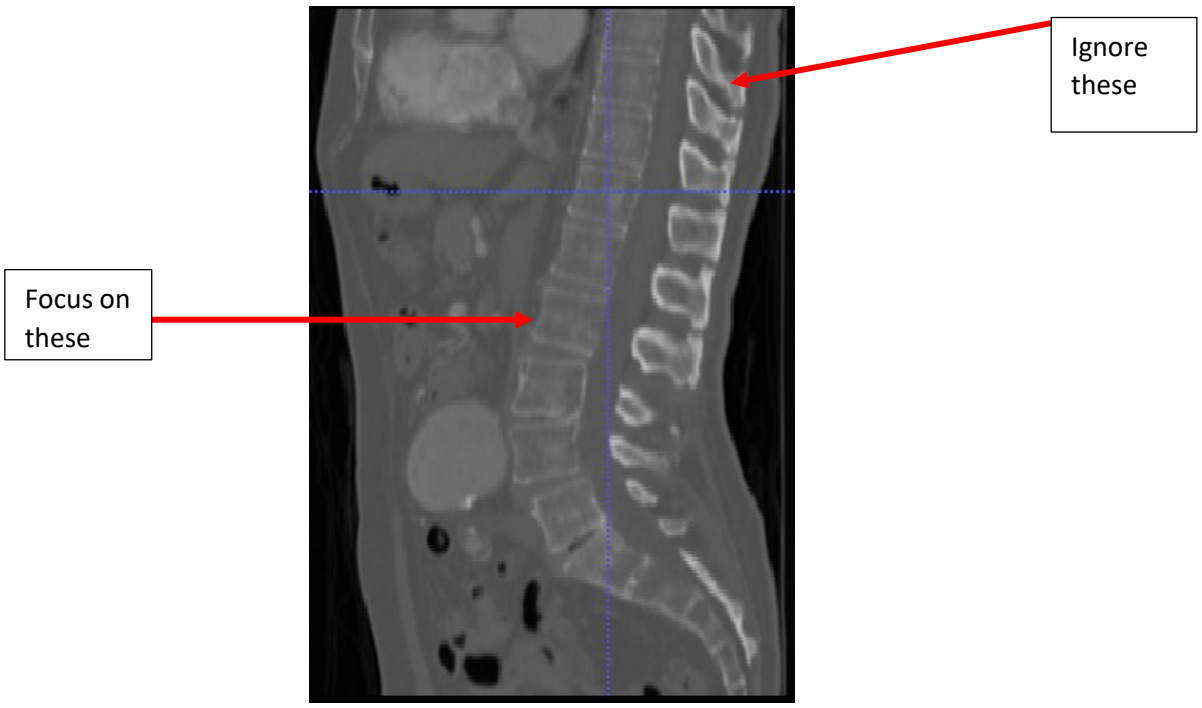
HC701 – Assignment # 2

In this assignment you must implement the following tasks and submit the code along with a brief report summarizing your findings. You will be able to find the data for each task in the following shared directory: (/apps/local/shared/HC701/assessment/assignment_2/). You must submit a PDF of your work.

Task 1: Windowing and detection in a CT scan (5/10 marks, 1+4 marks on the tasks respectively)

Task 1.1. Write a Python code to read the provided CTA abdominal scan (CTA-Abdomen.nrrd) and to find the optimal intensity window for the spine vertebrae (check the figure below) by visualizing the result when choosing the best range. Output a screen shot from the mid axial, coronal and sagittal views, and the best intensity window you selected.



Task 1.2. Following on from Task 1.1, regardless of the intensity window you used:

- Research and identify effective **non-ML** methods to locate the vertebrae in the 2D sagittal view (see image above, please work on the mid sagittal slice only). Report details about the method you used and the algorithms behind it (maximum 300 words).
- Write a Python code to remove all non-vertebra regions from the 2D image. Report your code and a paragraph (max 100 words) describing what you did. Also report a screenshot of the generated image.
- Develop a non-ML method to put a box around each vertebra and report a screen shot of the boxes. Describe how you achieved this task (max 100 words).
- Report the number of identified vertebra.

- Apply the same method on 10 sagittal slices before and after the mid-sagittal slice (21 slices in total). Generate a 3D box around each vertebra and report a 3D screen shot of the localization.

Task 2: Classification of Tuberculosis in X-ray (5/10 marks, 1 + 4 marks)

Get the X-ray Tuberculosis dataset from Kaggle. It contains many X-ray images from multiple classes, but we will only work with two classes (Healthy and TB) for binary classification. There are 3800 healthy x-ray images and 800 TB x-rays.

Task 2.1. Split data for training and testing. You need to use the **first** (sorted in an ascending order by ID) 20% of images **per class** for testing and the remaining 80% for training. Create a CSV file for this. Report the number and the range of filenames for each class in training and testing sets.

Task 2.2. Develop a machine learning algorithm trained and tested on Task 2.1 data.

- Describe the data augmentation strategy you use and the choice of network architecture.
- You must try **exactly** 5 experiments not more. You must make network selection and design decisions on what you want to experiment with before you start. Therefore, you must make informed decisions and demonstrate a research lead thought process. Report details of the 5 experiments within a table focusing on the backbone used with citation, main hyper-parameters used, why you decided to use this model and any other details you want to mention.
- Report the test accuracy, f1 score and confusion matrices for all experiments, as well as figures for training accuracy and loss for the top 2 experiments you did.
- Report the number of parameters and FLOPS in a table for each model.