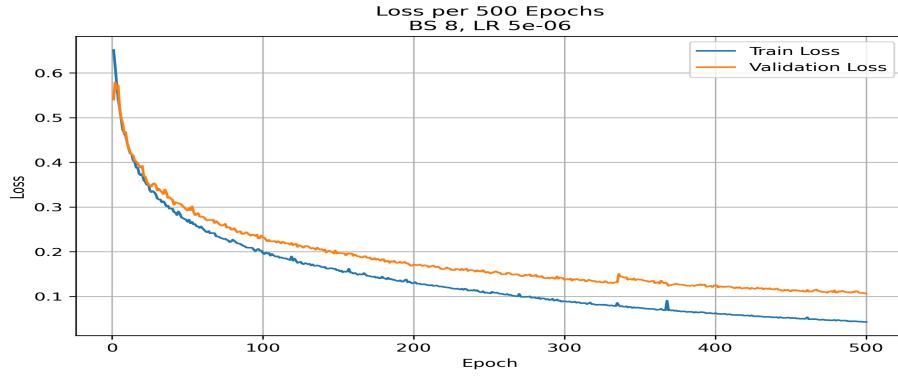


Fig. 16: Training and validation loss across 500 epochs

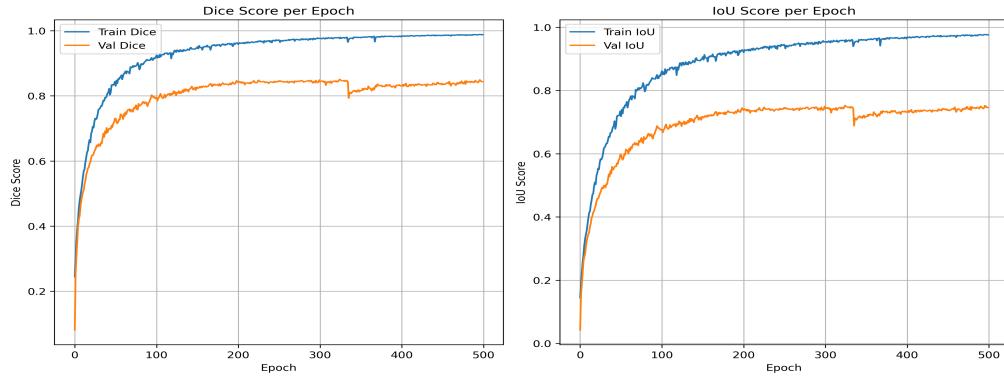


Fig. 17: Dice and IoU scores per epoch for training and validation sets.

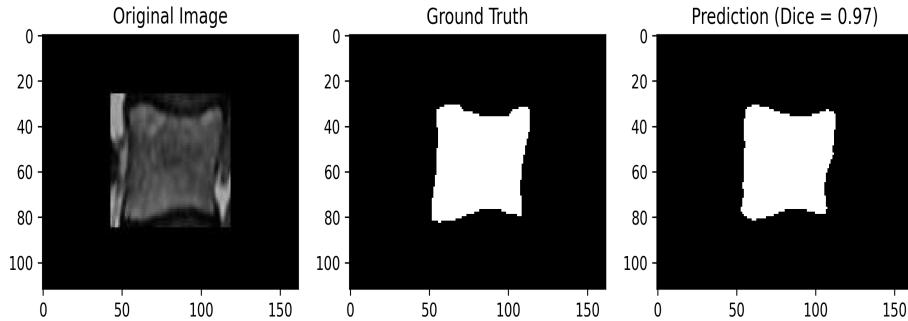


Fig. 18: Example of vertebra segmentation result using the Attention U-Net.

a basic linear representation of the spine. This can be used for manual analysis and sets the groundwork for future development.

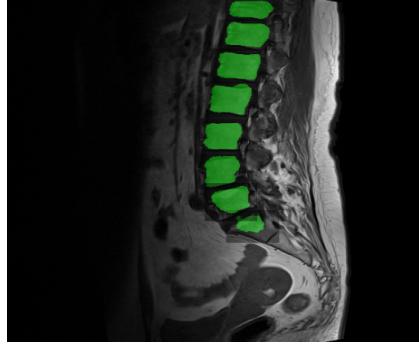


Fig. 19: Example of full spine segmentation result using the Attention U-Net.

5 Discussion

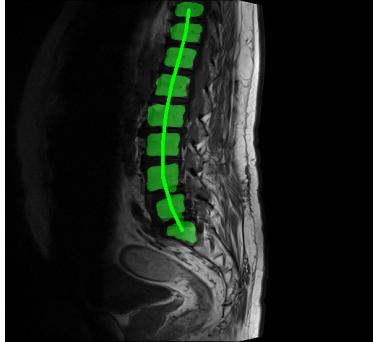
During the preparation of our dataset for model training, we faced several challenges. Initially, there were no existing annotations for our images, so we had to manually create them using the Roboflow platform. This required us to construct a small but reliable dataset to validate our project’s credibility and evaluate potential improvements. Integrating YOLO and U-Net models also presented difficulties, particularly regarding data compatibility and preprocessing. Specifically, we needed to ensure that the images output by YOLO could be properly formatted and resized as input for U-Net. This step was crucial to maintain consistency in image dimensions and formats between the two models. Annotation and preprocessing were necessary to make YOLO and U-Net work well together in our project.

5.1 Future Work

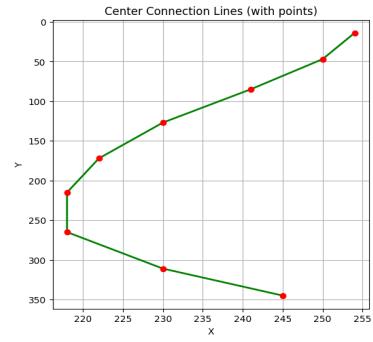
Moving forward, we intend to significantly increase the size and diversity of our dataset by including a larger number of annotated CT scans. Expanding the dataset will not only improve the model’s robustness and accuracy, but also enhance its ability to generalize to a wider range of patient anatomies and clinical scenarios. Incorporating more data will also allow the model to better identify rare or atypical spinal conditions. More over that we aim to provide evaluation of each patient’s scoliosis severity. This can be achieved by comparing the predicted alignment of the patient’s spine to the average alignment of healthy spines. By calculating the distance

$$d(C_{\text{truth}}, C_{\text{given}}) = \|C_{\text{truth}} - C_{\text{given}}\|$$

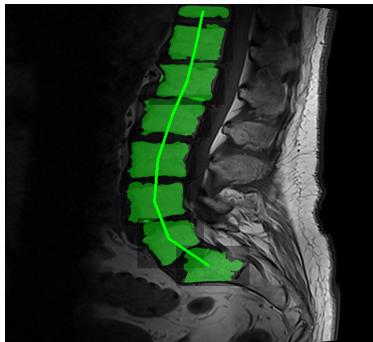
we can objectively measure deviations from the healthy reference. This approach will allow us to assign a scoliosis level based on how much the patient’s spine differs from the healthy average, offering a more precise and clinically meaningful assessment.



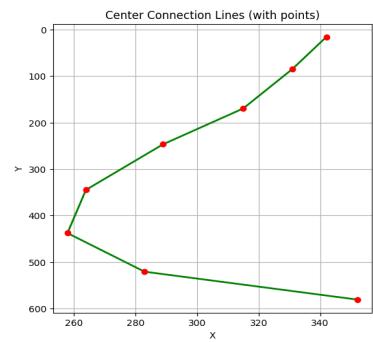
(a) Segmented spine A



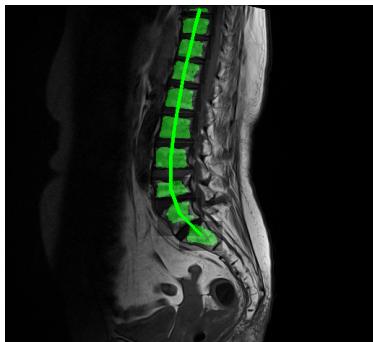
(b) Linear function A



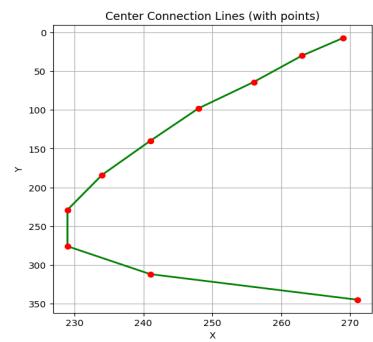
(c) Segmented spine B



(d) Linear function B



(e) Segmented spine C



(f) Linear function C

Fig. 20: Comparison of three segmented spines (left column) and their corresponding linear function (right column). Each pair demonstrates how the simplified model captures the general structure of the spine, allowing manual analysis of curvature and potential deviations.

6 Conclusion

Despite being trained on a relatively small, manually labeled dataset, our framework achieved strong performance in vertebra detection and segmentation tasks. The models demonstrated high accuracy and consistent results across the validation set, highlighting the effectiveness of the staged approach and tailored preprocessing pipeline. We anticipate that training on a larger and more diverse dataset would further enhance the model's generalization capabilities and robustness. More diverse dataset would allow the network to better handle anatomical variations, ultimately improving its reliability in clinical settings. Furthermore, access to a large-scale dataset would enable the construction of a more generalized and accurate spine structure analysis pipeline, including scoliosis detection and severity assessment.

7 User Documentation

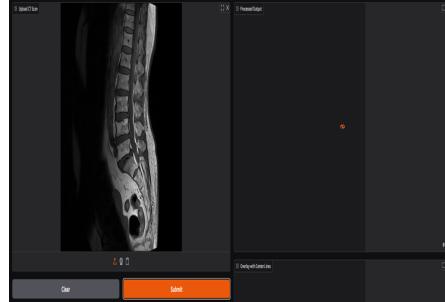
Purpose: This guide provides instructions for maintaining and updating the project deliverables after completion. It is intended to support future changes, updates, and improvements, ensuring continued use of the system.

7.1 GUI

The GUI allows users to upload medical imaging scans, such as CT images, directly into the system for processing. It provides an intuitive interface for visualizing both the original scans and the segmented vertebrae, with option export the segmentation data and scoliosis curve models for further analysis or incorporation into clinical reports.



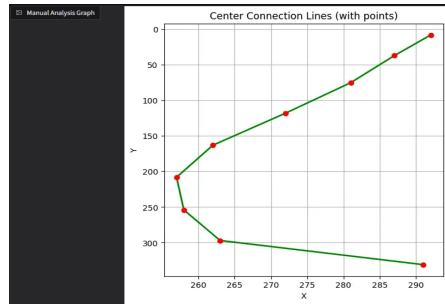
Segmented Spine Overview



Loading scan screen



Segmented Spine With Lines Overlay



Graph Representing the Spine - Ready For Manual Analysis

Fig. 21: Graphical User Interface (GUI) overview.

7.2 User Manuel

Purpose: This guide provides operating instructions for the Colab notebook used to segment spinal CT images and analyze scoliosis.

1. **Access the Notebook:** Open the 'BayuNet.ipynb' Google Colab notebook and sign in with your Google account.
2. **Set Up the Environment:** (Optional) Go to *Runtime* → *Change runtime type* and select *GPU* for faster performance.
3. **Ensure Model Weights Availability:** Verify that the files specified by `yolo_model_path` and `unet_model_path` are available.
4. **Run All Cells:** Execute the setup cells at the top of the notebook to load all dependencies and models. Scroll down to reach the user interface section. Note that this process may take some time, and you may be prompted to grant the notebook access to your Google Drive.
5. **Upload a CT Image:** Press the designated upload button and select your CT image file.

6. **Run Segmentation:** Press Execute for segmentation. Results, including the segmented vertebrae and spinal curve, will appear.
7. **Review Results:** Inspect the visual output and any computed measurements provided.
8. **Export Results:** Run the export cell and download the output files to your computer.

Note: For any technical questions, refer to comments in the notebook or contact us.

7.3 Maintainer Manual

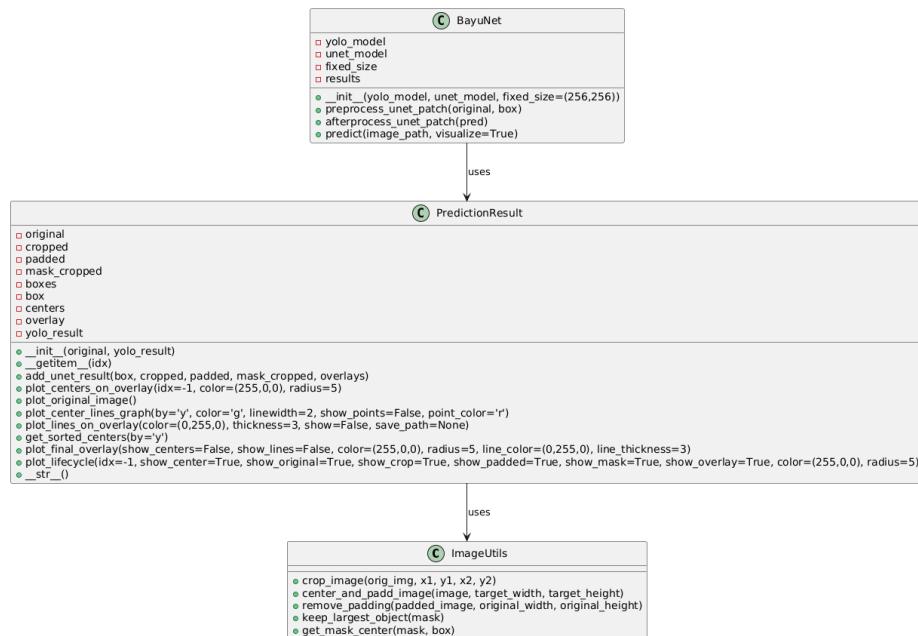


Fig. 22: BayuNet Implementation Class Diagram

Update Model Weights:

- Perform model training in `UNet+YOLOv8.ipynb`, following the workflow outlined for each model.
- Each training session saves the resulting model weights to Google Drive for easy access and backup.
- Model weights can be downloaded, loaded, and tested locally as needed.

Dataset Preparation and Annotation:

- Sign in to Roboflow, upload your dataset, and complete the required annotations (masks or bounding boxes).
- Export annotated data in YOLOv8 or COCO format.
- For UNet training, further process the COCO annotations into PNG masks and organize them into `train`, `valid`, and `test` folders using `UNet_dataset.ipynb`.
- Use the prepared dataset for model training and evaluation.

Application and BAYUNet Interface Modification: The `BayuNet.ipynb` notebook contains the main implementation of the BayuNet architecture, centered around the `BayuNet` class described at Fig.22. To extend the system's functionality or add new capabilities, you may modify or expand the `BayuNet` class directly within the notebook.

The user interface components, such as input widgets and result visualizations, are implemented in the final cells of the notebook. These can be updated to improve usability or add new features as required.

For any updates, enhancements, or bug fixes, ensure that changes are thoroughly tested within the notebook. It is recommended to save new versions of the notebook to your repository or shared drive and to document any modifications clearly to maintain code readability and facilitate future maintenance.

References

1. Anatomy, T.M.: Bones of the back (2025), <https://teachmeanatomy.info/back/bones/>
2. Blog, R.: What is yolov8? (2025), <https://blog.roboflow.com/what-is-yolov8/>
3. clevelandclinic: Spine structure and function (2023), <https://my.clevelandclinic.org/health/body/10040-spine-structure-and-function>
4. clevelandclinic: Scoliosis (2024), <https://my.clevelandclinic.org/health/diseases/15837-scoliosis>
5. Documentation, U.: Yolov8 (2023), <https://docs.ultralytics.com>
6. Fritz Hefti, M.D., P.: Pediatric orthopedics in practice - cobb and scoliosis (2007), https://books.google.co.il/books?id=VRnFkfvRT4EC&pg=PA56&source=gbs_selected_pages&cad=1#v=onepage&q=cobb&f=false
7. Frost, B.A., Camarero-Espinosa, S., Foster, E.J.: Materials for the spine: Anatomy, problems, and solutions. Materials **12**(2) (2019). <https://doi.org/10.3390/ma12020253>, <https://www.mdpi.com/1996-1944/12/2/253>
8. Oktay, O.: Attention u-net: Learning where to look for the pancreas (2024), <https://openreview.net/pdf?id=Skft7ciJM>
9. Orthopedics, Traumatology: Idiopathic scoliosis (2023), <https://www.elitenicosia.com/en/cocuklarda-ve-ergenlerde-idiyopatik-skolyoz/>
10. Radiopaedia: Cobb angle (2024), <https://radiopaedia.org/articles/cobb-angle?lang=en>
11. Redmon, J.e.a.: You only look once (yolo) (2025), <https://arxiv.org/pdf/1506.02640>
12. Reis, D.: Real-time flying object detection with yolov8 (2024), <https://arxiv.org/pdf/2305.09972>
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015), <https://arxiv.org/abs/1505.04597>
14. Stokes, I.A.F.C.: Three-dimensional terminology of spinal deformity (1994), https://journals.lww.com/spinejournal/abstract/1994/01001/three_dimensional_terminology_of_spinal_deformity_.20.aspx
15. Unknown, A.: Congenital scoliosis (2003), <https://link.springer.com/article/10.1007/s00586-003-0555-6>
16. Unknown, A.: Adolescent idiopathic scoliosis: What you need to know (2021), <https://skoliosis.my/scoliosis-types/adolescent-idiopathic-scoliosis/>